Алексей Стахнов

Linux

Санкт-Петербург
«БХВ-Петербург»
2002

УДК 681.3.06
ББК 32.973.26-018.2
С78

Стахнов А. А.

С78 Linux. — СПб.: БХВ-Петербург, 2002. — 912 с.: ил.

ISBN 5-94157-146-1

Книга посвящена операционной системе Linux. Приводятся подробные сведения о ее особенностях и возможностях, идеологии файловой системы, инсталляции и основных командах, вопросах компиляции ядра, настройках и сервисах. Большое внимание уделяется организации на базе Linux различных серверов и служб: электронной почты, WWW, FTP, INN, Proxy, NTP, а также проблемам администрирования сети, обеспечения безопасной работы и другим вопросам. Описаны способы настройки под Linux рабочих станций, в т. ч. и бездисковых, установки и эксплуатации на них графических сред типа X Window, а также конфигурирование модемных соединений, принтеров и сканеров, отладка взаимодействия с Linux-машинами такой "экзотической" периферии, как карманные компьютеры, мобильные телефоны, TV-тюнеры и т. п. Рассматриваемые в книге конфигурационные файлы и структура каталогов соответствуют дистрибутиву Red Hat Linux 7.*x*, тем не менее, при минимальной адаптации все упоминаемые в книге пакеты устанавливаются в любом дистрибутиве Linux.

Для начинающих администраторов или пользователей Linux

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Анатолий Адаменко*
Зав. редакцией	*Анна Кузьмина*
Редактор	*Григорий Добин*
Компьютерная верстка	*Натальи Смирновой*
Корректор	*Наталия Першакова*
Дизайн обложки	*Игоря Цырульникова*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 26.09.02.
Формат 70×100$^{1}/_{16}$. Печать офсетная. Усл. печ. л. 73,53.
Тираж 3000 экз. Заказ №
"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02
от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в ФГУП ордена Трудового Красного Знамени "Техническая книга"
Министерства Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.
198005, Санкт-Петербург, Измайловский пр., 29.

ISBN 5-94157-146-1

Содержание

Часть I

Введение в Linux

Глава 1

Особенности ОС Linux

Я считаю, что Microsoft создала объективно плохую операционную систему, и мне интересно наблюдать, как это постепенно доходит до людей.

Линус Торвальдс, создатель ОС Linux

Мир операционных систем предоставляет пользователям достаточно большое их количество. Мы не будем подробно останавливаться на истории и особенностях каждой операционной системы — для этого понадобится чрезмерно много места и времени. Да и не всем это интересно: раз вы читаете эту книгу, значит вас заинтересовала Linux. Особо любопытным можно предложить обратиться к соответствующей литературе и Интернету — там можно найти много интересной информации на этот счет.

Какие же операционные системы используются в настоящее время? Вот некоторые из них:

- DOS (MS-DOS, DR-DOS и их клоны);
- Windows 3.1*x*;
- OS/2;
- Windows 9*x*;
- Windows NT (Windows 2000);
- Mac OS;
- Mac OS X;
- семейство UNIX;
- FreeBSD, OpenBSD, NetBSD;
- Linux;
- BeOS;
- QNX.

Конечно, приведенный список далеко не полон, но мы сознательно ограничим его операционными системами, используемыми наиболее часто. Рассмотрим этот список с точки зрения человека, которому необходимо иметь

на компьютере операционную систему, удовлетворяющую нескольким, порой противоречивым, требованиям.

Операционные системы можно классифицировать по многим параметрам. Во-первых, все они делятся на два вида — платные и бесплатные (условно-бесплатные). Во-вторых, правомерно разделить их на операционные системы с открытым исходным кодом (с правом вносить изменения или без права внесения изменений) и с закрытым исходным кодом. В-третьих, операционные системы бывают одноплатформенные (способные функционировать только на одной платформе, например PC-совместимой) и многоплатформенные (способные функционировать на нескольких платформах, например PC-совместимой, Macintosh, Sun, PowerPC). В-четвертых, операционные системы могут быть однозадачными и многозадачными. В-пятых, однопользовательскими и многопользовательскими. В-шестых, серверными, клиентскими или универсальными. В-седьмых, иметь текстовый, графический или тот и другой интерфейсы. В-восьмых, ориентацией на работу с сетью и Интернетом. В-девятых, по потребляемым ресурсам и т. д. А ведь это только начало. Критериев, по которым выбирается операционная система, намного больше. Имеют право на существование и такие критерии, как "Она мне нравится" или "Друг себе поставил, чем я хуже?" А ведь выбор операционной системы определяет, как вы будете жить и работать ближайшие несколько лет (или десятилетий). Поэтому к выбору операционной системы следует относиться с большой тщательностью и достаточной долей скептицизма. Сжато охарактеризуем наиболее популярные операционные системы.

Замечание

В 1980 году была организована инициативная группа под названием /usr/group с целью стандартизации программного интерфейса UNIX. Стандарт был разработан к 1984 году и использовался комитетом ANSI при описании библиотек языка C. В 1985 году был создан Portable Operating System Interface for Computing Environment, сокращенно POSIX (переносимый интерфейс операционной системы для вычислительной среды). На сегодняшний день большинство операционных систем удовлетворяют (полностью или частично) стандарту POSIX.

DOS

Производитель — Microsoft. Была создана на заре эры PC. 16-разрядная однопользовательская, однозадачная система. Платная, закрытый исходный код. Одноплатформенная (PC-совместимая). Текстовый интерфейс (командная строка). Достаточно простой процесс установки. Файловая система — FAT 12, FAT 16, FAT 32. Поддержка локальной сети — драйверы и программы сторонних производителей. Интернет — аналогично. Морально устарела еще до рождения. Разрабатывалась поспешно, без учета развития ап-

паратных и программных средств. При появлении нового аппаратного обеспечения срочно дорабатывалась, более или менее устойчивая система появилась только к 1990-му году (по прошествии 10 лет!). Последняя самостоятельная версия (как отдельного продукта) — MS-DOS 6.22. Последние версии включали в основном косметические изменения. Но (благодаря дружбе IBM и Microsoft) — получила широчайшее распространение, имела малую потребность в ресурсах, поддерживала практически все выпускаемое для PC аппаратное обеспечение, и для своей ниши была почти оптимальным решением с точки зрения цена/ресурсы/производительность. Для DOS было выпущено несметное количество приложений, последние пять лет новые программные продукты для DOS не выпускаются. Существовали многочисленные клоны. Наиболее известные — IBM-DOS, PC-DOS, DR-DOS.

Windows 3.1*x*

Строго говоря — это даже не операционная система, а графическая оболочка или, если хотите, надстройка над DOS. Почему версии 3.1*x*? Потому что, по большому счету, только к версии 3.1 была достигнута достаточная стабильность, неплохая функциональность, накоплена критическая масса приложений и реализована многозадачность. Благодаря Windows в мир PC был внесен дух однообразия и предсказуемости. Интерфейс приложений стал стандартизирован, системные библиотеки унифицированы, внедрена поддержка локальной сети, появился достаточно большой набор драйверов для аппаратного обеспечения, поставляемых в дистрибутиве Windows. В комплект Windows вошли несколько десятков приложений, благодаря чему потребитель получал почти готовую систему для офисной работы. Доступ в Интернет по-прежнему осуществлялся с помощью программ сторонних производителей. Система получилась не очень надежная, достаточно часто происходили зависания. Для Windows 3.1*x* было выпущено много разнообразного программного обеспечения, впрочем, после 1996 года новые программные продукты для Windows 3.1*x* больше практически не выпускались. Достаточно простой процесс установки. Слабая поддержка мультимедийных устройств. Платная. Исходный код недоступен. Последняя версия (16 битная) Windows for Workgroups 3.11. Требования к аппаратному обеспечению — процессор 386, не менее 2 (лучше 4) Мбайт оперативной памяти, не менее 16 Мбайт места на жестком диске.

OS/2

Совместная разработка IBM и Microsoft. Впоследствии IBM и Microsoft разошлись, и OS/2 заканчивала и выпускала в свет только IBM, которая всегда позиционировала свою операционную систему для корпоративного использования. Из-за слабой маркетинговой политики IBM, сильной конкуренции

со стороны Microsoft, относительно малого количества разработанных под систему программных продуктов IBM отказалась от дальнейшего продвижения OS/2. Для своего времени (а это расцвет DOS и Windows 3.1*x*) была очень прогрессивна. В дальнейшем (версии 2—4) получила существенное расширение функциональности. 32-битная многозадачная операционная система. Отличная поддержка локальной сети и Интернета. Помимо программ, разработанных специально для OS/2, система позволяет запускать как DOS-приложения, так и приложения Windows 3.1*x*. Разрабатывается приложение Odin, позволяющее выполнять приложения Windows 9*x*. Файловая система — HPFS (High Performance File System). Поддерживает также достаточно большой список файловых систем, в том числе FAT 12, FAT 16, FAT 32. Требования к аппаратному обеспечению — процессор Pentium, не менее 32 Мбайт оперативной памяти, не менее 120 Мбайт места на жестком диске.

Windows 9*x*

32-битная операционная система с поддержкой 16-битных приложений (Windows 3.1*x*). Первая версия появилась в 1995 году. Затем в течение почти трех лет выходили исправления многочисленных ошибок. Попытка Microsoft объединить лебедя, рака и щуку — в одной операционной системе без проблем запускать исполняемый код DOS, Windows 3.1*x* (16 бит) и Windows 9*x* (32 бит). В результате получилась не очень устойчивая система, которую удалось отладить только к 1998 году (Windows 98). Несмотря на то, что Windows 9*x* объявлена операционной системой, она по-прежнему базируется на MS-DOS версии 7.0 со всеми ее рудиментами и наследственностью. В Windows 9*x* добавлено программное обеспечение, необходимое для работы в Интернете, и существенно расширен список драйверов для аппаратного обеспечения.

Windows 98 и последняя версия, Windows ME (Millennium Edition), фирмой Microsoft рассматриваются как переходный этап к Windows NT (Windows 2000), поэтому в Windows 9*x* частично включено программное обеспечение, характерное для серверов. Простой процесс установки и конфигурации системы, но отсутствует возможность тонкой ее настройки без стороннего программного обеспечения. Хорошая поддержка мультимедийных устройств и игр. Огромнейший список разработанного под систему программного обеспечения. На сегодняшний день Windows 9*x* и MS Office являются стандартом de-facto для настольных офисных систем. Платная. Исходный код недоступен. Требования к аппаратному обеспечению — процессор Pentium-166, не менее 32 Мбайт оперативной памяти, не менее 250 Мбайт места на жестком диске (для Windows ME не менее 64 Мбайт оперативной памяти, не менее 500 Мбайт места на жестком диске). Файловая система — FAT 32 (FAT 16).

Windows NT (Windows 2000)

Ощущая бесперспективность развития ветки DOS—Windows 3.1*x* Microsoft разработала новую операционную систему Windows NT (New Technology), базирующуюся на стандарте POSIX и новой файловой системе — NTFS (New Technology File System). Разработчики Windows NT серьезно взялись за проектирование операционной системы с учетом ее дальнейшего развития (совсем недавно вышла последняя в этой линейке операционная система — Windows XP). На сегодняшний день Windows NT — флагманская операционная система фирмы Microsoft. Существует две ее версии — Windows NT Workstation и Windows NT Server. Первая версия предназначена для настольных систем, вторая — серверная платформа. В обеих версиях используется графический интерфейс, что для сервера — в общем-то излишество и нерациональная трата ресурсов. Многозадачная, многопользовательская, одноплатформенная (PC), весьма устойчивая операционная система. Имеет встроенную поддержку многопроцессорных и кластерных систем. Хорошая поддержка мультимедийных устройств и игр (однако не все существующие игры надежно работают под Windows NT). Полного набора серверных приложений не имеет (нет, например, Web-сервера). Исходный код недоступен. Платная. Windows NT Server рассчитан на малые и средние рабочие группы, на большие нагрузки пока не рассчитан. Помимо NTFS поддерживает файловые системы FAT и OS/2 (HPFS). Требования к аппаратному обеспечению Windows NT Workstation — процессор Pentium, не менее 128 Мбайт оперативной памяти, не менее 500 Мбайт места на жестком диске; Windows NT Server — процессор Pentium, не менее 256 Мбайт оперативной памяти, не менее 500 Мбайт места на жестком диске.

Mac OS

Производитель — Apple. Операционная система для Macintosh и его пользователей, многозадачная, однопользовательская, графическая. Первая версия была выпущена более 15 лет назад. С выходом Mac OS X считается окончательно устаревшей и постепенно сходит со сцены. Великолепная система для пользователя по эргономичности, дружественности и простоте освоения. За свою историю накопила достаточно большое количество устаревших концепций, оставленных для совместимости. Последние версии Mac OS были рассчитаны на компьютеры Apple, базирующиеся на процессоре Motorola 68040 и PowerPC с не менее 16 Мбайт оперативной памяти.

Mac OS X

Производитель — Apple. Новая операционная система для компьютеров Macintosh, базирующихся на процессоре PowerPC. UNIX-подобная, POSIX-

совместимая, многозадачная операционная система с графическим интерфейсом. Способна выступать как в качестве сервера, так и в качестве клиентской операционной системы. Платная, закрытый исходный код. Фирма Apple заявила о скором переносе Mac OS X на платформу Intel. Поддерживает сетевую файловую систему NFS (Network File System). Mac OS X предназначена для работы на компьютерах Power Macintosh с процессором G3. Требования — не менее 128 Мбайт оперативной памяти. Рекомендуемый объем жесткого диска — 5 Гбайт.

Семейство UNIX

Группа операционных систем, имеющих общего предка и традиционно носящих название UNIX. Фирмы производители — AT&T, DEC, Sun, Hewlett-Packard, IBM, SCO и многие другие. Несмотря на то что первая версия UNIX была выпущена еще тридцать лет назад, UNIX до сих пор считается наиболее современной, надежной и динамично развивающиеся операционной системой. Большой вклад в успех UNIX внесли специалисты AT&T, студенты и преподаватели университета Беркли. На сегодняшний день UNIX той или иной фирмы-производителя установлен практически на каждом сервере уровня предприятия, больших кластерах и мультипроцессорных системах, а также на многих рабочих и графических станциях. Многоплатформенная, мультизадачная, многопользовательская операционная система. Поддерживает кластеризацию, мультипроцессорные системы, распределенные вычислительные среды, массивы накопителей огромной емкости и многое другое. На сегодняшний день трудно найти компьютер, на котором не смогла бы работать одна из версий UNIX. Как правило, UNIX, выпускаемая фирмами, — платная, с закрытым исходным кодом. Тем не менее, существует достаточно много (например, семейство BSD, Linux) бесплатных, с открытым исходным кодом UNIX-совместимых операционных систем. Благодаря стандарту POSIX практически любое приложение можно перенести из одного представителя семейства UNIX в другой. Благодаря этому для UNIX имеется огромное количество как бесплатных, так и коммерческих программ. Как правило, для каждой разновидности UNIX разработана своя файловая система, но все разновидности UNIX могут работать с распространенными файловыми системами. В том числе существуют и т. н. *журналируемые* файловые системы. В журналируемых файловых системах для решения проблемы повреждения структуры файловой системы или хранения данных применяют транзакции, используемые практически в любой базе данных. Транзакция считается незавершенной до тех пор, пока все изменения не сохранены на диске. А чтобы сбои, происходящие до завершения всех операций, входящих в транзакцию, не приводили к необратимым последствиям, все действия и все изменяемые данные протоколируются. В том случае, если все-таки сбой произойдет, по протоколу можно вернуть

систему в рабочее состояние. Требования к аппаратной платформе — самые разнообразные. Как уже упоминалось выше — трудно найти компьютер, на котором не смогла бы работать одна из версий UNIX.

FreeBSD, OpenBSD, NetBSD

Операционные POSIX-совместимые системы семейства UNIX на основе кода университета Беркли. Принципиальные различия:

- FreeBSD — очень надежная, достаточно консервативная (в хорошем смысле этого слова). Аппаратная платформа — Intel;
- NetBSD — переносимость на большое количество аппаратных платформ;
- OpenBSD — попытка объединить достоинства FreeBSD и NetBSD в одном дистрибутиве.

Бесплатные, открытый исходный код. На сегодняшний день наибольшее распространение из-за своей особой надежности получила FreeBSD. Двоичная совместимость со многими программами, построенными под SCO, BSD/OS, Net/Free/OpenBSD, 386BSD и Linux.

Linux

POSIX-совместимая UNIX-подобная операционная система. На сегодняшний день — самая распространенная бесплатная операционная система с открытым исходным кодом. При ее разработке из мира семейства UNIX старались взять все лучшее. Благодаря участию десятков тысяч разработчиков программного обеспечения и координации их действий через Интернет Linux и программное обеспечение для нее развивается очень динамично, ошибки и различные проблемы в программном обеспечении, как правило, исправляются в считанные часы после их обнаружения. Большую помощь в развитии и распространении Linux и сопутствующего ему программного обеспечения оказали фонд Свободного программного обеспечения (Free Software Foundation, USA) и лицензия GNU (The GNU General Public License, Универсальная общественная лицензия GNU) для программного обеспечения. На сегодняшний день существует одно ядро Linux, разработку которого координируют его создатели Линус Торвальдс и Алан Кокс, и множество дистрибутивов (не менее 2—3 десятков), отличающихся как функциональным назначением, так и составом программного обеспечения, входящим в дистрибутив. Существуют дистрибутивы, занимающие десяток компакт-дисков, и дистрибутивы, умещающиеся на одной-двух дискетах. Все, что справедливо для семейства UNIX — справедливо и для Linux. Широчайшая поддержка аппаратных платформ, малая требовательность к аппаратным ресурсам (процессор 486, 8 Мбайт оперативной памяти, винчестер

120 Мбайт). Масштабируемость, поддержка мультипроцессорных систем, кластеризация, поддержка распределенных вычислений, десятки графических оболочек — и это далеко не все. Поддерживаются десятки файловых систем, родная файловая система Ext2. И при всей мощи — достаточно дружественная операционная система, способная работать как на мощнейшем сервере, так и на стареньком "пентиуме" где-нибудь в офисе.

BeOS

Производитель — Be Inc. UNIX-подобная графическая операционная система. Однопользовательская. Сами разработчики позиционируют BeOS как операционную систему для работы с мультимедиа. Графический интерфейс. Очень молодая операционная система (по сравнению с ранее рассмотренными), поэтому относительно небольшой список поддерживаемого оборудования и программного обеспечения. BeOS поддерживает компьютеры с симметричной мультипроцессорной архитектурой (SMP) (до 8-ми процессоров), файловая система BFS (Be File System), 64-битная и журналируемая. Вытесняющая многозадачность, почти полная POSIX-совместимость. На сегодняшний день существуют две версии операционной системы — BeOS 5 PE (Personal Edition) — бесплатная (дистрибутив можно взять в Интернете, **www.be.com**) и BeOS 5 Pro — платная (поставляется на CD-ROM в коробке и с документацией). Исходный код недоступен (несколько программ, входящих в BeOS, имеют открытый исходный код). Поддерживает работу с несколькими файловыми системами — FAT 16, FAT 32 (возможно чтение и запись), NTFS (только чтение), HFS, ext2, CDFS. Аппаратные требования — Pentium-133, оперативная память — 32 Мбайт, место на винчестере — 512 Мбайт (минимально — 120 Мбайт). Возможен вариант установки под Windows и Linux. Аппаратная платформа — PC, PowerPC.

QNX

Производитель QNX — QNX Software Systems. UNIX-подобная POSIX-совместимая операционная система реального времени. 32-битная, многозадачная, многопользовательская, микроядерная. Первоначальное предназначение — промышленная операционная система, предназначенная для работы в режиме 99,999 % надежности ("пять девяток"). Используется для управления технологическими процессами, начиная от атомных электростанций и заканчивая производством мороженого. Исходный код закрыт. Проблемы с драйверами (малое количество). Минимальные требования для промышленного дистрибутива — 386-й процессор, 8 Мбайт ОЗУ. Помимо промышленных дистрибутивов QNX, стоящих немалые деньги, существует бесплатный вариант дистрибутива "QNX Real Time Platform", который загружается с сайта производителя (**www.qnx.com**). Минимальные требования

для бесплатного дистрибутива — процессор Pentium-200, 32 Мбайт ОЗУ, 100 Мбайт на жестком диске.

Ознакомившись с вышеприведенным кратким обзором операционных систем, можно представить в общих чертах их области применения, достоинства и недостатки. Поскольку наша книга посвящена Linux, а операционные системы Windows 9*x* или Windows NT/2000 установлены приблизительно на 90 % PC-совместимых персональных компьютеров, то все сравнения в дальнейшем мы будем проводить относительно этих трех операционных систем, не забывая, впрочем, и об остальных.

Небольшое отступление

Что такое пользователь? Никогда не задумывались? Понятие "пользователь" не подведешь под "среднестатистическое" значение. Он многолик и разнообразен. Единственное, что есть общего у всех пользователей компьютера — они сидят за компьютером. Пользователи с точки зрения системного администратора — все те, кто входит в систему в качестве пользователя, "юзера". С точки зрения системного программиста — все, кто запускает компьютер. Для разработчика прикладного программного обеспечения — пользователи его программы. Для авторов книг "... для чайников" — это люди, знающие о компьютере только то, что у него есть шнур питания и какая-то доска с кнопками. И так далее. Если попытаться обобщить, основной пользователь — это человек, который не разбирается в устройстве компьютеров, не знает, как настроить модем, не обязан знать тонкости операционной системы и т. п. Пользователь решает на компьютере свои профессиональные задачи, зачастую не имеющие с компьютерами ничего общего. На практике все это, конечно, не совсем так мрачно. Пользователь для успешной работы просто обязан знать, что такое файл, как настроить рабочий стол, установить программу, что такое вирусы и как с ними бороться и т. д. Пользователей условно можно разделить на три группы — не знающий о компьютере ничего, знающий кое-что и знающий многое. Соответственно, по уровням пользователей можно разделить операционные системы на три категории.

К первой категории можно отнести Mac OS и, в какой-то степени, Mac OS X, а также BeOS. Ко второй категории Windows 9*x*, OS/2. К третьей, как ни странно, — DOS, Windows 3.1*x*, Windows NT/2000, UNIX-семейство, BSD-семейство, Linux, QNX. Такое разбиение операционных систем не всегда соответствует официальному позиционированию фирм-разработчиков (например, Microsoft рекламирует Windows 9*x* как систему для домохозяек — включил и работай). Однако с точки зрения коллективного разума (по крайней мере, так считают авторы новостных конференций, посвященных сравнительному обзору операционных систем) данное нами разбиение операционных систем достаточно верно. Впрочем, жизнь, как всегда, не стоит на месте. Сейчас уже можно говорить, что Linux с ее графическими менеджерами окон KDE и GNOME постепенно переходит, если уже не перешла, ко второй категории (то есть для пользователей, знающих об операционной системе кое-что), при этом не теряя ни мощности, ни настраиваемости

всего и вся. Семейство Windows постепенно сдвигается к группе пользователей, не знающих об операционной системе ничего, при этом вызывая заметное раздражение знающих, или, как у них говорят — Advanced Users, своей уверенностью, что пользователь системе приносит только вред, а посему ничего настраивать он не должен, а если очень хочет — пусть платит за поддержку или специальное программное обеспечение. В идеале же операционная система должна удовлетворять, по меньшей мере, семи достаточно противоречивым требованиям.

1. Быть легкой в освоении и дружественной к пользователю (User Friendly).
2. Быть очень мощной и универсальной (способной работать на любом оборудовании).
3. В ней все должно настраиваться и достаточно просто.
4. Она должна быть очень надежна (в идеале — сверхнадежна).
5. Занимать как можно меньше места.
6. Разработчики моментально должны реагировать на проблемы, обнаруженные в процессе эксплуатации.
7. Под нее должен быть широкий выбор программного обеспечения.

В нескольких словах рассмотрим эти семь пунктов. Пункт *первый*. Тут, собственно, и так все ясно. От того, как быстро человек освоится с операционной системой и насколько удобно ему в ней работать, напрямую зависит производительность труда, да и просто хорошее настроение. Пункт *второй*. Можно, конечно, возразить, что чем более универсальный инструмент, тем слабее он для какого-нибудь специфического применения, и чисто теоретически это так. Но давайте посмотрим на универсальность с другой стороны. Теоретические принципы построения операционной среды, по большому счету, одинаковы, что для старенькой 386-й, что для новейших мультипроцессорных систем. Специфику платформы (тип процессора, мультипроцессорность, кластеризацию и т. п.) всегда можно учесть при разработке специфического ядра операционной системы или драйверов. Некоторая потеря в производительности с лихвой окупается тем, что пользователю, поработавшему на мощнейшем сервере и перешедшему на офисный компьютер, графическую станцию или домашний ПК, не придется осваивать другую операционную систему — его операционная система может работать на любом компьютере. А способность работать на любом компьютере автоматически подразумевает, что операционная система должна занимать как можно меньше места и потреблять мало аппаратных ресурсов. Пункт *третий*. И тут все понятно без пространных пояснений. Пользователь должен иметь возможность настроить операционную систему под свои нужды, не прибегая к стороннему (не входящему в поставку операционной системы) программному обеспечению. Пункт *четвертый*. Правда, больное место? У любого пользователя Windows со стажем наверняка происходило зависание компьютера,

причем в самое неподходящее время. И каждый пользователь хочет, чтобы зависания никогда не происходили на его компьютере. Пункт *пятый*. Это тоже понятно. Уже надоело каждые год-полтора менять жесткий диск только из-за того, что следующая версия операционной системы требует "совсем немного, только каких-то 3 Гбайта места на жестком диске". Пункт *шестой*. И это очевидно. Пользователь должен получить исправления к своей операционной системе при обнаружении просчетов ее разработчиков. Причем, как можно скорее, если операционная система удовлетворяет п. 4. И притом абсолютно бесплатно, поскольку это просчет разработчика. Пункт *седьмой*. Пусть операционная система будет самой распрекрасной, но если для нее нет программ, она не будет востребована.

Теперь оценим операционные системы на соответствие вышеперечисленным требованиям.

- DOS — не удовлетворяет ни одному пункту, кроме п. 7.
- Windows 3.1*x* — удовлетворяет п. 1 с оговорками, частично п. 3 и п. 5, удовлетворяет п. 7.
- OS/2 — удовлетворяет п. 1, п. 2 (с учетом одноплатформенности), п. 3, частично п. 4, п. 5 и п. 7.
- Windows 9*x* — удовлетворяет п. 1, частично п. 3, безусловно удовлетворяет п. 7.
- Windows NT (Windows 2000) — удовлетворяет п. 1, п. 2 (с учетом одноплатформенности и непомерных требований к аппаратному обеспечению), п. 3 и п. 4 с оговорками, безусловно удовлетворяет п. 7.
- Mac OC — безусловно удовлетворяет п. 1, п. 2 (с учетом одноплатформенности), частично п. 3, п. 4, п. 5, п. 6, удовлетворяет п. 7.
- Mac OC X — безусловно удовлетворяет п. 1, п. 2 (с учетом одноплатформенности и завышенных требований к аппаратному обеспечению), п. 3, п. 4, п. 6, пока не удовлетворяет п. 7.
- UNIX-семейство — безусловно удовлетворяет всем пунктам, кроме первого, да и то, в последнее время легкость освоения и дружественность у UNIX-разработчиков стоят на первом месте.
- FreeBSD, OpenBSD, NetBSD — все сказанное о UNIX-семействе справедливо и для этих операционных систем.
- Linux — безусловно удовлетворяет всем пунктам, особенно п. 2, п. 3, п. 6, п. 7.
- BeOS — удовлетворяет всем пунктам кроме (пока) п. 7.
- QNX — удовлетворяет всем пунктам.

Попробуем выбрать операционную систему, исходя из вышеперечисленных пунктов. DOS и Windows 3.1*x* отпадают сразу, как морально и физически

устаревшие продукты. OS/2 — очень неплохая операционная система, имеющая несколько недостатков: отсутствие перспектив (IBM отказалась от выпуска следующих версий), не очень большой выбор программного обеспечения, одноплатформенность. Mac OS, Mac OS X — также неплохие операционные среды как с точки зрения пользователя, так и с точки зрения администратора. Но — это операционные системы *только* для компьютеров фирмы Apple. А в нашей стране этих компьютеров не наберется и одного процента от общего количества персональных ЭВМ. QNX — достаточно специфичная система, рассчитанная для применения в сверхнадежных системах реального времени. Очень хорошая, но для нашего пользователя она стала доступна относительно недавно, поэтому в отношении к ней есть элементы недоверия и незнания, кроме того, у нее относительно малый список программного обеспечения общего назначения (офисные приложения, работа с графикой, игры, наконец). Что остается — семейство Windows 9*x*—Windows NT (включая Windows XP), семейство UNIX, а также представители "свободного мира" UNIX — FreeBSD, OpenBSD, NetBSD, Linux и стоящая немного особняком BeOS.

Теперь попытаемся максимально корректно сопоставить Windows-семейство и семейство UNIX. Сначала проведем четкий водораздел между операционными системами Windows 9*x*/ME и Windows NT/2000/XP. Подсознательно (в силу сходства названий, да и внешнего вида) пользователь, а иногда и администратор, отождествляет Windows 9*x*/ME и Windows NT/2000, хотя это далеко не одно и то же. Если внимательно посмотреть на характеристики Windows 9*x*/ME и немного сопоставить факты, станет понятно, что Windows 9*x*/ME — это затянувшийся на шесть лет переход от DOS/Windows 3.1*x* к Windows NT/2000, принесший, однако Microsoft огромный доход. С чисто технологической стороны UNIX-семейство корректно сравнивать только с Windows NT/2000, поскольку только Windows NT/2000, как система истинно многозадачная и многопользовательская, поддерживающая мультипроцессорность и кластеризацию, корректно сопоставима с UNIX-подобными системами. Не надо сразу обижаться за любимую Windows 9*x*/ME. Если отбросить эмоции и посмотреть правде в глаза, Windows 9*x*/ME — операционная система для домохозяек и игрушек, которая благодаря простоте освоения методом "научного тыка" незаконно, или правильней сказать безосновательно (с точки зрения системного администратора, безопасности, надежности, да и по целому ряду других причин), проникла в корпоративный мир. Это, конечно, понятно — пользователи знают Windows 9*x*/ME (дома стоит, в институте на ней работали) и пытаются опробованное решение применить в масштабах предприятия. Когда предприятие это 3—5 машин в локальной сети — еще терпимо, но когда на предприятии хотя бы пара десятков компьютеров, и, не дай бог, несколько сетей, которые надо объединить — администратор начинает жить на работе. Вирусы, зависания машин, непонятные эффекты, отсутствие в операционной системе необхо-

димого программного обеспечения для администрирования и управления сетями и рабочими местами — вот сотая часть того, с чем сталкивается администратор. А если в сети появляется сервер... Windows 9*x*/ME уже ничто не поможет. Ведь сервер — это и RAID-массив, и несколько процессоров, причем необязательно это процессоры от Intel. Windows 9*x*/ME на такую машину не поставишь.

Таким образом, корректно сравнивать (с учетом приведенных выше требований) можно только семейство UNIX и Windows NT/2000. Относительно Windows NT/2000 существует основанное на схожести интерфейса и названия с Windows 9*x*/ME заблуждение, что настроить Windows NT/2000 дело пяти минут, и после настройки все работает годами без вмешательства администратора. Внешнее сходство этих систем с Windows 9*x*/ME создает обманчивую иллюзию понимания там, где им и не пахнет, а увеличение нагрузки на сервер заставляет остро чувствовать программистскую поговорку "Памяти мало никогда не бывает". Поскольку книга эта об Linux, сравним семейство Windows с Linux.

Во-первых, что очень выгодно отличает Linux от Windows — ее *бесплатность*. За Windows 9*x*/ME по сегодняшним ценам придется уплатить около 150—200 долларов, а за Windows NT/2000 и того больше. Кроме того, для работы нужен и Microsoft Office, за стандартный вариант которого придется уплатить около 600 долларов, и, если надо еще что-то — продолжать платить и платить. Сегодня никого не удивляет, когда стоимость установленного программного обеспечения больше, чем стоимость самого компьютера. А если у вас несколько компьютеров — умножьте затраты на их количество. Вот и получается, что маленькая фирма с 5-ю компьютерами потратит 7—10 тыс. долларов только на программное обеспечение. Но это только начало. Политика Microsoft очень проста и действенна — раз в полгода-год выходит новая версия программного продукта, который все вольно или невольно вынуждены покупать, потому что партнеры присылают вам файлы в формате Excel 97, а ваш Excel 95 отказывается их понимать. В результате за всю жизнь компьютера (3—5 лет) только на программное обеспечение придется потратить порядка 2—5 тыс. долларов. С другой стороны, Linux обойдется в 5—15 долларов, за которые можно купить 2—3 компакт-диска, заполненных бесплатным, с открытым исходным кодом, программным обеспечением. Даже если скачивать дистрибутив Linux через Интернет — все равно не потратить больше 30 долларов (приблизительно столько стоит месяц неограниченного подключения к Интернету). И что характерно — с этого дистрибутива можно сколько угодно раз инсталлировать Linux *на абсолютно законных основаниях*. Можно возразить, что за потраченные на продукты Microsoft деньги пользователи получают поддержку сервис-центра Microsoft. Увы, это не так — нормальной поддержки на территории СНГ до недавнего времени не было, а звонить сейчас в Москву, например, из Беларуси или Владивостока и получать телефонную консультацию в течение пятнадцати-

двадцати минут весьма накладно. Поддержку же для Linux и ее программного обеспечения получить очень просто, нужно только знать, куда обращаться. Поскольку Linux — дитя Интернета, решение проблем надо искать там. Помимо Интернета, где находятся тысячи Web-сайтов, посвященных как Linux в целом, так и конкретному программному продукту для нее, существуют десятки групп новостей, а, помимо всего прочего, в дистрибутив входит более 15 тыс. страниц документации, описывающих все и вся. Есть правда одно неудобство — поскольку Linux разрабатывается и сопровождается людьми со всех стран мира, то и документация для него, в основном, на английском языке. Впрочем, это небольшая плата за обладание практически бесплатным программным обеспечением. Тем не менее, существует достаточно большой пласт литературы и на русском языке.

Во-вторых, Linux *способна функционировать* на множестве аппаратных платформ и *с минимальными требованиями к аппаратуре*. С Windows сложнее. Она функционирует только на процессорах Intel или их клонах, а по требованиям к аппаратуре превосходит Linux. И если Windows 9*x*/ME достаточно сносно работает на Pentium-166 с 64 Мбайт оперативной памяти, то для Windows NT/2000 требуется хотя бы Pentium II 350 МГц и 128, а лучше 256 Мбайт оперативной памяти.

По поводу дружественности, легкости в освоении и инсталляции. На сегодняшний день *установить Linux на абсолютно чистый диск сможет любой пользователь*, для этого нужно только взять соответствующий дистрибутив. Например, дистрибутив Red Hat Linux 7.1 все сделает сам (если, конечно, это нужно) — самостоятельно разобьет и отформатирует жесткий диск, настроит нужную раскладку языка и интерфейс (богатейший выбор из более чем ста языков: русский, украинский, белорусский в том числе), определит аппаратное обеспечение компьютера и настроит его на максимальную производительность. Установит необходимое программное обеспечение в зависимости от выбранного профиля компьютера (сервер, рабочая станция, ноутбук или выборочная установка), при этом ни в коей мере не ограничивая владельца в самостоятельной конфигурации. Что примечательно, установку Linux можно производить в текстовом (обычно так поступают опытные пользователи, или если слабая машина) или в графическом интерфейсе, с CD-ROM, жесткого диска или даже по сети, загрузив компьютер со специально изготовленной дискеты. При инсталляции можно указать Linux при старте сразу загружать графическую оболочку. Поэтому миф о сложности инсталляции можно считать не соответствующим действительности. С легкостью освоения, несомненно, похуже. Для грамотной работы в Linux необходимо иметь представление об операционной системе. К сожалению, Windows приучила пользователя щелкать мышкой и не думать. Плюс еще наш менталитет — "сами с усами", метод "тыка". В UNIX это не проходит. Там подход другой — прочитай, разберись и можешь быть уверен, что это функционирует в любой UNIX-подобной системе одним и тем же способом. Еще

нюанс — документация для Linux пишется в расчете на грамотного, способного размышлять человека. Это, разумеется, отпугивает пользователя, привыкшего руководствоваться инструкцией-комиксом, и порождает очередной миф о недружелюбности Linux. Однако приятно, что творцы документации считают тебя умным человеком, а не семилетним ребенком.

По части настройки операционной системы. Microsoft внедрила в свою операционную систему непродуманную идею — системный реестр. В результате получился монстрообразный (зачастую в 4—5 Мбайт) файл двоичного формата, от целостности которого зависит жизнеспособность операционной системы. Очевидно разработчики совсем забыли старое изречение "Не клади все яйца в одну корзину". Очень часто (по меньшей мере, в 30—40 % случаев) ошибки функционирования операционной системы связаны с повреждением файла реестра. Еще одна проблема настраиваемости системы — очень много настроек Windows не описаны в документации, и необходимо перерыть горы литературы, чтобы по крохам насобирать информацию о тонкой настройке системы. Есть, конечно, программное обеспечение, позволяющее тонко настроить Windows, но, как правило, оно не бесплатно. В Linux все более надежно и доступно. Практически все о настройке системы или программного обеспечения можно узнать из документации. Конфигурационные файлы обычно для каждой программы отдельные, и практически все имеют понятный текстовый формат с подробными комментариями. А настроить в Linux можно все, причем для каждого пользователя в системе отдельно.

О надежности. Конечно, семейство Windows — это не Windows 3.1*x* и даже не Windows 95, для которых ни дня не проходило без сбоя, но до надежности и живучести Linux (не говоря уже о проверенных десятилетиями UNIX) Windows еще далеко. О чем говорить, если во время демонстрации Биллом Гейтсом новых возможностей Windows 2000 (флагманского продукта!) операционная система дала сбой, выдав при этом на монитор "синий экран смерти".

Занимаемое место — в настоящее время, наверное, уже не совсем актуально, сколько операционная система занимает места на жестком диске — 500 Мбайт или Гбайт, но, все равно — чем меньше система, тем она быстрее и надежнее. Тут опять в лидерах Linux — ее можно установить на одну дискету 1,44 Мбайт. Вполне функциональный Интернет-сервер можно уместить в 80—150 Мбайт. С Windows 98, а уж тем более с Windows NT, такого сделать не удастся.

Реакция разработчиков на ошибки. До недавнего времени у Microsoft была достаточно интересная политика: извлекать прибыль даже из собственного брака. Для Windows NT было выпущено 6 сервис-паков, в которые вошли исправления тысяч мелких и крупных ошибок, и все эти сервис-паки продавались за деньги. Скорость исправления ошибок в большинстве случаев

достаточно мала. Надежность программ пропорциональна количеству человек, которое участвовало в процессе тестирования. У производителей закрытого коммерческого ПО процесс тестирования является, по большей части, внутренним. С открытыми программами, в частности с Linux и программным обеспечением для нее, дело обстоит гораздо проще. Практически у каждого проекта есть две ветки — стабильная и текущая. В стабильную входит код, который был проверен большим количеством пользователей в течение некоторого разумного времени. Текущая ветка содержит рабочую версию, которая может изменяться ежедневно, содержит все последние нововведения, но при этом не гарантирована от ошибок. Каждый для себя решает, чем пользоваться — стабильной веткой или нестабильной, но содержащей все нововведения. Поскольку процесс тестирования открытого ПО не имеет ограничений по времени, он продолжается все то время, что существует конкретное программное обеспечение. Более того, программист, имея на руках исходные тексты, может сам исправить ошибку, не дожидаясь, пока это сделают за него. Благодаря интернет-сообществу практически всегда ошибки, обнаруженные в программном обеспечении для Linux, исправляются в течение суток и тут же становятся доступными для скачивания из Интернета.

Единственное, в чем Windows пока превосходит Linux — это *в количестве и разнообразии прикладного программного обеспечения*. Тем не менее, в последние полтора-два года очень бурно пошел процесс переноса под Linux коммерческого программного обеспечения. Пожалуй, сейчас мало осталось направлений, для которых в Linux нет бесплатного или, на худой конец, платного программного обеспечения. Офисные программные комплексы, совместимые по форматам файлов с Microsoft Office есть, об интернет-приложениях и говорить нечего, базы данных, мультимедиа-приложения и т. д., и т. п. Конечно, есть и незанятые ниши — например, нет того изобилия приложений для многоцветной полиграфии, трехмерного моделирования и анимации, видеомонтажа или игр. Но давайте себя спросим — много ли людей занимаются видеомонтажом или анимацией? Наверное, даже не сотая часть процента компьютерных пользователей. А если на компьютере только играть — зачем, вообще, ПК? Есть ведь Sony Play Station, Microsoft Xbox.

И отдельного упоминания заслуживает *безопасность*. Нехорошо, когда чуть ли не каждую неделю по всему офису прокатывается эпидемия компьютерного вируса, который, ко всему, портит данные на жестом диске. Или кто-то удалил на вашем компьютере данные случайно. Или еще чего похуже. Ведь Windows 9*x*/ME не имеют абсолютно никакой политики безопасности, и навредить системе может практически каждый. На сегодняшний день для Windows существует более двадцати девяти тысяч вирусов или программ-троянских коней, причем каждый пятый несет в себе деструктивные функции. Из-за совместимости с Windows 9*x*/ME Windows NT/2000 тоже не минула чаша сия. Конечно, с безопасностью в Windows NT намного лучше, но, тем не менее, ее намного чаще взламывают через сеть, чем UNIX. Без прав

администратора пользователь может навредить только сам себе, не затрагивая других пользователей. В настоящее время существует около 40 (!) вирусов или троянских коней для Linux, причем реально опасных из них всего 2 или 3. И программы, через которые происходило проникновение троянских коней, давно уже избавлены от этого недостатка.

Подведем итог — почему выбирают Linux.

Почему выбирают Linux

Приведем ряд аргументов.

- Самая лучшая операционная система — UNIX. Linux — это современный UNIX, работающий практически на всех платформах.
- В отличие от большинства операционных систем дистрибутивы Linux бесплатны, их можно скачивать из Интернета.
- В стандартный дистрибутив Linux входят сотни программ, с помощью которых можно решить 95 % задач, решаемых с помощью компьютера.
- Исходный код всех программ под Linux открыт, при желании его можно модифицировать так, как нужно.
- На базе Linux достаточно легко создать очень надежные (99,99 %) центры данных с поддержкой кластерных конфигураций и высокой степенью масштабирования.
- Корпоративная intranet-сеть "из коробки", элементарная установка интернет-сервисов и серверов, практически сразу настроенных для стандартного применения.
- Высокая степень безопасности и ограничения доступа к ресурсам и данным системы.
- Большое количество поддерживаемых Linux аппаратных платформ.
- Графический интерфейс с десятками оконных менеджеров, позволяющих создать эксклюзивную графическую среду, точно настроенную для нужд пользователя и аппаратных ресурсов.
- Относительно малые требования к аппаратным ресурсам, достаточно новый дистрибутив вполне можно установить на старших 486-х компьютерах.
- Огромнейшая библиотека документации, ежедневно улучшающаяся и дополняющаяся.
- Великолепная поддержка программного обеспечения, ответы практически на любой вопрос можно найти в Интернете, а на оставшиеся вопросы можно получить ответ у самих разработчиков, которые не скрываются за копирайтом большой фирмы.

- В Linux можно настроить все и вся. Простота конфигурации и подробное описание конфигурационных файлов выгодно отличают Linux от большинства коммерческих операционных систем.
- Можно инсталлировать Linux на одну дискету, и при этом она окажется способна выполнять функции маршрутизатора или отправлять электронную почту.
- Постоянное обновление и улучшение как ядра Linux, так и большинства программных продуктов для Linux
- Отсутствие зависимости от патентов и лицензий.

Разные факты

Фирма Intel сотрудничает с основными разработчиками ядра Linux, а IBM вкладывает около миллиарда долларов в продвижение и поддержку Linux в своих офисах. Версия Red Hat Linux 4.1 использовалась для создания спецэффектов при съемках фильма "Титаник". Правительство Китая утвердило Linux как операционную систему для государственных структур. Кинокомпания Dream Works, создатель мультфильмов "Побег из курятника", "Принц Египта", "Шрек" и "Антс" полностью переводит весь цикл разработки мультфильмов на Linux. Компания Corel заявила о создании версий своего программного обеспечения для Linux. По заказу Национального научного фонда США (NSF) будет создан TeraGrid — самый мощный суперкомпьютер, используемый в научных целях. В TeraGrid будут применяться кластерные серверы IBM, работающие под управлением ОС Linux и связанные между собой высокоскоростной оптической сетью Qwest.

Ссылки

Сайты, посвященные BeOS:

- **www.benews.ru** — новости мира BeOS на русском языке;
- **www.bebits.com** — крупнейший ресурс программного обеспечения для BeOS;
- **besoft.org** — программное обеспечение и документация для BeOS.

Сайты, посвященные QNX:

- **www.qnx.com** — сайт фирмы QNX Software Systems, разработчика QNX;
- **qnx.boom.ru** — программы, документация, новости QNX;
- **qnxworld.main.ru** — еще один сайт по QNX;
- **www.crosswinds.net/~kthulu/russian** — много документации по QNX.

Сайты, посвященные FreeBSD:

- **www.freebsd.org** — сайт FreeBSD;
- **www.freebsd.ru** — русскоязычный сайт.

Сайты, посвященные Linux:

- **www.linux.org.ru** — отличный сайт о Linux;
- **www.linux.org** — сайт о Linux;
- **www.linuxdocs.org** — много литературы о Linux;
- **www.linuxrsp.ru** — русскоязычный сайт;
- **www.redhat.com** — сайт версии Red Hat.

Сайты, посвященные Windows:

- **www.microsoft.com** — официальный сайт фирмы Microsoft;
- **www.winfiles.com** — обширная коллекция программ для Windows.

Сайты, посвященные Apple:

- **www.apple.com** — официальный сайт Apple;
- **www.apple.ru** — русскоязычный сайт Apple.

Глава 2

Возможности Linux

В этой главе пойдет разговор об администраторах, офисном и домашнем применении Linux. Автор достаточно долго занимался сопровождением как программ, так и локальных сетей и компьютеров, поэтому не понаслышке знает проблемы администратора. Администратор — это человек, который во время рабочего дня ничего не делает, пьет кофе и играет в компьютерные игры. В идеале, администратору платят зарплату за то, что он бездельничает. В том смысле, что надежное и не весьма требовательное к сопровождению программное обеспечение (включая, разумеется, и операционную систему), будучи один раз правильно отлажено, должно потом долго работать, не требуя постоянного вмешательства администратора для дополнительных перенастроек, переналадок и инсталляций. Если в вашей организации это не так, вывод может быть один: либо у вас неудачное программное обеспечение, либо плохой администратор. Поэтому всех, кто отвечает за бесперебойную работу локальных сетей, весьма интересует, как проблема администрирования решена в той или иной операционной системе. И если решена она недостаточно хорошо, заставляя делать изо дня в день одно и то же, такая операционная система вызывает раздражение и желание сменить ее на более "дружелюбную". Офисное применение Linux интересует нас с точки зрения применимости ее на рабочем месте, в фирме, на предприятии. Домашнее применение тоже, разумеется, будет рассмотрено. О серверном использовании Linux в этой главе мы подробно говорить не будем, потому что применению Linux в качестве серверной ОС посвящена добрая половина книги. Помимо этого, практически все знают, что Linux и сервер — "близнецы-братья", а о применении ее в офисе или дома еще мало кто задумывался. Но начнем с азов. Как выразился один из грандов компьютерного бизнеса "Компьютер — это сеть".

Сеть

Сетевые протоколы и аппаратура

Linux по умолчанию работает со своим "родным" протоколом TCP/IP, протоколом, на котором функционирует Интернет. Но это вовсе не означает,

что она, кроме этого протокола, ничего не понимает. При установке соответствующего программного обеспечения Linux способна также работать с протоколом IPX/SPX фирмы Novell Netware, протоколом NetBIOS (Microsoft Windows 3.1*x*, Windows 9*x*/ME, Windows NT/2000) и AppleTalk (Apple Mac OS). И это еще не все, что она понимает и поддерживает, хотя перечисленные четыре сетевых протокола сегодня используются, наверное, более чем в 95% случаев. Из аппаратных сетевых средств Linux способна работать практически с любым оборудованием, предназначенным в том или ином виде для использования в сетевых соединениях: сетевые карты Ethernet, Radio Ethernet, ArcNet, аппаратура для спутникового Интернета, ISDN, ATM, обычные модемы и многое другое. Конечно, с аппаратным обеспечением не все так гладко, как хотелось бы. Не для всех устройств есть драйвера под Linux, однако, как правило, для всех распространенных устройств они есть. Были проблемы с драйверами для так называемых Win-модемов, но в последнее время решаются и они. Можно сказать, что при наличии соответствующего программного обеспечения и драйверов сетевые протоколы и аппаратура под Linux очень хорошо настраиваются с помощью текстовых конфигурационных файлов или специальными программами, например, netconf.

Замечание

С написанием названий программ ситуация двойственная — в UNIX (и, соответственно, в Linux) регистр символов имеет значение, и поэтому названия программ в командной строке необходимо набирать правильно. Традиционно системные утилиты пишутся исключительно строчными, "маленькими" буквами, хотя в документации к ним же можно увидеть, что некоторые имена содержат и прописные, "большие" буквы. Такая двойственность в ряде случаев имеет место и в этой книге.

Сетевые сервисы

О сетевых сервисах более подробно будет рассказано в пятой части книги, а сейчас — краткий обзор. Начнем с того, где зарождалась и развивалась Linux — с Интернета. Было бы удивительно, если бы дитя Интернета и представитель семейства UNIX (колыбели Интернета) не предоставлял всей полноты интернет-сервисов. Что интересует пользователя в Интернете? На первый, поверхностный, взгляд Web-сайты, FTP, электронная почта и новости. Но для нормального (и комфортного) функционирования Интернета необходимо множество других сервисов — это и DNS, и прокси-серверы, и серверы точного времени и многое-многое другое. Все это для Linux есть, и не в единственном экземпляре — нужно только выбрать, какой "тяжести" инструмент необходим. Сказанное касается и серверного программного обеспечения, и клиентского. Так, например, Web-браузеров существует более десяти: Lynx — текстовый браузер, Netscape Navigator, Mozilla, Opera, Konqueror и др.

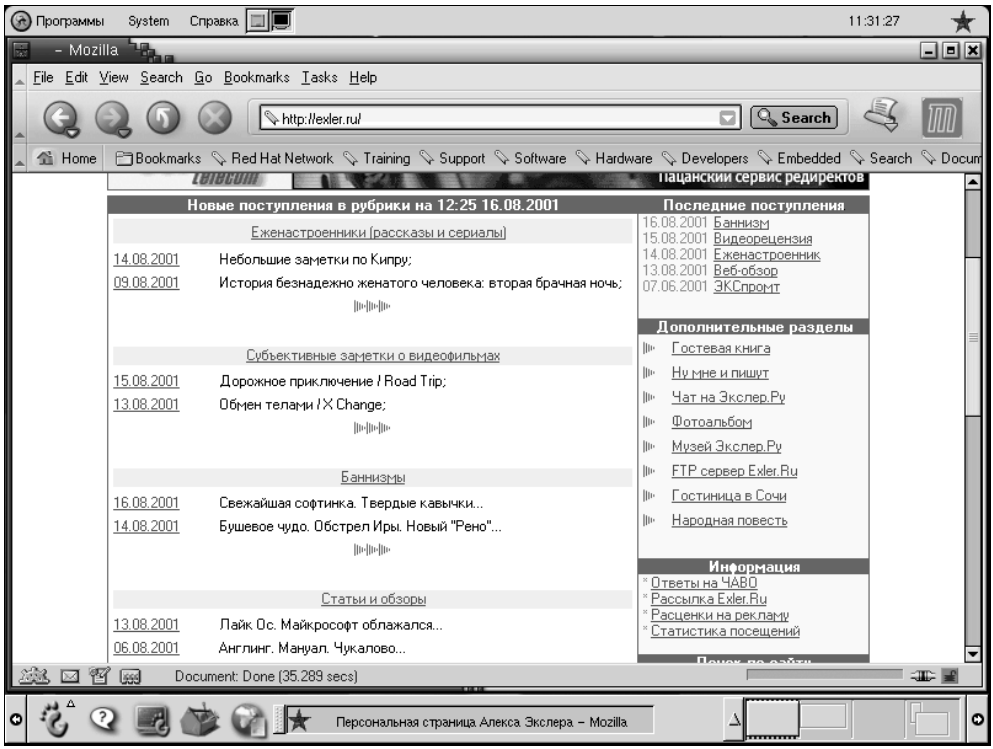

Рис. 2.1. Web-браузер Mozilla

Почтовых клиентов также существует несколько десятков — как текстовых, так и графических: Pine, Netscape-клиент, Kmail, Evolution и т. д.

Можно рассматривать любой интернет-сервис, и всегда в список клиентских приложений для этого сервиса войдет не менее десятка программ. Если необходим файл-сервер — тоже есть большой выбор. Можно пользоваться "родным" NFS, можно Mars — файл-сервером для сетей Netware, можно Samba — файл-сервером для сетей Microsoft. Для всех упомянутых типов файловых серверов, конечно же, есть и клиентское программное обеспечение. При желании можно создать сетевой компьютер — с отсутствующим жестким диском, без каких-либо накопителей, загружающийся через сеть и нормально функционирующий (причем, с графической оболочкой). Решены для Linux и вопросы статистики. Множество пакетов могут собрать, обработать, представить в текстовом и графическом виде информацию о любой стороне функционирования Linux, в частности, о загрузке сети, входящем и исходящем трафике, построить диаграммы, отобразить их на Web-странице и, если необходимо, адекватно отреагировать на какое-то отклонение в функционировании сети. Настройку множества сервисов можно произвести или с помощью специальных программ — например, linuxconf, или отредак-

тировав конфигурационные файлы. У большинства сервисов есть еще одна возможность — настройка через Web-интерфейс. Существуют и совместимые с ICQ интернет-пейджеры: licq, kicq, GNOMEICQ, micq, в том числе, и для текстовой консоли.

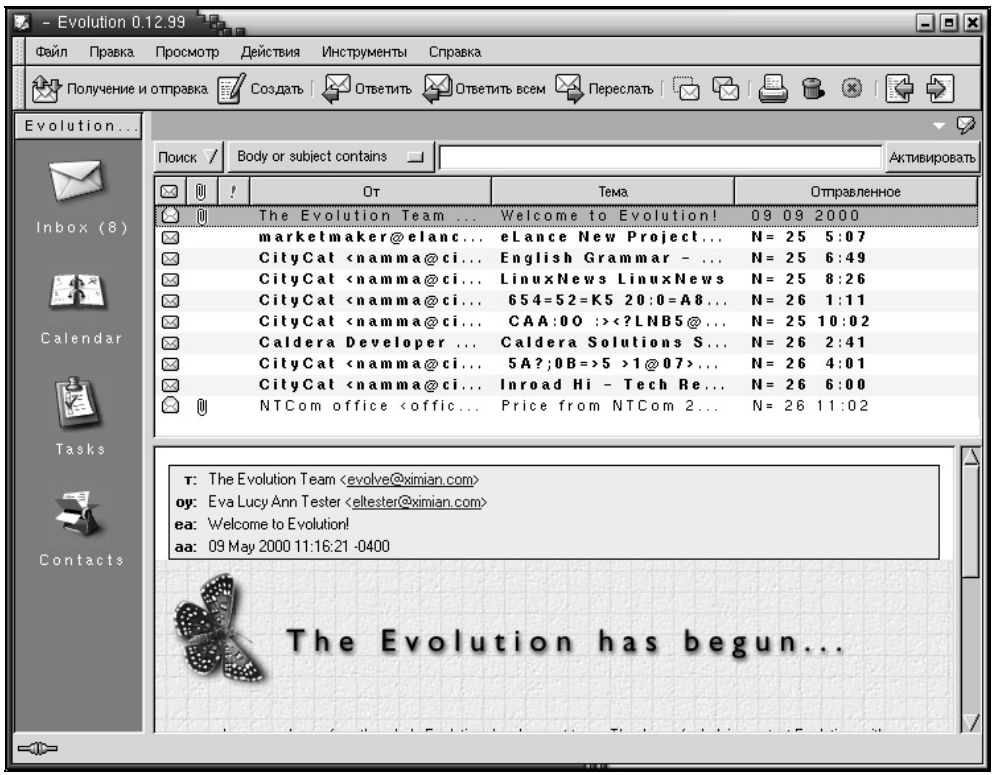

Рис. 2.2. Почтовый клиент Evolution, по совместительству — органайзер

Файловые менеджеры

Для пользователей старой закалки, знакомых еще с DOS, непременным атрибутом работы за компьютером был файловый менеджер, который подменял собой скуку командной строки и черноту экрана. Хотя адепты Linux упорно твердят о полной ненужности файлового менеджера для Linux, тем не менее, спрос порождает предложение. Есть несколько файловых менеджеров и для нашей операционной системы. Как обычно, есть они и для текстовой консоли, и для X Window. Самый известный и, наверное, один из старейших текстовых файловых менеджеров — Midnight Commander (почти полный эквивалент Norton Commander).

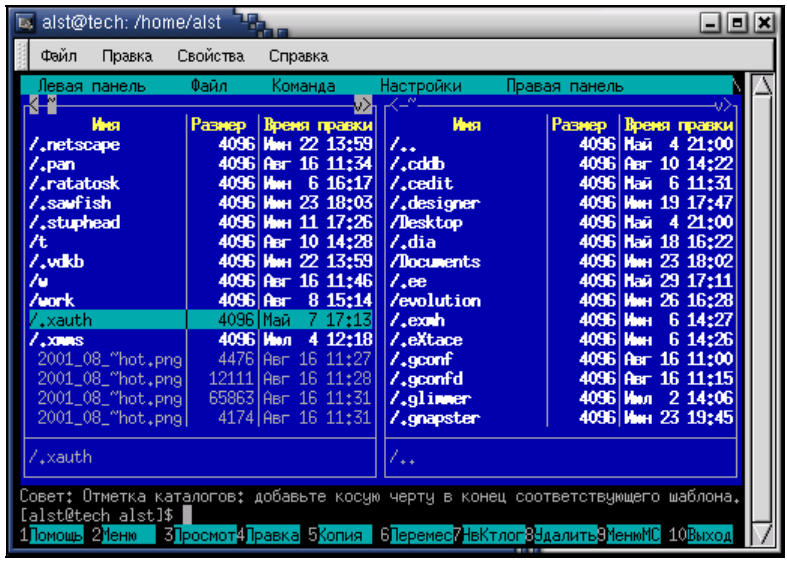

Рис. 2.3. Файловый менеджер Midnight Commander

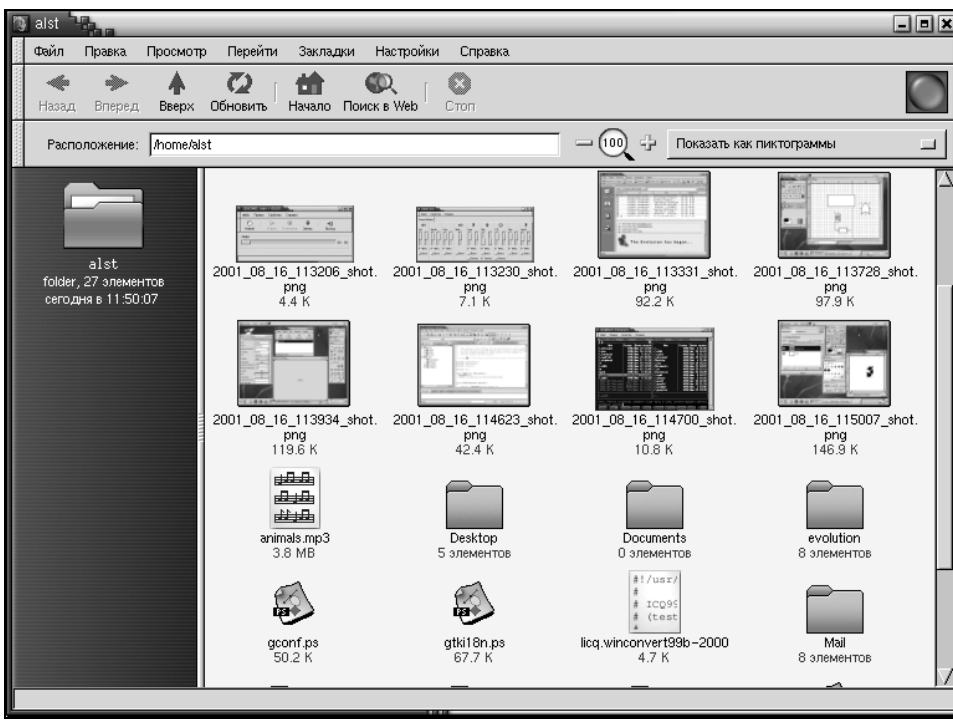

Рис. 2.4. Один из графических файловых менеджеров

Позволяет работать с файлами, редактировать их, осуществлять поиск, работать с архивами и т. д. Однако есть еще и другие достоинства — способность устанавливать пакеты RPM, работать с FTP, просматривать HTML-документы, подключаться к сетевым дискам. Приблизительно такого же плана текстовый файловый менеджер XNC. Помимо текстовых, достаточно много и графических файловых менеджеров для X Window, например, Kcommander или Kruiser.

Текстовые редакторы

Тут выбор широчайший — от простейшего строчного текстового редактора до пакетов, которые текстовым редактором и назвать трудно. И такое разнообразие наблюдается и для текстовой консоли, и для X Window. Конечно, в сегодняшний век торжества "графики" многие удивляются наличию большого числа текстовых консольных редакторов. Однако надо вспомнить о широкой распространенности Linux, в том числе и на не очень мощных машинах, куда не имеет смысла устанавливать объемную графическую оболочку и не менее объемный графический текстовый редактор только для того, чтобы откорректировать несколько конфигурационных файлов. Не исключена также необходимость отредактировать тот или иной Web-скрипт на удаленной машине через Интернет. Или совсем неприятный вариант — сбой системы, не позволяющий загрузиться в графическом режиме. Поэтому до сих пор существуют текстовый редактор vi, появившийся в самом начале становления UNIX, или его более функциональные потомки vim, joe, pico, jed, встроенный редактор Midnight Commander, EMACS и система верстки (ее тоже можно отнести к текстовым редакторам) TeX.

Под X Window еще больше редакторов. Очень много простых, типа Windows Notepad, и, конечно, много мощных текстовых процессоров, часть из которых входит в офисные пакеты. В качестве примера можно привести Kedit, Gedit, Kwrite, Kword, Ted, Abiword, StarWord и др. Более подробно о редакторах будет сказано ниже.

Графические оболочки

Неоднократно опровергаемое утверждение, что Linux — чисто текстовая среда, почему-то очень живуче. Хотя по разнообразию графических оболочек (или менеджеров окон) он оставляет далеко позади семейство Windows, да и большинство UNIX-собратьев. В отличие от Windows, в Linux (UNIX) графическая оболочка (X Window) разделена на два приложения: X-сервер и менеджер окон. Сервер в какой-то мере специфичен для аппаратных средств (зависит от видеокарты, шины данных и т. д.) и выполняет роль рабочей лошадки, а менеджер окон обеспечивает внешний вид приложений, отрисовку окон, меню и прочих элементов графического интерфейса. Благодаря такой независимо-

сти пользователь получает богатейший выбор средств для персонализации своего рабочего места. Можно поставить IceWM или AfterStep и получить легкую и мощную графическую среду (вполне нормально функционирующую на старших 486-х процессорах), для тех, кому надо "как в Windows" — FVWM95, для тех кому "как в Windows, но лучше" — KDE или GNOME.

Рис. 2.5. GNOME, Win4Lin, OpenOffice, Licq

И это далеко не предел — менеджеров окон (только самых известных) существует десятка полтора, и все они легко настраиваются по всем своим параметрам. Конечно, неопытного пользователя очень смущает текстовая консоль, но можно при инсталляции Linux (или позже) установить загрузку X Window сразу при старте системы. Тем более, что практически все текстовые программы или дублируются графическими, или имеют графический интерфейс.

Графические редакторы

В этой категории тоже достаточно много программ. От самых простых до очень сложных, ничем не уступающих по возможностям CorelDRAW и

Photoshop. Как обычно — редакторы есть векторные и растровые. Для примера Gimp — мощнейший редактор, перенесенный, в частности, под Windows, StarDraw — программа создания рисунков на основе векторной графики, StarImage — программа создания рисунков на основе битовых образов, KimageShop и множество других.

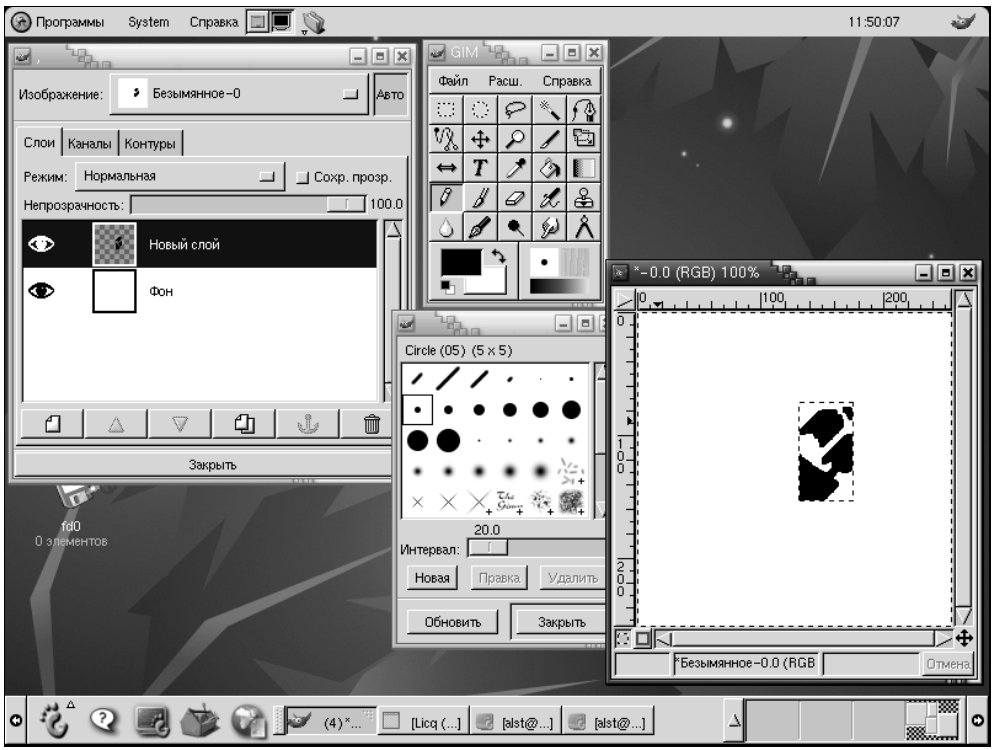

Рис. 2.6. Графический редактор Gimp

Web-инструментарий

Традиционно лучшим редактором для Web-дизайнера считается простой текстовый редактор, однако достаточно много людей работают в специализированных HTML-редакторах. Для Linux, однако, выбор HTML-редакторов не очень большой. К примеру, программа подготовки HTML-файлов StarWriter/Web, WebMaker (разработка Алексея Дець, Россия) или Quanta Plus (разработка Дмитрия Поплавского и Александра Яковлева, Украина).

Офисные пакеты

Так исторически сложилось, что разработкой полноценного офисного пакета для Linux сообщество озаботилось только полтора-два года назад. По всей видимости, это связано с тем, что только сейчас Linux стала продвигаться на офисные рабочие места, оставаясь до последнего времени серверной операционной системой или системой, установленной на домашнем компьютере. Конечно, и до этого существовали текстовые редакторы, электронные таблицы, органайзеры и программы презентаций. Однако в полноценный офисный пакет они не складывались из-за ряда нерешенных проблем, в т. ч. несовместимости с Microsoft Office, отсутствия тесной интеграции программ от разных разработчиков и отсутствия одного разработчика, способного создать все составные части пакета, а также отсутствия полноценной поддержки русского языка.

Под офисным пакетом будем понимать набор программ, включающих в себя:

- текстовый редактор (процессор);
- программу для работы с электронными таблицами;
- программу обработки электронной почты (в принципе необязательно);
- программу подготовки презентаций;
- программу для работы с изображениями (или несколько);
- персональный органайзер;
- программу для организации работы в группе

и т. д.

Офисный пакет может включать и другие программы или, наоборот, некоторые из упомянутых программ могут не входить в общий пакет. Но комплект программ можно назвать пакетом только тогда, когда все входящие в него программы обладают единым интерфейсом и позволяют обмениваться информацией между собой.

При оценке офисных пакетов мы вынуждены сравнивать их с Microsoft Office, поскольку на сегодняшний день подавляющая часть пользователей, так или иначе, работает с этим пакетом. Поэтому при выборе программ, которые можно отнести к разряду офисных, обязательно подразумевается совместимость по форматам файлов с Microsoft Office. Даже если пакет полностью работает под Linux, рано или поздно возникнет необходимость отправить партнерам файл в формате Microsoft Office или наоборот, получить от них такого рода файл. И никому не будет дела до того, что в вашей фирме не признают программное обеспечение от Microsoft.

Поэтому рассмотрим офисные пакеты под Linux с учетом приведенных выше требований.

В настоящее время существует достаточно много офисных пакетов как платных, так и с открытым исходным кодом. Начнем с платных пакетов.

Applixware компании Applix — судя по отзывам в Интернет, очень неплохой офисный пакет, работает стабильно и значительно быстрее, чем описанный ниже StarOffice версии 5.2. WordPerfect Office 2000 — в отличие от Applixware, менее надежен, работает помедленней и, что самое неприятное, наблюдаются проблемы с русским языком.

StarOffice 5.2

Пакет StarOffice разработан немецкой фирмой Star Division, в последующем был куплен фирмой Sun Microsystems. В настоящее время пакет StarOffice 5.2 бесплатен, фирма Sun Microsystems открыла его исходный код и разрабатывает на его базе StarOffice 6.

В состав пакета входят (при инсталляции можно отказаться от установки некоторых частей пакета):

- текстовый процессор StarWriter;
- программа подготовки HTML-файлов StarWriter/Web;
- программа работы с электронными таблицами StarCalc;
- программа подготовки презентаций StarImpress;
- программа создания рисунков на основе векторной графики StarDraw;
- программа создания рисунков на основе битовых образов StarImage;
- система управления базами данных StarBase;
- почтовая программа StarMail;
- StarDiscussion;
- StarChart;
- StarMath;
- StarSchedule;
- StarDesktop — основная оболочка, через которую организуется работа остальных частей пакета, и которая может полностью заменить интегрированную оболочку, такую как KDE или GNOME.

Непривычная для нас идеология — основная оболочка, из которой происходит запуск остальных приложений, с одной стороны, создает тесную интеграцию частей офисного пакета, но с другой — достаточно сильно замедляет работу системы и потребляет много ресурсов. Кроме того, редкий пользователь работает сразу со всеми приложениями пакета. Учитывая это, в 6-й версии и в OpenOffice отказались от основной оболочки, что в лучшую сторону сказалось на производительности и ресурсоемкости.

И, что особенно приятно, выпущена *русифицированная* версия StarOffice 5.2. Поскольку на сегодня это чуть ли не единственный приемлемый вариант офисного пакета, рассмотрим его подробнее.

Установка программ проходит без всяких проблем. Требования к компьютеру:

- процессор Pentium или выше;
- 64 Мбайт ОЗУ;
- не менее 180 Мбайт свободного места на жестком диске (в зависимости от типа инсталляции может потребоваться до 250 Мбайт);
- монитор VGA или выше с 256 цветами и разрешением не ниже 800×600;
- CD-ROM (это требование относится к тем, кто устанавливает StarOffice с CD-ROM);
- ядро Linux версии 2.0.*x* или выше;
- должна быть установлена система X Window с одним из оконных менеджеров;
- должна быть установлена библиотека GLibc версии 2.1.1 или выше.

На сервере Sun находится очень хорошая инструкция на русском языке по установке, настройке и решению возможных проблем. К сожалению, файл помощи пока не русифицирован. Совместимость с Microsoft Office удовлетворительная, однако могут возникать проблемы с таблицами и со связанными файлами (например, файл Excel, внедренный в файл Word).

OpenOffice

Проект базирующийся на исходном коде StarOffice. На данный момент содержит следующие приложения:

- OpenCalc — электронные таблицы;
- OpenDraw — графический редактор;
- OpenWriter — текстовый редактор;
- Impress — программа презентации.

Koffice

Очень динамично развивающийся пакет. Является частью проекта KDE. В состав входят:

- KSpread — электронные таблицы;
- KPresenter — создание презентаций;
- KChart — создание диаграмм;
- Krayon — растровый графический редактор;

- Kontour — векторный графический редактор;
- KFormula — математический пакет;
- KWord — WYSIWYG-текстовый редактор;
- KOrganizer — органайзер;
- Kivio — программа создания диаграмм;
- Kugar — инструмент для генерации бизнес-отчетов;
- Kplato — программа для планирования и управления проектами.

Помимо выдержанного в стиле KDE-интерфейса, отличной интеграции с другими KDE-приложениями и нормальной поддержкой русского языка, что немаловажно, заявлена совместимость с файлами Microsoft Office, а также возможность обработки файлов в форматах CSV, RTF. Очень простая инсталляция. Достаточно скромные требования к ресурсам.

GNOME Workshop

Еще один офисный пакет от создателей GNOME. В него входят следующие программы:

- AbiWord — популярный мультиплатформенный текстовый редактор;
- Achtung — программа презентаций;
- Balsa — мощный почтовый клиент;
- Dia — отличное приложение для создания различных диаграмм, аналог Microsoft Visio;
- Eye of GNOME — программа просмотра графических изображений;
- Evolution — мощная программа, аналог Microsoft Outlook;
- Galeon — быстрый Web-браузер;
- Gfax — программа для приема и отправления факсов;
- GIMP — великолепный графический редактор;
- GNOME-DB — средство для работы с БД;
- Gnucash — персональный финансовый менеджер;
- Gnumeric — электронные таблицы;
- Guppi — программа для рисования;
- MrProject — инструмент управления проектами;
- Sketch — редактор векторной графики;
- Sodipodi — редактор векторной графики;
- Toutdoux — инструмент управления проектами.

Со временем разработчики обещают тесную интеграцию пакета с OpenOffice.

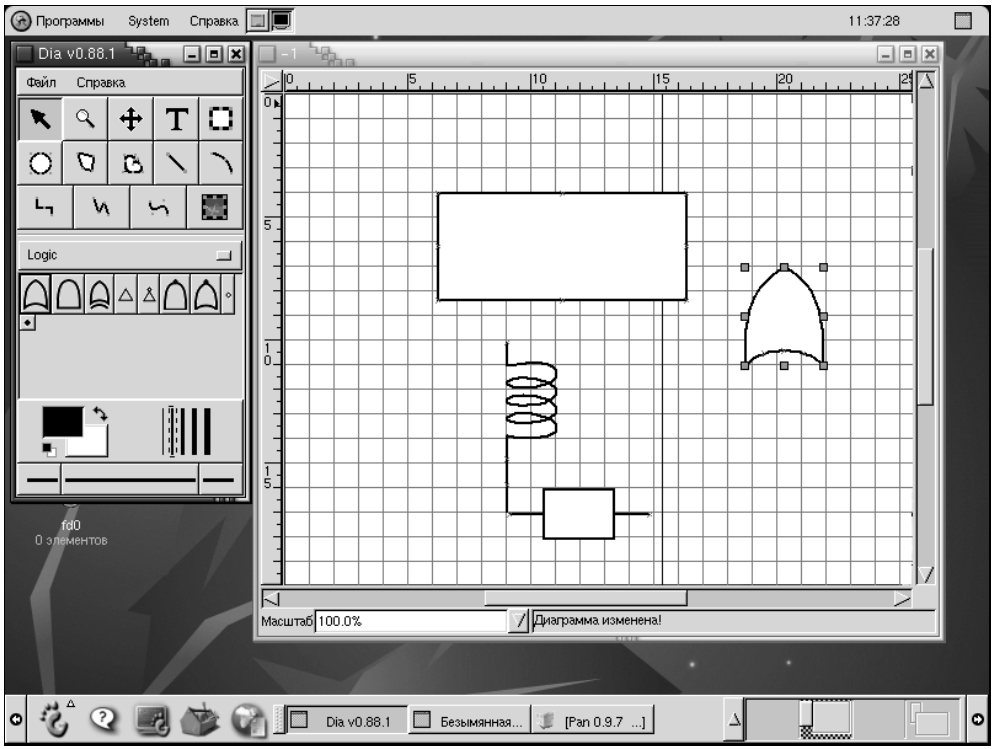

Рис. 2.7. Программа для создания диаграмм — Dia

Базы данных

Вопреки распространенному мнению, под Linux разработано и перенесено большое количество серверов данных — от настольных до уровня предприятия. В их числе IBM DB2, Informix, Oracle, Sybase SQL Anywhere, Interbase, PostgreeSQL, MySQL. Причем последние три — бесплатные, с открытым исходным кодом. Остальные, при определенных условиях, можно получить бесплатно или пользоваться бесплатно определенное время.

Эмуляторы Windows

Существует немало эмуляторов Windows в среде Linux: Citrix MetaFrame, Mainsoft's MainWin, Win4Lin, VMWare, WINE и др. Они различны по функциональным возможностям: одни обеспечивают работу приложений для Windows 9*x*, другие способны запускать еще и продукты для Windows NT/2000. Есть здесь и сложность — отсутствие поддержки DirectX. Пре-

красно запускаются под Linux игры, разработанные в расчете на OpenGL, но большинство самых современных игр, которые выпускаются в расчете на DirectX, пока не работают под эмуляторами. Однако, по заявлениям разработчиков, эта проблема может быть вскоре преодолена. Особняком стоит VMWare. Это не эмулятор Windows, это эмулятор компьютера! Благодаря чему под VMWare работает практически любая программа, но взамен потребляется много ресурсов.

Средства разработки программ

Для Linux, как и для UNIX, "родным" языком является, естественно, C/C++, но это совершенно не означает, что кроме них, никаких других компиляторов (или интерпретаторов) языков не существует. Большего разнообразия языков на одной платформе встретить невозможно. Настоящее вавилонское смешение! Трудно найти какой-либо язык, компилятора или интерпретатора которого не существует для Linux: C/C++, Pascal, Perl, Java, Lisp, Rexx, Fortran и т. д., и т. п. Не обойдены стороной и интегрированные среды разработки. Событием стал выпуск фирмой Borland интегрированной среды Kylix — Linux-аналога Delphi (Windows).

Kylix

Приложения, написанные в Delphi 6 с использованием специальной библиотеки, можно практически без переделок перенести в Linux. Наряду с коммерческой версией Kylix существует и версия для разработки программного обеспечения с открытым исходным кодом, скачать которую можно бесплатно с Web-сайта фирмы Borland. Разработчики обещают обеспечить совместимость и с C Builder. Впрочем, и здесь есть своя ложка дегтя. Во-первых, при работе Kylix использует эмулятор Windows — Wine. Это понятно, программисты из Borland облегчили себе перенос Delphi в Linux, но поскольку Wine — программа, не до конца реализовавшая в себе Windows-совместимость и постоянно модернизируемая, Kylix временами работает нестабильно. И во-вторых, совместно с вновь созданным в Kylix приложением необходимо распространять некоторые специфические библиотеки.

KDevelop

Программа предназначена для разработки приложений под KDE с использованием библиотеки Qt. Можно разрабатывать консольные приложения. Обладает интерфейсом, похожим на MS Visual C++. Требует много сторонних приложений типа a2ps, Khexedit, KTranslator и т. д. Встроен достаточно удобный интерактивный отладчик.

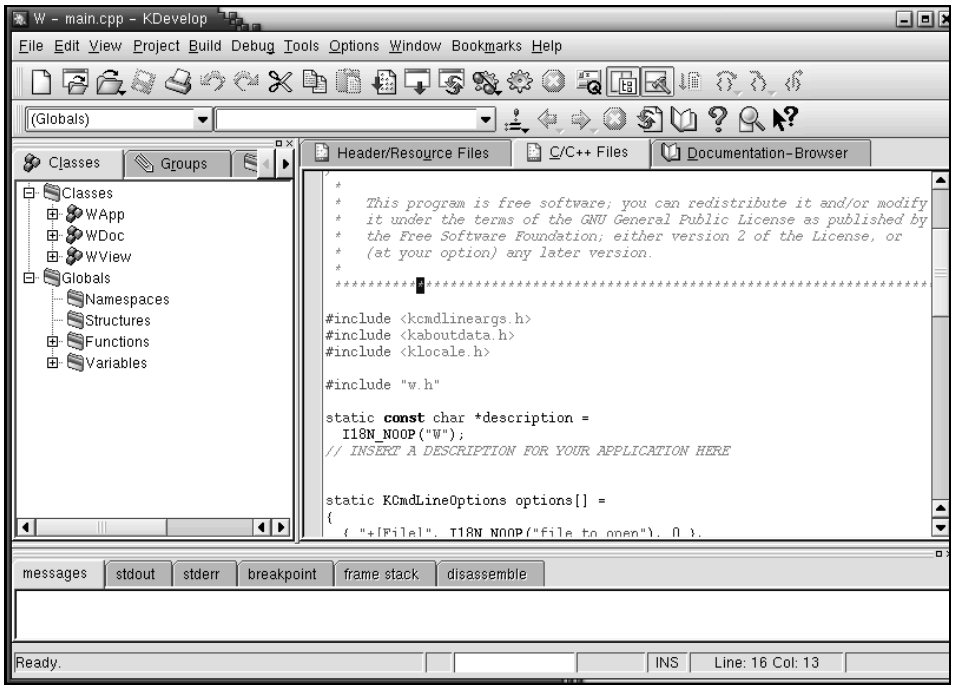

Рис. 2.8. Интегрированная оболочка разработки программного обеспечения KDevelop

Glade

Визуальная среда для разработки приложений для GNOME. Достаточно неплоха, поддерживает несколько языков программирования. По сравнению с коммерческими средами оставляет ощущение незавершенности. Входит в состав GNOME.

VDK Builder

По интерфейсу напоминает Borland Delphi/Borland C Builder, является развитием Glade. Позволяет разрабатывать приложения для GNOME. VDK — классы, позволяющие программисту получать GNOME-интерфейс, применяя только C++. Есть возможность создания консольных приложений. Нет полноценной системы справки.

Motor

Редактор с подсветкой синтаксиса, менеджер проектов, генератор makefile, интегрирован с gcc и gdb. Поддерживает CVS. Умеет генерировать проекты из шаблонов. Полезная возможность — генерация RPM из проекта.

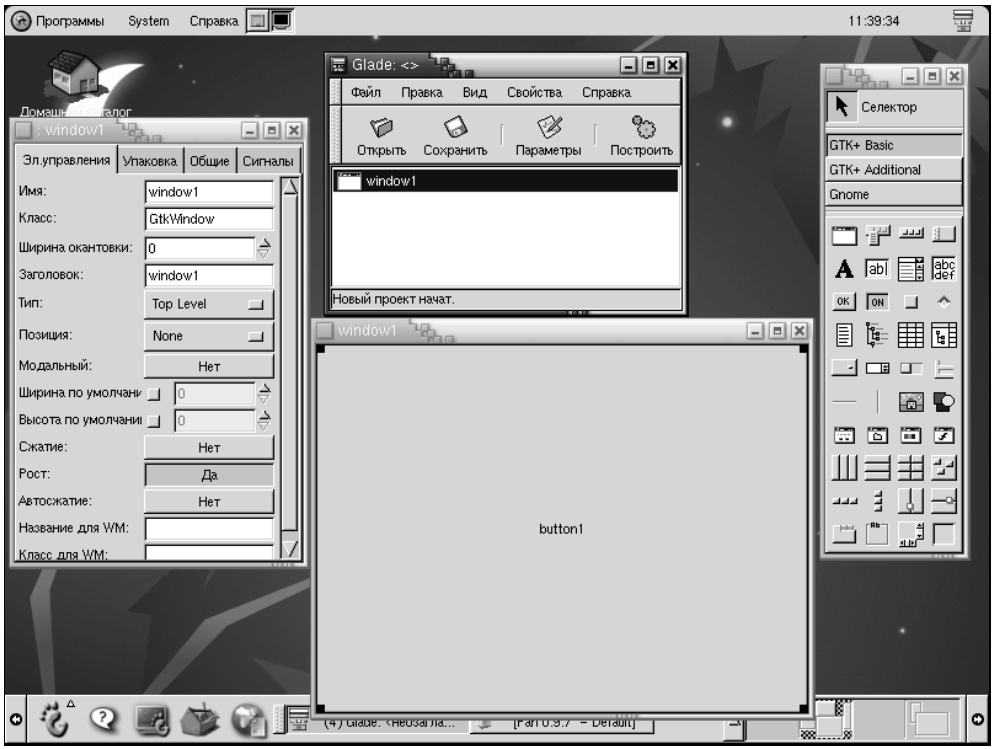

Рис. 2.9. Визуальная среда для разработки приложений Glade

Rhide

Написана на перенесенной из DOS библиотеке Turbo Vision. Поддерживает C, C++, Assembler, Pascal и Fortran. Оболочка для gdb выделена в отдельное приложение, благодаря этому ее можно использовать как Turbo Debugger. Можно настраивать цвета, компилятор и его опции, языки.

SNiFF+ Penguin IDE

Качественный продукт, основан на коммерческой разработке. Включает в себя анализатор кода. Просмотр кода и навигация по коду производятся по дереву проекта. Поддерживает CVS и RCS. Имеет встроенный редактор документации, можно связывать пункты документации и исходный код. Сгенерированная документация для дальнейшей публикации может конвертироваться в различные форматы. Поддерживает командную разработку. Возможно использование внешних редакторов. Есть версии для Windows NT/2000, Linux, Solaris и других операционных систем.

Code Forge

Платное программное обеспечение. В этой среде можно программировать на значительном количестве языков (C, C++, Assembler, Perl, GTK, TCL, TK, Tkl++, Python, Java, Basic, Fortran, Prolog, SGML, HTML и др.). Имеет настраиваемую подсветку синтаксиса, позволяет использовать любой компилятор (для C++ предлагает три разных), любые отладчики, вести версии, создавать документацию. Однако не имеет визуальных средств.

CodeWarrior

Профессиональная, мощная среда разработчика IDE, интегрированная с EGCS/GNU. Редактор поддерживает подсветку синтаксиса, многооконность, быстрый доступ к функциям и многое другое. Имеется менеджер проектов с настройкой компилятора, линкера, отладчика и редактора. Можно использовать внешний редактор. Умеет запускать скрипты на этапах компиляции и линковки. Хорошо документирована. Кроме версии для Linux, имеются версии для Java, Mac OS, Windows и Solaris. Также указывается, что есть инструменты разработчика для PlayStation, Palm OS, PowerPC, MIPS.

CRiSP

Работает как в X Window, так и в консоли. По внешнему виду напоминает HomeSite. Редактор поддерживает настраиваемую подсветку синтаксиса, работу с тэгами, многооконность. Имеется набор шаблонов языковых конструкций для Ada, C, C++, SQL, HTML, Latex и других. Встроен клиент FTP. Есть версии для Windows, BSD, SGI.

Как видите, выбор обширен, и всегда можно найти продукт, удовлетворяющий самому требовательному вкусу.

Мультимедиа-приложения

Аудио

Звуковые средства должны воспроизводить, как минимум, WAV- и MIDI-файлы, MPEG-3, а также обычные аудио-CD.

Времена сложного ручного конфигурирования этих устройств (достаточно подробно описанные в литературе), похоже, закончились. По крайней мере, в RedHat и его клонах поддержка звука предполагается по умолчанию. Поддерживаются почти все мало-мальски распространенные устройства. В том числе дешевые ISA- и PCI-карты. Обычно после инсталляции дистрибутива звуковая карта уже сконфигурирована и вполне работоспособна. Впрочем,

иногда все же ее приходится настраивать. Для этого достаточно запустить в командной строке программу sndconfig. Она проведет тестирование звукового устройства и в случае благоприятного результата выдаст примеры WAV- и MIDI-звуков.

С аудиодисками также все просто. В состав KDE входит вполне нормальный (и несложный в использовании CD-плейер), аналогичный таковому из комплекта Windows. Кроме того, имеется еще несколько похожих средств как графических, так и консольных, например, несколько проигрывателей входят в состав GNOME.

Для управления звуком, как и в других операционных средах, используется микшер. Микшеров под Linux также очень много, существуют микшеры консольные и графические. Для примера, в составе KDE и GNOME имеется микшер, позволяющий регулировать громкость и баланс при воспроизведении звуков разного типа.

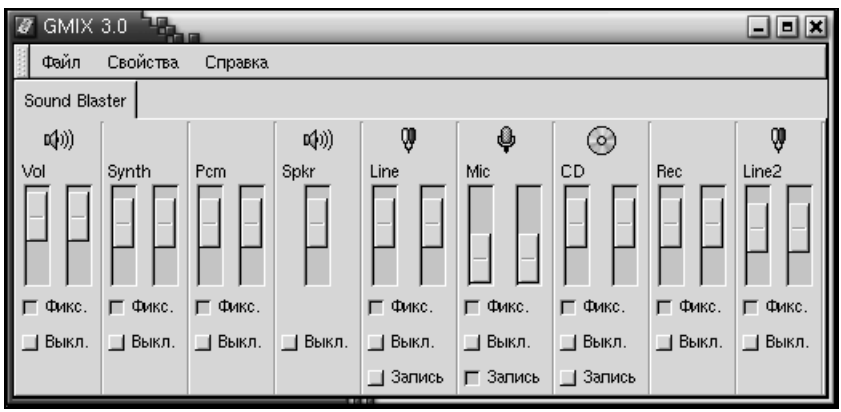

Рис. 2.10. Микшер GMIX, один из многих

KDE и GNOME также включают штатный проигрыватель для воспроизведения WAV-звука. Имеется несколько программ для проигрывания MPEG-3. Во-первых, это mpg123 — консольный проигрыватель, который потребляет очень мало ресурсов. Несколько простеньких MP3-проигрывателей входят в KDE и GNOME, имеются также X11amp (полный функциональный аналог известного Winamp для Windows) и KJukeBox. А если требуется преобразовать аудиодиск в MP3-файлы — есть программы, которые сделают и это, причем можно выбрать различные кодеки, качество оцифровки, получить названия треков из базы данных аудиодисков (при подключении к Интернету). Динамично развивается Vorbis — неплохой кодек для музыки.

Видео

Начнем этот обзор с телевидения, которое принимается на компьютер с помощью TV-тюнера. Наиболее распространенные их модели поддерживаются Linux, правда не совсем понятна ситуация с USB TV-тюнерами. А для воспроизведения видеофайлов, как обычно, выбор достаточно большой. Во-первых, для KDE есть штатный видеоплеер широкого назначения — aKtion. Считывает форматы AVI, MPEG, Quick Time, а также FLI/FLC-анимации. Позволяет масштабирование (оригинальный размер, удвоенный, максимальный), а также полноэкранное воспроизведение, любой из вариантов может быть установлен по умолчанию. Имеет регулятор громкости. Есть возможность захвата кадров (в формате BMP и еще нескольких). Довольно много всяких настроек.

Для воспроизведения видео-CD специально предназначены программы mtv и Xtheater. Для воспроизведения видео, записанного в получающем все большее распространение формате MPEG-4 (DivX), можно воспользоваться программой Mplayer.

Однако поддержкой специальных плат нелинейного монтажа Linux похвастаться пока не может. Есть драйверы, написанные для плат серии Miro, однако они постоянно совершенствуются и поэтому не до конца стабильны. Маловато и программного обеспечения для нелинейного видеомонтажа. Остается надеяться, что положение постепенно исправится.

Игры

С играми, наверное, хуже всего. Одно из основных применений домашнего компьютера помимо мультимедиа — это игры. Зачастую только из-за них приобретается четверть всех домашних компьютеров. Проблема с разнообразием и качеством игр напрямую вытекает из технологии их создания. Для Linux есть большое количество небольших игр типа тетриса, пасьянсов, шахмат, нардов, го и реверси. То есть таких, которые не требуют огромной работы по программированию, написанию сценария, разработке трехмерных моделей и невообразимого количества текстур и рисунков. Как только дело доходит до серьезных игр — сразу образуется вакуум. Фирмы-разработчики игр почему-то не считают рынок Linux перспективным. Игры разрабатываются в расчете на Sony Play Station, Nintendo или Windows. А на рынок Linux с играми они выходить не спешат. Тем не менее (вот парадокс) программ-серверов, рассчитанных на Linux для игры через Интернет, достаточно много (те же Quake, Unreal). До недавнего времени только фирма Id Software выпускала Linux-версии своих игр. Фирма Loki Entertaiment разработала специальную библиотеку и перенесла из Windows в Linux достаточно много популярных игр. Кроме коммерческих игр, есть несколько игр с открытым исходным кодом, хотя, в основном, это Linux-реализации давно

известных коммерческих игр мира Windows. Самыми яркими представителями здесь являются FreeCiv и FreeCraft. Так что, если вы требовательны к разнообразию и качественности игр, к сожалению, Linux пока не для вас.

Итоги

Как следует из материалов этой главы, Linux отлично справится со всякими серверными приложениями и сервисами. С точки зрения администрирования тоже особых проблем нет. Миф о том, что Linux — чисто серверная платформа, и решать на ней офисные задачи невозможно, — только миф. Буквально на днях вышла русская редакция OpenOffice 1, причем, как под Linux, так и под Windows. Помимо этого, динамично развиваются офисы KDE и GNOME. В части домашнего применения картина складывается противоречивая. С одной стороны, отличная поддержка мультимедиа, с другой стороны, практически полное отсутствие современных игр под Linux. Остается надеяться, что с увеличением пользователей Linux производители игр будут выпускать и версии для Linux.

Ссылки

- **www.freshmeat.net** — сайт, содержащий большое количество программ для Linux и не только.
- **www.openoffice.org** — официальный сайт OpenOffice.
- **www.sun.com** — сайт фирмы Sun.
- **koffice.kde.org** — официальный сайт Koffice.
- **www.gnome.org/gnome-office** — официальный сайт GNOME-Office.
- **www.kdevelop.org** — официальный сайт KDevelop.
- **www.codeforge.com** — официальный сайт Code Forge.
- **www.borland.com** — официальный сайт фирмы Borland, разработчика Kylix.
- **www.gnome.org** — официальный сайт GNOME.
- **www.kde.org** — официальный сайт KDE.
- **www.mozilla.org** — официальный сайт Mozilla.
- **www.opera.com** — сайт фирмы-разработчика Opera.
- **www.ximian.com** — сайт фирмы Ximian, внесшей весомый вклад в разработку GNOME, а также почтового клиента Evolution.
- **www.linuxdocs.org** — одно из наиболее полных собраний документации о Linux.

- **www.linux-ve.chat.ru** — электронная библиотека, посвященная Linux.
- **www.citforum.ru** — большое собрание русскоязычной документации и книг, в том числе посвященных Linux.
- **www.linux.org.ru** — один из основных русскоязычных сайтов, посвященных Linux.
- **www.applix.com** — сайт фирмы-разработчика Applixware.
- **www.mysql.org** — официальный сайт MySQL.
- **www.interbase.com** — официальный сайт Interbase.
- **www.idsoftware.com** — разработчик игр Doom, Quake, Quake II, Quake III.
- **www.lokigames.com** — сайт фирмы, которая переносит Windows-игры для Linux.
- **www.linuxgames.org.ru** — сайт, посвященный играм для Linux.

Часть II

Базовая информация о Linux

Глава 3

Работа в сети. Основные понятия

В этой главе будут рассмотрены базовые понятия, лежащие в основе всего последующего изложения. Как уже упоминалось, "компьютер — это сеть". С рассмотрения основных сведений о работе в сети мы и начнем.

Модели сетевых взаимодействий

Как и любая сложная система, сеть должна опираться на стандарты, без которых невозможно нормальное ее функционирование. За последние двадцать лет было создано множество концепций сетевых взаимодействий, однако наибольшее распространение получили всего две:

- модель взаимодействия открытых систем (OSI);
- модель сетевого взаимодействия TCP/IP.

Терминология

Для облегчения понимания содержимого этой главы приведем основные термины (табл. 3.1).

Таблица 3.1. *Базовые сетевые термины*

Термин	Определение
Датаграмма	Пакет, пакет данных. Обозначает единицу информации при сетевом обмене
DNS (Domain Name Service, служба доменных имен)	Специально выделенные компьютеры, которые производят поиск соответствия символического имени хоста и цифрового адреса хоста
Интернет	Глобальная компьютерная сеть, основанная на семействе протоколов TCP/IP
FTP (File Transfer Protocol, протокол передачи файлов)	Используется для приема и передачи файлов между двумя компьютерами

Таблица 3.1 *(окончание)*

Термин	Определение
IP (Internet Protocol, протокол Интернета)	Основа основ семейства протоколов TCP/IP. Практически любой протокол из этого семейства базируется на протоколе IP
NFS (Network File System, сетевая файловая система)	Система виртуальных дисков, позволяющая клиентским компьютерам использовать каталоги сервера в качестве диска
NIC (Network Information Center, сетевой информационный центр)	Организация, которая отвечает за администрирование и раздачу сетевых адресов и имен
Узел (Node, Host)	Компьютер в сети. Название применимо как к клиенту, так и к серверу
OSI (Open System Interconnection, взаимодействие открытых систем)	Модель взаимодействия открытых систем
RFC (Request For Comments, запрос для пояснений)	Стандарты протоколов Интернета и их взаимодействия
RIP (Routing Information Protocol, протокол маршрутизации информации)	Протокол, используемый для обмена информацией между маршрутизаторами
SMTP (Simple Mail Transfer Protocol, простой протокол передачи электронной почты)	Используется для обмена электронной почтой
SNMP (Simple Network Management Protocol, простой протокол управления сетью)	Используется для управления сетевыми устройствами
TCP (Transmission Control Protocol, протокол управления передачей)	Протокол управления передачей. Используется для надежной передачи данных
Telnet	Протокол, осуществляющий удаленное сетевое подключение к компьютеру, эмулирующее терминал
UDP (User Datagram Protocol, протокол пользовательских датаграмм)	Протокол пользовательских датаграмм, используемый для обмена блоками информации без установления соединения

Модель взаимодействия открытых систем (OSI)

Еще в 1983 году Международная организация по стандартизации (International Organization for Standardization, ISO) разработала стандарт взаимодействия открытых систем (Open System Interconnection, OSI).

В результате получилась семиуровневая модель:

1. Физический уровень (Physical Level).
2. Уровень данных (Data Link Level).
3. Сетевой уровень (Network Level).
4. Транспортный уровень (Transport Level).
5. Уровень сессии (Session Level).
6. Уровень представления (Presentation Level).
7. Уровень приложения (Application Level).

Первый уровень самый элементарный, последующие — все более и более абстрагируются от особенностей физической среды передачи информации.

Каждый уровень модели OSI решает свои задачи, использует сервисы, предоставляемые предыдущим уровнем и, в свою очередь, предоставляет сервисы следующему уровню. Согласно этой модели, уровни не могут перескакивать через соседей, например, транспортный уровень не может непосредственно пользоваться сервисом физического уровня, он обязан пройти по цепочке: Сетевой уровень → Уровень данных → Физический уровень. В табл. 3.2 приведено описание уровней сетевой модели OSI.

Таблица 3.2. *Уровни сетевой модели OSI*

Уровень	Название	Описание
1	Физический уровень	Отвечает за физическое подключение компьютера к сети. Определяет уровни напряжения, параметры кабеля, разъемы, распайку проводов и т. п.
2	Уровень данных	Физически подготавливает данные для передачи (разбивая их на кадры определенной структуры) и преобразует обратно во время приема (восстанавливая из кадров)
3	Сетевой уровень	Маршрутизирует данные в сети
4	Транспортный уровень	Обеспечивает последовательность и целостность передачи данных
5	Уровень сессии	Устанавливает и завершает коммуникационные сессии
6	Уровень представления	Выполняет преобразование данных и обеспечивает передачу данных в универсальном формате
7	Уровень приложения	Осуществляет интерфейс между приложением и процессом сетевого взаимодействия

На каждом уровне блоки информации имеют собственное название (табл. 3.3).

Таблица 3.3. *Название блока информации в модели*

Уровень	Название уровня	Название блока информации
1	Физический уровень	Бит
2	Уровень данных	Кадр (пакет)
3	Сетевой уровень	Датаграмма
4	Транспортный уровень	Сегмент
5, 6, 7	Уровень приложения	Сообщение

Несмотря на то что OSI является международным стандартом и на его основе правительство США выпустило спецификации GOSIP (Government Open Systems Interconnection Profile, Государственный регламент взаимодействия открытых систем), у производителей программного обеспечения стандарт OSI широкой поддержки не получил. Это объясняется несколькими причинами:

- на длительное время растянувшаяся процедура принятия стандарта;
- его "оторванность" от реалий;
- наличие большого числа уровней трудно для реализации и приводит к потере производительности;
- широчайшее распространение протокола TCP/IP и нежелание потребителей отказываться от него.

В результате, спецификации OSI сегодня — это, в основном, страницы в учебнике, в реальной жизни они не применяются.

Модель сетевого взаимодействия TCP/IP

Архитектура семейства протоколов TCP/IP (Transmission Control Protocol / Internet Protocol, протокол управления передачей / интернет-протокол) основана на представлении, что коммуникационная инфраструктура содержит три вида объектов: процессы, хосты и сети.

Основываясь на этих трех объектах, разработчики выбрали четырехуровневую модель:

1. Уровень сетевого интерфейса (Network interface layer).
2. Уровень межсетевого интерфейса — интернета[1] (Internet layer).

[1] Здесь "интернет" — термин, указывающий на межсетевой характер взаимодействия, а, отнюдь, не Глобальная сеть Интернет. — *Ред.*

3. Транспортный уровень (Host-to-host Layer).
4. Уровень приложений/процессов (Application/process layer).

Сопоставление сетевых моделей OSI и TCP/IP

Нетрудно заметить, что модель TCP/IP отличается от модели OSI. В табл. 3.4 показано соответствие модели TCP/IP и модели OSI.

Таблица 3.4. *Соответствие модели TCP/IP и модели OSI*

TCP/IP	OSI
Уровень приложений	Уровень приложений
	Уровень представления
	Уровень сеанса
Транспортный уровень	Транспортный уровень
Межсетевой уровень (интернет)	Сетевой уровень
Уровень сетевого интерфейса	Уровень канала данных
	Физический уровень

Как видно из таблицы, уровень сетевого интерфейса сетевой модели TCP/IP соответствует сразу двум уровням сетевой модели OSI, а уровень приложений сетевой модели TCP/IP — трем уровням сетевой модели OSI.

Сетевые протоколы

В этом разделе мы рассмотрим различные сетевые протоколы, используемые в современной компьютерной индустрии.

Семейство протоколов TCP/IP

Семейство протоколов TCP/IP включает следующие протоколы:

- межсетевой протокол (Internet Protocol — IP, протокол интернета) — соответствует уровню интернет-модели TCP/IP. Отвечает за передачу данных с одного хоста на другой;
- межсетевой протокол управления сообщениями (Internet Control Message Protocol, ICMP) — отвечает за низкоуровневую поддержку протокола IP, включая подтверждение получения сообщения, сообщения об ошибках и многое другое;

- протокол преобразования адресов (Address Resolution Protocol, ARP) — выполняет преобразование логических сетевых адресов в аппаратные MAC-адреса (Media Access Control). Соответствует уровню сетевого интерфейса;
- протокол пользовательских датаграмм (User Datagram Protocol, UDP) — обеспечивает пересылку данных без проверки с помощью протокола IP;
- протокол управления передачей (Transmission Control Protocol, TCP) — обеспечивает пересылку данных (с созданием сессии и проверкой передачи данных) с помощью протокола IP;
- множество протоколов уровня приложений (FTP, Telnet, IMAP, SMTP и др.).

Схема протоколов семейства TCP/IP представлена в табл. 3.5.

Таблица 3.5. *Схема семейства протоколов TCP/IP*

Уровень приложений	FTP	SMTP	NFS	SNMP
Транспортный уровень	TCP		UDP	
Межсетевой уровень (интернет)	IP		ARP/RARP	ICMP
Уровень сетевого интерфейса	Ethernet, FDDI, ATM			
	Витая пара, коаксиальный кабель, оптический кабель и т. п.			

Протоколы межсетевого уровня (интернет)

Протоколы межсетевого уровня (интернет) являются базовыми протоколами в семействе протоколов TCP/IP. Это протоколы TCP/IP, ARP/RARP и ICMP.

Протокол IP

Первоначальный стандарт IP разработан в конце 1970-х годов и не был рассчитан на огромное количество хостов, которое сейчас находится в Интернете. Поэтому в настоящее время утвержден новый стандарт IP (в литературе часто старый стандарт встречается как IPv4, а новый — как IPv6). Однако массового применения он пока не нашел из-за огромного количества программных и аппаратных средств, не способных работать с IPv6, поэтому мы здесь будем рассматривать, в основном, протокол IPv4.

Формат пакета IPv4

Пакет IP состоит из заголовка и поля данных. *Заголовок* пакета имеет следующие поля:

- поле *Номер версии* (VERS) указывает версию протокола IP. Сейчас повсеместно используется версия 4 и готовится переход на версию 6;
- поле *Длина заголовка* (HLEN) пакета IP. Занимает 4 бита и указывает значение длины заголовка, измеренное в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при увеличении объема служебной информации эта длина может быть увеличена за счет использования дополнительных байтов в поле *Резерв* (IP OPTIONS);
- поле *Тип сервиса* (SERVICE TYPE) занимает 1 байт и задает приоритетность пакета и вид критерия выбора маршрута. Первые три бита этого поля образуют подполе приоритета пакета (PRECEDENCE). Приоритет может иметь значения от 0 (нормальный пакет) до 7 (пакет управляющей информации). Поле *Тип сервиса* содержит также три бита, определяющие критерий выбора маршрута. Установленный бит D (delay) говорит о том, что маршрут должен выбираться для минимизации задержки доставки данного пакета, бит T — для максимизации пропускной способности, а бит R — для максимизации надежности доставки;
- поле *Общая длина* (TOTAL LENGTH) занимает 2 байта и указывает общую длину пакета с учетом заголовка и поля данных;
- поле *Идентификатор пакета* (IDENTIFICATION) занимает 2 байта и используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета. Все фрагменты должны иметь одинаковое значение этого поля;
- поле *Флаги* (FLAGS) занимает 3 бита, оно указывает на возможность фрагментации пакета (установленный бит Do not Fragment, DF — запрещает маршрутизатору фрагментировать данный пакет), а также на то, является ли данный пакет промежуточным или последним фрагментом исходного пакета (установленный бит More Fragments, MF — говорит о том, что пакет переносит промежуточный фрагмент);
- поле *Смещение фрагмента* (FRAGMENT OFFSET) занимает 13 битов, оно используется для указания в байтах смещения поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации. Используется при сборке/разборке фрагментов пакетов при передачах их между сетями с различными величинами максимальной длины пакета;
- поле *Время жизни* (TIME TO LIVE) занимает 1 байт и указывает предельный срок, в течение которого пакет может перемещаться по сети. Время жизни данного пакета измеряется в секундах и задается источником передачи средствами протокола IP. На шлюзах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица,

единица вычитается также при каждой транзитной передаче (даже если не прошла секунда). По истечении времени жизни пакет аннулируется;

- поле *Идентификатор протокола верхнего уровня* (PROTOCOL) занимает 1 байт и указывает, какому протоколу верхнего уровня принадлежит пакет (например, это могут быть протоколы TCP, UDP или RIP);
- поле *Контрольная сумма* (HEADER CHECKSUM) занимает 2 байта, она рассчитывается по всему заголовку;
- поля *Адрес источника* (SOURCE IP ADDRESS) и *Адрес назначения* (DESTINATION IP ADDRESS) имеют одинаковую длину — 32 бита и одинаковую структуру;
- поле *Резерв* (IP OPTIONS) является необязательным и используется обычно только при отладке сети. Это поле состоит из нескольких подполей, каждое из которых может быть одного из восьми предопределенных типов. Так как число подполей может быть произвольным, то в конце поля *Резерв* должно быть добавлено несколько байтов для выравнивания заголовка пакета по 32-битной границе.

Максимальная длина *поля данных* пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байтов, однако при передаче по сетям различного типа длина пакета выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. В большинстве типов локальных и глобальных сетей определяется такое понятие, как максимальный размер поля данных кадра, в который должен разместить свой пакет протокол IP. Эту величину обычно называют максимальной единицей транспортировки — MTU (Maximum Transfer Unit). К примеру, сети Ethernet имеют значение MTU, равное 1500 байтов, сети FDDI — 4096 байтов.

IP-маршрутизаторы не собирают фрагменты пакетов в более крупные пакеты, даже если на пути встречается сеть, допускающая такое укрупнение. Это связано с тем, что отдельные фрагменты сообщения могут перемещаться в интерсети по различным маршрутам.

При приходе первого фрагмента пакета узел назначения запускает таймер, который определяет максимально допустимое время ожидания прихода остальных фрагментов этого пакета. Если время истекает раньше прибытия последнего фрагмента, то все полученные к этому моменту фрагменты пакета отбрасываются, а в узел, пославший исходный пакет, с помощью протокола ICMP направляется сообщение об ошибке.

Протокол IPv6

Основные причины, из-за которых разрабатывался IPv6:

- протокол IPv4 разрабатывался в конце 1970-х годов с учетом существовавшей на тот момент сетевой инфраструктуры и аппаратного обеспечения. С

того времени производительность массовых компьютеров увеличилась в десятки раз, и во столько же увеличилась пропускная способность сетей;

- появление приложений, использующих Интернет для передачи данных в реальном времени (звук, видео). Эти приложения чувствительны к задержкам передачи пакетов, т. к. такие задержки приводят к искажению передаваемых в реальном времени речевых сообщений и видеоизображений. Особенностью этих приложений является передача очень больших объемов информации. Однако в IPv4 не предусмотрено специального механизма резервирования полосы пропускания или механизма приоритетов;
- бурное развитие сети Интернет. Наиболее очевидным следствием такого развития стало почти полное истощение адресного пространства Интернета, определяемого полем адреса IP в четыре байта. Конечно, были разработаны механизмы компенсации нехватки адресов, однако это не решает проблему.

Основным предложением по модернизации протокола IP является предложение, разработанное группой IETF (Internet Engineering Task Force, группа решения задач межсетевого взаимодействия). В предложении IETF протокол IPv6 оставляет неизменными основные принципы IPv4. К ним относятся датаграммный метод работы, фрагментация пакетов, разрешение отправителю задавать максимальное число хопов (хоп — количество пересылок пакета от одного сетевого интерфейса к другому, иногда называется временем жизни пакета) для своих пакетов. Однако, в деталях реализации протокола IPv6 имеются существенные отличия от IPv4. Эти отличия коротко можно описать следующим образом:

- использование более длинных адресов. Новый размер адреса — наиболее заметное отличие IPv6 от IPv4. Версия 6 использует 128-битные адреса (16 байтов);
- гибкий формат заголовка. Вместо заголовка с фиксированными полями фиксированного размера (за исключением поля Резерв), IPv6 использует базовый заголовок фиксированного формата плюс набор необязательных заголовков различного формата;
- поддержка резервирования пропускной способности;
- поддержка расширяемости протокола. Это одно из наиболее значительных изменений в подходе к построению протокола — от полностью детализированного описания протокола к протоколу, который разрешает поддержку дополнительных функций.

Адресация в IPv6

Адреса в IPv6 имеют длину 128 битов или 16 байтов. Версия 6 обобщает специальные типы адресов версии 4 в следующих типах адресов:

- Unicast — индивидуальный адрес. Определяет отдельный узел — компьютер или порт маршрутизатора. Пакет должен быть доставлен узлу по кратчайшему маршруту;

- Cluster — адрес кластера. Обозначает группу узлов, которые имеют общий адресный префикс (например, присоединенных к одной физической сети). Пакет должен быть маршрутизирован группе узлов по кратчайшему пути, а затем доставлен только одному из членов группы (например, ближайшему узлу);
- Multicast — адрес набора узлов, находящихся в том числе в различных физических сетях. Копии пакета должны быть доставлены каждому узлу набора с использованием аппаратных возможностей групповой или широковещательной доставки, если это возможно.

Как и в версии IPv4, адреса в версии IPv6 делятся на классы, в зависимости от значения нескольких старших битов адреса.

Большая часть классов зарезервирована для будущего применения. Наиболее интересным для практического использования является класс, предназначенный для провайдеров услуг Интернета, названный Provider-Assigned Unicast.

Для обеспечения совместимости со схемой адресации версии IPv4, в версии IPv6 имеется класс адресов, имеющих 0000 0000 в старших битах адреса. Младшие 4 байта адреса этого класса должны содержать адрес IPv4. Маршрутизаторы, поддерживающие обе версии адресов, должны обеспечивать трансляцию при передаче пакета из сети, поддерживающей адресацию IPv4, в сеть, поддерживающую адресацию IPv6, и наоборот.

Сетевые пакеты

Как уже упоминалось, информация по сети передается определенными порциями — пакетами. Причем, на каждом уровне пакет имеет свой размер и структуру. В результате в пакет нижнего уровня вкладывается пакет следующего уровня и т. д. Так же понятно, что чем более высокого уровня пакет, тем меньше информации он может содержать в себе. Размеры пакетов ограничиваются как особенностями аппаратуры, так и требованиями протоколов.

Маршрутизация пакетов

Маршрутизация — механизм передачи пакетов между сетями. При маршрутизации пакетов решается задача, как за наименьшее время, по кратчайшему пути, с минимальной стоимостью доставить пакет. Как правило, в совокупности решить эту задачу не представляется возможным. Поэтому протоколы маршрутизации пакетов должны иметь возможность задавать различные правила и стратегии маршрутизации. К примеру, доставить пакет с максимальной скоростью или с минимальной стоимостью.

Протоколы маршрутизации

Протоколы маршрутизации классифицируются как протокол внутреннего шлюза (Interior Gateway Protocol, IGP) или протокол внешнего шлюза (Exterior Gateway Protocol, EGP).

Протокол внутреннего шлюза управляет маршрутизацией в пределах сети или группы сетей одного владельца, носящей название "автономная система". Внутри автономных систем имеется только список сетей, входящих в автономную систему, и известны точки взаимодействия с внешним миром.

Протокол внешнего шлюза отвечает за маршрутизацию между автономными системами.

На сегодняшний день широко используются следующие протоколы маршрутизации:

- RIP (Routing Information Protocol) — протокол данных маршрутизации. Устаревший протокол. Тем не менее, достаточно широко распространен благодаря утилите routed, которая является стандартной программой для операционных систем UNIX-семейства;
- OSPF (Open Shortest Path First) — протокол выбора кратчайшего пути. Протокол промышленного уровня. Рассчитан на крупные сети со сложной топологией. Более гибок, чем протокол RIP, однако по сравнению с ним сложнее в администрировании;
- IGRP (Interior Gateway Routing Protocol) — протокол маршрутизации внутреннего шлюза. Используется маршрутизаторами CISCO. По всей видимости, скоро сойдет со сцены;
- EGP (Exterior Gateway Protocol) — протокол внешнего шлюза. Старый протокол времен зарождения Интернета. Практически вытеснен протоколом BGP;
- BGP (Border Gateway Protocol) — протокол граничного шлюза. Протокол, в отличие от EGP, поддерживает сложную топологию сети. Имеет возможность широкой настройки стратегии маршрутизации;
- DVMRP (Vector Multicast Routing Protocol) — протокол групповой маршрутизации по вектору расстояния;
- RIP, OSPF и IGRP — внутренние протоколы; EGP и BGP — внешние протоколы.

Адресация в TCP/IP

Каждый компьютер в сети IP имеет адреса трех уровней:

1. *Локальный адрес узла*, определяемый технологией (например, Ethernet), с помощью которой построена отдельная сеть, в которую входит данный узел. Для узлов, входящих в локальные сети — это MAC-адрес (Media Access Control) сетевого адаптера. MAC-адреса назначаются производителями оборудования и являются (теоретически) уникальными адресами,

т. к. управляются централизованно, однако большинство производителей Ethernet-карт предоставляют утилиту для переназначения MAC-адреса. Для всех существующих технологий локальных сетей MAC-адрес имеет 6-байтовый формат: старшие 3 байта — идентификатор фирмы-производителя, а младшие 3 байта назначаются уникальным образом самим производителем.

2. *IP-адрес*, состоящий из 4 байтов (стандарт IPv4) или 16 байтов (стандарт IPv6). Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования сети. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно либо назначен по рекомендации специального подразделения Интернета (Network Information Center, NIC), если сеть должна работать как составная часть Интернета.
3. Символьный *идентификатор-имя*, например, tosser.mail.ru. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени фирмы, имени домена. Такой идентификатор-имя используется на прикладном уровне, например, в протоколе FTP.

IP-адрес состоит из двух частей: сетевой части и адреса хоста. На основании сетевой части адреса принимается решение о сетевой маршрутизации. Адрес хоста однозначно определяет сетевое устройство, которое, в большинстве случаев, совпадает с хостом (как обычно, не обойтись без исключений — некоторые компьютеры имеют несколько IP-адресов). Стандартно IP-адреса записываются как десятичные числа (по одному на каждый байт адреса), разделенные точками. К примеру, 192.168.44.2. Однако не все сетевые адреса могут использоваться для назначения их компьютерам. Исключениями являются адреса 0.0.0.0, 127.0.0.1, 255.255.255.255 и некоторые другие. Существует несколько классов сетевых адресов (табл. 3.6).

Таблица 3.6. *Распределение сетевых адресов по классам сетей*

Класс	Первый байт	Формат*	Комментарии
A	1–26	C.M.M.M	Очень крупные сети, как правило — корпорации или большие государственные учреждения
B	128–191	C.C.M.M	Крупные сети — крупные фирмы, большие интернет-провайдеры
C	192–223	C.C.C.M	Обычная сеть на 254 компьютера
D	224–239	—	Как правило, подсети, выдаваемые провайдерами клиентам
E	240–254	—	Экспериментальные адреса

* В колонке "Формат" буквы C обозначают сетевую часть адреса, а буквы M — его компьютерную часть.

Выделением IP-адресов занимается служба регистрации информационного центра InterNIC, но если необходимо получить 4—5 IP-адресов, то их вполне может предоставить любой интернет-провайдер. Однако не все адреса предназначены для доступа из Интернета. Существует группа адресов, предназначенных для использования только в локальных сетях. Вот эти адреса:

- 10.0.0.0—10.255.255.255
- 172.16.0.0—172.31.255.255
- 192.168.0.0—192.168.255.255

Как можно видеть, это адреса классов A, B и C соответственно.

Несколько IP-адресов имеют специальное значение:

- адрес, в котором сетевая часть содержит нули, соответствует хосту в локальной сети. Например, 0.0.0.145 соответствует рабочей станции 145 в локальной сети, а адрес 0.0.0.0 — текущему хосту;
- сеть с адресом 127.X.X.X — является фиктивной сетью, не имеющей никаких аппаратных сетевых интерфейсов и состоящей только из локального компьютера. Адрес 127.0.0.1 всегда обозначает текущую машину и имеет символическое имя localhost;
- адрес, содержащий в какой-либо части число 255, обозначает широковещательный адрес. Например, пакет, посланный по адресу 192.168.3.255, будет отослан всем компьютерам в сети 192.168.3, а пакет, посланный по адресу 255.255.255.255, отправится по *всем* компьютерам Интернета.

Символьные имена Интернета имеют следующую структуру:

Имя_компьютера.домен3уровня.домен2уровня.домен1уровня

Например: **www.rambler.ru**, **www.yahoo.com**, **www.fklan.com.ua**.

Домены первого уровня стандартизированы и состоят из двух или трех букв латинского алфавита. Правда, в последнее время вводятся домены первого уровня, состоящие из более чем трех букв, но пока массового распространения они не получили. Как правило, домен первого уровня может иметь имя типа com, org, net, mil или двухсимвольного названия страны, за которой закреплен домен: ru — Россия, ua — Украина, uk — Великобритания. Относительно имени домена второго уровня строгих правил нет. Для доменов первого уровня типа com домен второго уровня имеет имя компании или фирмы. Для домена страны правило именования несколько другие. В частности, для России имя домена второго уровня определяется покупателем — к примеру, exler.ru, а для Украины имя домена второго уровня — это либо название областного центра (odessa.ua), либо имя типа com, org, net, mil. Похожая ситуация наблюдается и в других странах — Швеция, Франция,

Германия имена доменов второго уровня жестко не закрепляют, а Великобритания, Тайвань, Япония — закрепляют.

Протокол адресации ARP/RARP

Не смотря на то, что адресация IP-пакетов осуществляется при помощи IP-адресов, при передаче данных с компьютера на компьютер необходимо использовать аппаратные MAC-адреса (конечно, кроме тех случаев, когда используется соединение типа "точка-точка"). Для определения соответствия аппаратных MAC-адресов IP-адресам служит протокол ARP (Address Resolution Protocol) — протокол преобразования адресов. Он применяется в сетях любых типов, использующих широковещательный режим. ARP можно применять только в пределах одной сети. Однако это не мешает передавать пакет через несколько сетей, просто при прохождении пакетом маршрутизатора, он определяет новый MAC-адрес приемника. Каждый компьютер в сети создает кэш ARP, который содержит последние запросы.

Иногда аппаратные адреса необходимо транслировать в IP-адреса. Для этого используется протокол RARP (Reverse Address Resolution Protocol, обратный протокол преобразования адресов).

Протокол ICMP

Протокол ICMP — межсетевой протокол управления сообщениями (Internet Control Message Protocol) отвечает за низкоуровневую поддержку протокола IP, включая подтверждение получения сообщения, сообщения об ошибках и многое другое. В какой-то мере может использоваться для маршрутизации пакетов.

Протоколы транспортного уровня

Протоколы транспортного уровня базируются на протоколе IP. Существуют два протокола транспортного уровня — TCP и UDP. Эти протоколы обеспечивают передачу данных с заданными характеристиками между источником и приемником данных. Эти протоколы вводят новый уровень адресации, так называемый номер порта (port number), который определяет, какому процессу на хосте передаются данные. Номера портов занимают два байта. Существует список соответствия номеров портов приложениям, определенный в RFC1700 (Request For Comments, запрос для пояснений. Данные документы описывают стандарты протоколов Интернета и их взаимодействия). Некоторые зарезервированные порты приведены в табл. 3.7.

Таблица 3.7. *Сервисы и закрепленные за ними порты*

№ порта	Сервис	Описание
7	Echo	Echo
20	FTP-data	Передача данных
21	FTP	Управляющие команды
23	Telnet	Удаленный доступ в систему
25	SMTP	Протокол электронной почты
53	Domain	Сервер доменных имен DNS
80	HTTP	Сервер WWW
110	POP3	Протокол электронной почты
119	NNTP	Телеконференции
123	NTP	Синхронизация времени
161	SNMP	Протокол управления сетевыми устройствами
179	BGP	Маршрутизация

Протокол TCP

Протокол TCP поддерживает надежную передачу данных с предварительным установлением связи между источником и приемником информации. На базе этого протокола реализована бóльшая часть протоколов уровня приложений.

Протокол TCP имеет следующие характеристики, обуславливающие его широкое использование:

- перед началом передачи данных протокол создает канал между источником и приемником информации путем передачи запроса на начало сеанса и получение ответа. По окончании передачи данных сеанс должен быть явно завершен путем передачи соответствующего запроса;
- доставка данных является надежной. Перед отправкой следующего пакета источник информации должен получить подтверждение о приеме предыдущего пакета от приемника информации;
- возможность управления потоком данных;
- возможность доставки экстренных данных.

Эти возможности позволяют программам, использующим протокол TCP, не заботиться об организации надежной передачи данных. С другой стороны, использование этого протокола приводит к уменьшению скорости передачи данных.

Протокол UDP

Протокол UDP обеспечивает логический канал между источником и приемником данных без предварительного установления связи. То есть пакеты, передаваемые по протоколу UDP, не зависят друг от друга, и никакого подтверждения доставки пакета протоколом не предусматривается. Это сильно напоминает бросание бутылки с запиской в море — авось дойдет. Поэтому программы, использующие этот протокол, должны сами организовывать проверку факта доставки информации. Однако благодаря своей простоте протокол UDP может при нормальных условиях передать гораздо больше информации, чем парный ему протокол TCP.

В качестве примера приведем несколько приложений, использующих протокол UDP:

- сервер DNS;
- программы, использующие протокол синхронизации времени NTP;
- программы, использующие протокол удаленной загрузки BOOTP.

Для всех перечисленных программ предполагается, что в случае утери пакета необходимые действия (повторная посылка пакета, выдача сообщения и тому подобные действия) осуществляются самими программами. В случае необходимости гарантированной доставки данных используется протокол TCP.

Протоколы уровня приложений

Последний, четвертый уровень — уровень приложений. К сожалению, почти каждый разработчик программ, использующих протокол уровня приложения, изобретает свой протокол или модифицирует уже существующие. Однако существует некий костяк протоколов, описанный в соответствующих RFC. В зависимости от используемого протокола транспортного уровня протоколы уровня приложений либо полагаются на надежную доставку данных (протокол TCP), либо придумывают свой способ контроля достоверности данных (при использовании протокола UDP). Бóльшая часть протоколов уровня приложений в качестве команд используют обычные английские слова (к примеру, протокол SMTP, HTTP), что значительно упрощает отладку приложений.

Протокол FTP

Протокол передачи файлов. Используется для организации и приема файлов. Позволяет просматривать каталоги и файлы, переименовывать их, удалять и т. п. При пересылке файлов контролирует их целостность. Существует "младший брат" протокола FTP — TFTP, который намного проще в

реализации, и, в основном, используется для загрузки информации на бездисковые рабочие станции.

Протокол SMTP

Простой протокол передачи почтовых сообщений. Позволяет работать с электронной почтой. Благодаря тому, что все команды — обычные английские слова, можно с помощью программы telnet подключиться на 25-й порт (SMTP) и передавать соответствующие команды с консоли.

Протокол Telnet

Протокол предназначен для удаленного доступа в систему. К примеру, можно с домашнего компьютера через Интернет зайти на рабочий компьютер и выполнять на нем любые команды (запускать программы, редактировать файлы и т. п.). Используется, в основном, для удаленного администрирования системы. Считается небезопасным с для операционной системы, т. к. при входе в систему логин (имя пользователя) и пароль передаются в открытом виде. Повсеместно заменяется на протокол SSH.

Сетевая файловая система NFS

Протокол, разработанный фирмой Sun, предназначен для использования дисков и каталогов сервера рабочими станциями в качестве "псевдодисков". Возник очень давно, когда винчестер в сто мегабайт стоил весьма дорого, и использование одного диска, распределяемого через сеть, приводило к существенной экономии денежных средств. Сегодня использование протокола NFS постепенно сходит на нет, одно из немногих мест его применения — бездисковые рабочие станции, использующие NFS в качестве "своей" файловой системы.

Протокол IPX

IPX (Internet Packet Exchange) — протокол обмена пакетами между сетями, разработан фирмой Novell для своего программного продукта NetWare. Однако начиная с четвертой версии своей операционной системы, фирма Novell стала внедрять поддержку протокола TCP/IP, а в пятой версии протокол TCP/IP стал практически "родным" для NetWare. Тем не менее, протокол IPX еще достаточно широко используется.

Протокол IPX произошел от протокола межсетевых датаграмм IDP (Internet Datagram Protocol), разработанного в научно-исследовательском центре Xerox. Протокол IPX реализует механизм сокетов с негарантированной дос-

тавкой датаграмм. Поверх протокола IPX могут функционировать большое количество протоколов, в том числе:

- протокол данных маршрутизации RIP;
- протокол обмена нумерованными пакетами SPX (Sequenced Packet Exchange), гарантированная доставка;
- протокол Echo;
- протокол сообщений об ошибках;
- протокол обмена пакетами PEP (Packet Exchange Protocol);
- протокол сервисных объявлений SAP (Service Advertisement Protocol).

Существует программное обеспечение под Linux (Mars), выполняющее функции сервера NetWare, и программное обеспечение, выступающее клиентом для серверов NetWare. Также есть программное обеспечение под Linux, позволяющее маршрутизировать пакеты IPX.

Протокол AppleTalk

Протокол AppleTalk используется в сетях фирмы Apple. Реально, помимо компьютеров Apple, он не используется нигде. В операционной системе Linux существует поддержка этого протокола, что позволяет взаимодействовать с компьютерами Apple.

Протокол NetBIOS

Протокол, используемый фирмой Microsoft в своих продуктах.

Протокол DECnet

Группа сетевых продуктов фирмы DEC. Поддержка в операционной системе Linux этого протокола существует. Однако маловероятно, что вы столкнетесь с этим протоколом.

Стандарты в Интернете

Стандарты Интернета описаны в документах, известных как RFC (Request For Comments). В табл. 3.8 приведены некоторые стандарты.

Таблица 3.8. *Список основных стандартов Интернета*

Номер стандарта	Комментарий
RFC	Описание протокола UDP
RFC791	Описание протокола IP

Таблица 3.8 *(окончание)*

Номер стандарта	Комментарий
RFC792	Описание протокола ICMP
RFC793	Описание протокола TCP
RFC821	Описание протокола SMTP
RFC826, RFC903	Описание протокола ARP/RARP
RFC827, RFC904, RFC911	Описание протокола маршрутизации EGP
RFC854	Описание протокола Telnet
RFC950	Описание процедуры выделения подсетей
RFC959	Описание протокола FTP
RFC1058	Описание протокола RIP
RFC1094	Описание протокола NFS
RFC1157	Описание протокола SNMP
RFC1178	Рекомендации по выбору сетевого имени компьютера
RFC1180	Введение в TCP/IP
RFC1208	Сетевые термины
RFC1219	Порядок присвоения номеров подсетей
RFC1234	Спецификация по прохождению IPX-пакетов по сетям IP
RFC1245, RFC1246, RFC1247, RFC1583	Описание протокола маршрутизации OSPF
RFC1267	Описание протокола BGP
RFC1597	Распределение локальных IP-адресов
RFC1700	Зарезервированные номера портов

Ссылки

- **www.rfc-editor.org** — сайт, посвященный RFC.
- Man-страница (встроенная страница помощи) arp — программы, работающей с ARP-таблицей.

Глава 4

Идеология файловой системы

Одним из столпов операционной системы является файловая система. От ее архитектуры, возможностей, надежности во многом зависит работоспособность операционной системы. Помимо продуманной "родной" файловой системы крайне желательно, чтобы была возможность также работать с другими наиболее распространенными файловыми системами (например, FAT 16/FAT 32). В этой главе мы подробно рассмотрим, что нам предлагает Linux.

История развития файловых систем Linux

Первоначально Linux разрабатывался как расширение операционной системы Minix, и было вполне логично взять от предшественника все, что можно, поскольку такое решение позволяло достаточно быстро пройти этап проектирования (ведь все уже и так разработано, надо было только создать соответствующий программный код). На тот момент (начало 1990-х, компьютеры на базе 386-го процессора считались мощными, в порядке вещей был жесткий диск емкостью 120 Мбайт) файловая система Minix была достаточно эффективна. Однако ее архитектурные ограничения (адреса блоков 16-битные, что ставит предел максимального объема файловой системы в 64 Мбайт, каталоги содержат записи с ограниченным размером, имя файла не должно превышать 14 символов) очень скоро вынудили разработчиков задуматься об альтернативной файловой системе. Была разработана "Extended File System" (Ext FS — расширенная файловая система), затем ее сменила в качестве стандартной "Second Extended File System" (Ext2FS — вторая расширенная файловая система). Сегодня практически завершена разработка следующей версии файловой системы Ext3 — с поддержкой журналирования. Существуют также и другие журналируемые файловые системы: ReiserFS (стандарт de-facto журналируемых файловых систем для серверов на базе Linux) и JFS от фирмы IBM. По всей видимости, в ближайший год будет произведен полный перевод Linux на журналируемую файловую систему. Пока же большинство системных администраторов используют файловую систему ReiserFS. Однако достаточно много аналитиков предска-

зывают большую вероятность использования системы JFS. Аргументы в ее пользу достаточно весомы: имя IBM, отлаженность, хорошая масштабируемость и надежность. Посмотрим, кто победит.

Файл

Ключевым понятием в операционной системе Linux является концепция файла. Практически все моменты, связанные с данными, в том или ином виде представляются в виде файла или операций с файлами. Для операционной системы Linux по большому счету, все равно, с каким устройством или процессом взаимодействовать — система работает с *файлом*. В результате получается весьма унифицированный интерфейс.

Типы файлов

Поскольку понятие файла применяется к достаточно разнородным вещам (файл как таковой, физические устройства, каталоги и т. п.), поневоле возникает разделение файлов на типы. В Linux существует шесть типов файлов:

- файл;
- каталог;
- файл устройства;
- канал (FIFO, PIPE);
- ссылка (link).
- сокет (Socket).

Файл

Содержит информацию в некотором формате. Для операционной системы — просто набор байтов. Вся интерпретация содержимого файла осуществляется прикладной программой.

Каталог

Каталоги являются элементами иерархического дерева. Любой каталог может содержать файлы и подкаталоги. Каталог — это файл, содержащий список записей. Каждая запись содержит номер индексного дескриптора и имя файла. Структуру записи см. в *разд. "Физическая структура Ext2"*.

Файл устройства

В операционной системе Linux доступ к устройствам осуществляется через специальные файлы. Такой файл является точкой доступа к драйверу устройства. Существует два типа файлов устройств: символьные и блочные.

Символьный файл устройства используется для небуферизированного обмена данными с устройством — байт за байтом.

Блочный файл устройства используется для обмена с устройством блоками данных. Некоторые устройства имеют как символьный, так и блочный интерфейс.

Канал

Файлы этого типа используются для связи между процессами для передачи данных.

Ссылки

Индексный дескриптор может быть связан с несколькими именами файлов. Дескриптор содержит поле, хранящее число, с которым ассоциируется файл. Добавление ссылки заключается в создании записи каталога, где номер индексного дескриптора указывает на другой дескриптор, и увеличении счетчика ссылок в дескрипторе. При удалении ссылки ядро уменьшает счетчик ссылок и удаляет дескриптор, если этот счетчик станет равным нулю. Такие ссылки называются *жесткими* и могут использоваться только внутри одной файловой системы.

Так же существует еще один тип ссылок, называемый *символической* ссылкой. Эта ссылка содержит только имя файла. Так как символическая ссылка не указывает на индексный дескриптор, то возможно создание ссылок на файлы, расположенные в другой файловой системе. Эти ссылки могут указывать на файл любого типа, даже на несуществующий.

Сокет

Сокеты предназначены для взаимодействия между процессами. Часто используются для доступа к сети TCP/IP.

Владельцы файлов

Файлы в Linux имеют трех владельцев — собственно владельца, группу и прочих пользователей. Существует только один владелец, любое количество членов группы и все остальные, которые не входят в группу. Привилегия владения — одно из ключевых понятий в системе защиты операционной системы Linux.

Каждый тип владельца может (или не может) иметь право на чтение и/или запись и/или исполнение файла, владельцем которого он является. На основе этих трех групп владельцев можно построить политику прав доступа к файлам и каталогам, позволяющую достаточно надежно и непротиворечиво обезопасить операционную систему.

Как правило, права доступа к файлу изменяются от максимальных у владельца файла до минимальных (вплоть до полного отсутствия) у всех остальных. Устанавливать и изменять права доступа к файлу или каталогу могут только два пользователя — владелец файла и администратор системы (пользователь root). Изменить права доступа к файлу можно утилитой chmod.

Права доступа к файлам

Права доступа к файлу или к каталогу описываются тремя восьмеричными цифрами, самая левая из этой тройки — права владельца, средняя — права группы, правая — права всех остальных. Каждая из этих восьмеричных цифр представляет собой битовую маску из трех битов. Эти биты отвечают за права на (слева направо) чтение, запись и исполнение файла или каталога. Если установлена единица — доступ разрешен, если ноль — запрещен. Таким образом, права доступа к файлу, описанные цифрой 644, означают, что владелец может писать и читать файл, группа и остальные пользователи — только читать.

Посмотрим, что означает чтение, запись и выполнение файла с точки зрения функциональных возможностей.

- Чтение:
  - возможность просмотра содержимого файла;
  - возможность чтения каталога.
- Запись:
  - возможность добавить или изменить файл;
  - возможность удалять или перемещать файлы в каталоге.
- Выполнение:
  - возможность запуска программы;
  - возможность поиска в каталоге в комбинации с правом чтения.

Узнать о том, какие права доступа установлены к файлам и каталогам, можно, используя команду `ls`. Ниже приведен результат выполнения команды `ls -l`

```
lrwxrwxrwx    1 root     root      4 Авг 31  10:15 [ -> test
-rwxr-xr-x    1 root     root     93 Янв 22  2001  4odb_clean
-rwxr-xr-x    1 root     root     93 Янв 22  2001  4odb_clear
-rwxr-xr-x    1 root     root     95 Янв 22  2001  4odb_create
-rwxr-xr-x    1 root     root     97 Янв 22  2001  4odb_destroy
-rwxr-xr-x    1 root     root     89 Янв 22  2001  4odb_dig
-rwxr-xr-x    1 root     root     93 Янв 22  2001  4odb_grant
```

```
-rwxr-xr-x    1 root     root     97 Янв 22  2001 4odb_metadig
-rwxr-xr-x    1 root     root     99 Янв 22  2001 4odb_odmsdump
drwxr-xr-x    1 root     root     99 Янв 22  2001 t
```

В первой колонке представлены права доступа к файлу, во второй — количество жестких ссылок, в третьей — имя владельца файла, в четвертой — название группы владельца файла, в пятой — дата создания и в шестой — имя файла или каталога. В первой строке листинга вы видите ссылку на `test` (буква `l` в правах доступа обозначает, что это не файл, а ссылка). В последней строке листинга вы видите каталог `t` (буква `d` в правах доступа обозначает, что это каталог (directory), а не файл). Остальные строки листинга — файлы. В правах доступа вы видите десять символов. Первый слева — тип файла (файл, ссылка, каталог и т. п.). Следующие три символа — права доступа владельца файла: `rwx` — чтение, запись, исполняемость файла. Следующие символы, соответственно, права доступа группы и права доступа прочих.

Модификаторы прав доступа

Как у любого правила, в жесткой системе прав доступа существуют свои исключения. Это так называемые дополнительные атрибуты файла:

- Sticky bit (Save Text Attribute) — "липкий" бит;
- SUID (Set User ID) — установка идентификатора пользователя;
- SGID — установка идентификатора группы.

Рассмотрим эти атрибуты подробнее.

- Sticky bit для файлов. В современных операционных системах потерял свое значение.

  Sticky bit для каталогов. Если sticky bit установлен для каталога, то пользователь, несмотря на то, что ему разрешена запись в этот каталог, может удалять только те файлы, владельцем которых он является или к которым ему явно заданы права записи.

- SUID для файлов. Если установлены права доступа SUID и файл исполняемый, то файл при запуске на выполнение получает не права пользователя, запустившего его, а права владельца файла. Такие фокусы используются для того, чтобы пользователь мог работать с некоторыми системными файлами, владельцем которых является некий привилегированный пользователь. К примеру, для того, чтобы пользователь мог самостоятельно изменить свой пароль при помощи утилиты passwd, у этой утилиты (владельцем которой является пользователь root) должен быть установлен бит SUID, поскольку она работает с файлами (/etc/passwd), модификацию которых имеет право производить только пользователь root.

- SGID для файлов. Если установлены права доступа SGID, то это аналогично установке бита SUID, только вместо владельца файла используется группа владельца.

  SGID для каталогов. В случае установки SGID для каталога файлы, содержащиеся в этом каталоге, будут иметь установки группы такие же, как у каталога.

Узнать о том, какие дополнительные права доступа к файлам и каталогам установлены, можно, используя команду `ls`. Ниже приведен результат выполнения команды `ls -l`

```
-r-s--x--x    1 root     root        13536 Июл 12  2000 passwd
```

Как видно из прав доступа, у этого файла установлен SUID-бит (буква `s` в списке прав доступа).

Файловые системы

Файловая система — это методы и структуры данных, которые используются операционной системой для хранения файлов на диске или в его разделе.

Перед тем как раздел или диск могут быть использованы для размещения файловой системы, она должна быть инициализирована, а требуемые служебные данные перенесены на этот раздел или диск. Этот процесс называется созданием файловой системы (иногда его еще называют форматированием, что в принципе неверно).

Основными понятиями в файловой структуре Linux (и в большинстве операционных систем UNIX-семейства) являются:

- суперблок;
- индексный дескриптор (inode);
- блок данных;
- блок каталога;
- косвенный блок;
- файл.

Подробную информацию см. в *разд. "Физическая структура Ext2"*.

Типы файловых систем

Linux поддерживает большое количество типов файловых систем. Наиболее важные из них приведены ниже.

- Minix — старейшая файловая система, ограниченная в своих возможностях (у файлов отсутствуют некоторые временные параметры, длина име-

ни файла ограничена 30-ю символами) и доступных объемах (максимум 64 Мбайт на одну файловую систему).

- Xia — модифицированная версия системы minix, в которой увеличена максимальная длина имени файла и размер файловой системы.
- Ext — предыдущая версия системы Ext2. В настоящее время практически не используется.
- Ext2 — наиболее богатая функциональными возможностями файловая система Linux. На данный момент является самой популярной системой. Разработана с учетом совместимости с последующими версиями.
- Ext3 — модернизация файловой системы Ext2. Помимо некоторых функциональных расширений является журналируемой. Пока широкого распространения не получила. Конкурирующая журналируемая файловая система — ReiserFS.
- VFS — виртуальная файловая система. По сути — эмулятор-прослойка между реальной файловой системой (MS-DOS, Ext2, xia и т. д.) и ядром операционной системы Linux.
- Proc — псевдо-файловая система, в которой посредством обычных файловых операций предоставляется доступ к некоторым параметрам и функциям ядра операционной системы.
- ReiserFS — журналируемая файловая система. Наиболее используемая среди журналируемых файловых систем для Linux.

В операционную систему Linux для обеспечения обмена файлами с другими операционными системами включена поддержка некоторых файловых систем. Однако их функциональные возможности могут быть значительно ограничены по сравнению с возможностями, обычно предоставляемыми файловыми системами UNIX.

- msdos — обеспечивается совместимость с системой MS-DOS.
- umsdos — расширяет возможности драйвера файловой системы MS-DOS для Linux таким образом, что в Linux появляется возможность работы с именами файлов нестандартной длины, просмотра прав доступа к файлу, ссылок, имени пользователя, которому принадлежит файл, а также оперирования с файлами устройств. Это позволяет использовать (эмулировать) файловую систему Linux на файловой системе MS-DOS.
- iso9660 — стандартная файловая система для CD-ROM.
- xenix — файловая система Xenix.
- sysv — файловая система System V (версия для x86).
- hpfs — доступ "только для чтения" к разделам HPFS.

- nfs — сетевая файловая система, обеспечивающая разделение одной файловой системы между несколькими компьютерами для предоставления доступа к ее файлам со всех машин.

В табл. 4.1 содержится общая информация о функциональных возможностях, предоставляемых различными файловыми системами.

Таблица 4.1. *Сравнение файловых систем*

	Minix FS	Xia FS	Ext FS	Ext2 FS
Максимальный объем файловой системы	64 Мбайт	2 Гбайт	2 Гбайт	4 Тбайт
Максимальная длина файла	64 Мбайт	64 Мбайт	2 Гбайт	2 Гбайт
Максимальная длина имени файла	30 символов	248 символов	255 символов	255 символов
Поддержка трех ячеек времени изменения файла	Нет	Да	Нет	Да
Возможность расширения	Нет	Нет	Нет	Да
Изменяемый размер блока	Нет	Нет	Нет	Да
Защита информации	Да	Да	Нет	Да

Установка файловой системы

Файловая система устанавливается при помощи команды `mkfs`. Для каждого типа файловой системы существует своя версия этой программы. Команда `mkfs` запускает требуемую программу в зависимости от типа файловой системы.

Параметры командной строки, передаваемые `mkfs`, слегка различаются для разных типов файловых систем. Полное описание параметров командной строки `mkfs` можно найти в соответствующем разделе man (справочной системы программы). С помощью параметров командной строки можно задать тип создаваемой файловой системы, произвести верификацию диска и маркировку сбойных блоков или получить список сбойных блоков из текстового файла.

Монтирование и демонтирование файловой системы

Для нормальной работы операционной системы ядро каким-то образом должно получить параметры файловых систем, используемых во время работы, и определенным образом настроить специальные таблицы. Для этого существует, по крайней мере, два способа:

1. Каким-то образом один раз получить тип и параметры файловой системы и использовать их все время.
2. Получать их каждый раз при обращении к файловой системе.

У обоих вариантов имеются свои плюсы и минусы. Плюсы первого варианта — уменьшаются затраты времени на определение файловой системы и инициализацию таблиц ядра операционной системы. Минусы — невозможно "на ходу" заменить одно устройство (носитель информации) на другое (к примеру, диск Zip100 на Zip250), поскольку в таблицах ядра зафиксированы емкость носителя, емкость кластеров, используемые блоки и тому подобная информация. Плюсы и минусы второго варианта прямо противоположны первому — возможно "на ходу" заменить устройство (носитель информации), большие затраты времени на определение файловой системы и инициализацию таблиц ядра операционной системы. К тому же, во втором варианте намного труднее достичь надежности хранения данных.

Поэтому большинство операционных систем (не только UNIX) в явной или неявной форме используют первый вариант взаимодействия с файловой системой. Для этого в Linux используются операция "монтирования" и обратная ей "демонтирования" файловой системы. Подробную информацию *см. в гл. 5.*

Поскольку в операционной системе Linux используется единое связанное дерево каталогов, то, в отличие от DOS/Windows, не существует такого понятия файловой системы, как диск. Все дисковые устройства (файловые системы) интегрируются в дереве каталогов в так называемые точки монтирования, в качестве которых выступают обычные каталоги. Причем, если до монтирования в этом каталоге содержались какие-то файлы, то они становятся недоступны до тех пор, пока вы не смонтируете эту файловую систему. Для операции монтирования/демонтирования используются две команды `mount` и `umount`.

Команда `mount` принимает несколько параметров, из которых обязательными являются всего два. Первый их них — файл устройства, соответствующий диску или разделу, на котором расположена файловая система, или его псевдоним (к примеру — CD-ROM, floppy). Вторым параметром является имя каталога, к которому будет монтироваться система. Например, `mount /dev/hda1 /mnt`.

Помимо обязательных параметров можно задавать тип монтируемой файловой системы (при отсутствии этого параметра команда пытается самостоятельно определить ее тип), режим доступа, используемую в именах файлов кодировку и некоторые другие параметры.

Существует специальный файл /etc/fstab, содержащий список файловых систем и их параметры монтирования. Этот файл используется ядром операционной системы при ее старте. Ядро пытается смонтировать файловые системы, описанные в этом файле, с соответствующими параметрами монтирования.

После того как отпала необходимость в использовании файловой системы, ее можно демонтировать. Чаще всего это необходимо при работе с дискетами или дисками CD-ROM (один диск необходимо заменить на другой). Для демонтирования используется команда `umount`. В качестве параметра указывается файл устройства или точка монтирования. Например, `umount /dev/hda1` или `umount /mnt/floppy`.

По окончании работы со сменным носителем информации его обязательно необходимо отмонтировать. Поскольку ядро Linux осуществляет "отложенную" запись на диск, то к тому моменту, когда вы извлечете из дисковода дискету без отмонтирования, информация еще может быть не записана на диск из системного буфера.

Для выполнения операций монтирования и демонтирования требуется наличие прав доступа root. Но при работе на своем персональном компьютере это усложняет процедуру. Есть несколько вариантов решения такой проблемы:

- в KDE или GNOME обычному пользователю можно монтировать CD-ROM и дисковод;
- осуществить временный вход в систему пользователем root, монтировать/демонтировать диск и немедленно выйти;
- применить программу sudo, позволяющую пользователям, для которых это разрешено, использовать команду `mount`;
- применить пакет mtools, используемый для работы с файловой системой MS-DOS;
- поместить список файлов устройств, используемых при работе с гибкими дисками, и доступных узлов монтирования вместе с нужными опциями (разрешением монтирования пользователем) в файл /etc/fstab.

Поддержка работоспособности файловых систем

Даже самая надежная файловая система не обладает стопроцентной надежностью. Рано или поздно целостность файловой системы нарушается. Это

может произойти от некорректного завершения работы системы (нажата кнопка Reset, перебои в электропитании) или повреждения носителя информации. Для проверки и восстановления целостности файловой системы используется команда `fsck`. Она при загрузке системы запускается автоматически, поэтому возможные неполадки будут обнаружены (и может быть исправлены) перед использованием файловой системы.

Полная проверка файловой системы на современных жестких дисках может занять достаточно большое время, поэтому существуют некоторые способы избежать таких проверок. В файловой системе Ext2 существует специальный флаг, расположенный в суперблоке, который используется для выявления корректности демонтирования файловой системы при последнем выключении системы. Так же можно принудительно отключить проверку файловой системы, создав файл /etc/fastboot.

Автоматическая проверка используется только для файловых систем, монтируемых во время загрузки. Для проверки других систем команда `fsck` должна выполняться вручную.

Если `fsck` находит неисправность, которую не может исправить, то для восстановления структуры файловой системы или потерянной информации могут потребоваться глубокие знания и понимание работы файловых систем и их типов.

Команда `fsck` должна использоваться только для демонтированных систем (за исключением корневой файловой системы, которая проверяется смонтированной в режиме read-only), т. к. при ее работе используется прямой доступ к диску, и информация о внесении каких-либо изменений в файловую систему может быть недоступна операционной системе, что, обычно, приводит к нарушению ее работы.

Так же рекомендуется использовать утилиту badblocks. При ее выполнении выводится список номеров найденных на диске поврежденных блоков. Этот список может быть использован программой fsck для внесения изменений в структуру файловой системы.

Виртуальная файловая система (VFS)

База, на которой основывается использование всего многообразия поддерживаемых файловых систем.

Принцип функционирования

Ядро системы Linux содержит в себе программный код-посредник, выполняющий функции виртуальной файловой системы. Этот код обрабатывает запросы к файлам и вызывает необходимые функции соответствующей файловой системы для выполнения операции ввода/вывода. Такой механизм

работы с файлами используется для упрощения объединения и использования нескольких типов файловых систем.

Пусть программа записывает информацию в файл (или считывает ее, не суть важно). Программой вызывается библиотечная функция, отвечающая за запись (или чтение) информации в файл. Эта функция определенным образом подготавливает информацию, которая затем передается в ядро системы. Ядро, в свою очередь, вызывает соответствующую функцию виртуальной файловой системы. Эта функция определяет, с каким типом файловой системы будут производиться манипуляции, подготавливает данные и вызывает необходимую функцию соответствующей файловой системы, с которой производится операция. Такая многоуровневая структура позволяет максимально абстрагироваться от особенностей операционной системы и, в случае необходимости, безболезненно эмулировать недостающие атрибуты файла.

Структура VFS

Виртуальная файловая система содержит набор функций, которые должна поддерживать любая файловая система (создание, удаление, модификация файла, каталога и тому подобные действия). Этот интерфейс состоит из функций, которые оперируют тремя типами объектов: файловые системы, индексные дескрипторы и открытые файлы.

Виртуальная файловая система использует таблицу, в которой во время компиляции ядра сохраняется информация о всех типах поддерживаемых файловых систем. Запись в таблице содержит тип файловой системы и указатель на соответствующую функцию монтирования файловой системы. При монтировании файловой системы эта функция возвращает виртуальной файловой системе дескриптор, который используется в дальнейшем в операциях ввода/вывода.

Дескриптор смонтированной файловой системы содержит определенный набор информации: указатели на функции, служащие для выполнения операций данной файловой системы, и данные, используемые этой системой. Указатели на функции, расположенные в дескрипторе файловой системы, позволяют виртуальной файловой системе получить доступ к функциям, специфичным для данной файловой системы.

В виртуальной файловой системе применяются еще два типа дескрипторов: индексный дескриптор и дескриптор открытого файла. Каждый из них содержит информацию, связанную с обрабатываемыми файлами и набором операций, используемых файловой системой. Индексный дескриптор содержит указатели к функциям, применяемым к любому файлу, а дескриптор открытого файла содержит указатели к функциям, оперирующим только с открытыми файлами.

Файловая система Ext2

Файловая система Ext2 (The Second Extended File System, вторая расширенная файловая система) была разработана с целью устранения ошибок, обнаруженных в предыдущей системе Ext (Extended File System), и снятия некоторых ее ограничений.

Стандартные возможности Ext2

Файловая система Ext2 поддерживает стандартные типы файлов UNIX:

- файлы;
- каталоги;
- файлы устройств;
- символические ссылки.

Ext2 может управлять файловыми системами, установленными на очень больших дисковых разделах. Система поддерживает имена файлов большой длины — до 255 символов. Ext2 резервирует некоторое количество блоков для пользователя root, что позволяет системному администратору избежать нехватки объема жесткого диска при его заполнении другими пользователями.

Дополнительные возможности Ext2

В файловой системе Ext2 может использоваться синхронная модификация данных. Она применяется для достижения высокой плотности записи информации, но одновременно приводит к ухудшению производительности.

Ext2 позволяет при создании файловой системы выбрать размер логического блока. Он может быть определен в 1024, 2048 или 4096 байтов. Организация блоков большого объема приводит к ускорению операций чтения/записи, но при этом дисковое пространство используется нерационально.

Ext2 позволяет применять ускоренные символические ссылки. В этом случае блоки данных файловой системы не используются. Имя файла назначения хранится не в блоке данных, а в самом индексном дескрипторе. Такая структура позволяет сохранить дисковое пространство и ускорить обработку символических ссылок. Максимальная длина имени файла в ускоренной ссылке равна 60 символам.

Ext2 использует отдельное поле в суперблоке для индикации состояния файловой системы. Если файловая система смонтирована в режиме read/write, то ее состояние устанавливается как Not Clean. Если же она демонтирована или смонтирована заново в режиме read-only, то ее состояние устанавливается в Clean. Во время загрузки операционной системы и проверки состояния файловой системы эта информация используется для опре-

деления необходимости такой проверки. Ядро также помещает в это поле некоторые ошибки. При определении ядром какого-либо несоответствия файловая система помечается как Erroneous.

Длительное отсутствие проверки может привести к проблемам функционирования файловой системы, поэтому Ext2 включает в себя два метода для организации принудительной проверки. В суперблоке содержится счетчик монтирования системы. Этот счетчик увеличивается каждый раз, когда система монтируется в режиме read/write. Если его значение достигает максимального значения (оно также хранится в суперблоке), то запускается программа проверки файловой системы, даже если ее состояние является Clean. В суперблоке также хранится последнее время проверки, и максимальный интервал между проверками. При превышении этого интервала также запускается программа проверки файловой системы.

В системе Ext2 имеются утилиты для ее настройки. Так, программа tune2fs используется для определения порядка действий при обнаружении ошибки. Может быть выполнено одно из трех следующих действий:

- продолжение выполнения;
- монтирование файловой системы заново в режиме read-only;
- перезагрузка системы для проверки файловой системы.

Кроме того, эта программа позволяет задать:

- максимальное значение числа монтирований файловой системы;
- максимальный интервал между проверками файловой системы;
- количество логических блоков, зарезервированных для пользователя root.

Физическая структура Ext2

Как и во многих файловых системах, в Ext2 существует загрузочная область. На первичном разделе (primary, в терминологии программы Fdisk фирмы Microsoft) она содержит загрузочную запись — фрагмент кода, который инициирует процесс загрузки операционной системы при запуске. Все остальное пространство раздела делится на блоки стандартного размера. Блок может иметь размер 1, 2 или 4 Кбайт. Блок является минимальной логической единицей дискового пространства (в других операционных системах такой блок называют кластером). Выделение места файлам осуществляется целыми блоками.

Блоки, в свою очередь, объединяются в группы блоков. Каждая группа блоков имеет одинаковое строение. Рассмотрим подробнее их структуру (рис. 4.2).

Рис. 4.1. Структура группы блоков

Суперблок одинаков для всех групп, все же остальные поля индивидуальны для каждой группы. Суперблок хранится в первом блоке каждой группы блоков, является начальной точкой файловой системы, имеет размер 1024 байта и располагается по смещению 1024 байта от начала файловой системы. Копии суперблока используются при восстановлении файловой системы после сбоев.

Информация в суперблоке служит для доступа к остальным данным на диске. В суперблоке определяется размер файловой системы, максимальное число файлов в разделе, объем свободного пространства. При старте операционной системы суперблок считывается в память, и все изменения файловой системы сначала записываются в копию суперблока, находящуюся в оперативной памяти, и только затем сохраняются на диске. При описании структуры суперблока используются следующие значения:

- `SHORT` — короткое целое — 1 байт;
- `USHORT` — беззнаковое короткое целое — 1 байт;
- `LONG` — длинное целое — 4 байта;
- `ULONG` — беззнаковое длинное целое — 4 байта.

Структура суперблока приведена в *приложении 1 (табл. П1.1)*.

После суперблока следует являющееся массивом описание группы блоков (Group Descriptors). Структура описания группы блоков приведена в *приложении 1 (табл. П1.2)*.

Битовая карта блоков (Block Bitmap) — это структура, каждый бит которой показывает, отведен ли соответствующий ему блок какому-либо файлу. Если бит равен 1, то блок занят. Эта карта служит для поиска свободных блоков в тех случаях, когда надо выделить место под файл.

Битовая карта индексных дескрипторов (Inode Bitmap) выполняет аналогичную функцию по отношению к таблице индексных дескрипторов — показывает, какие дескрипторы заняты.

Индексные дескрипторы файлов

Индексные дескрипторы файлов содержат информацию о файлах группы блоков. Каждому файлу на диске соответствует один и только один индексный дескриптор файла, который идентифицируется своим порядковым номером — индексом файла. Отсюда следует, что число файлов, которые могут быть созданы в файловой системе, ограничено числом индексных дескрипторов. Структура индексного дескриптора файла приведена *в приложении 1 (табл. П1.3)*.

Поле типа и прав доступа к файлу (i_mode) представляет собой слово, каждый бит которого служит флагом. Список флагов, описывающих тип и права доступа к файлу, приведен в *приложении 1 (табл. П1.4)*.

Некоторые индексные дескрипторы используются файловой системой в специальных целях. Описание специальных индексных дескрипторов приведено в *приложении 1 (табл. П1.5)*.

Каталог, по сути, является специальным файлом, содержимое которого состоит из записей определенной структуры. Структура записи в файле каталога приведена в *приложении 1 (табл. П1.6)*.

Система адресации данных

Система адресации данных позволяет находить нужный файл среди блоков на диске. В Ext2 система адресации реализуется полем i_block индексного дескриптора файла.

Поле i_block в индексном дескрипторе файла представляет собой массив из 15 адресов блоков. Первые 12 адресов в этом массиве (EXT2_NDIR_BLOCKS [12]) представляют собой прямые ссылки на номера блоков, в которых хранятся данные из файла. Следующий адрес в этом массиве является косвенной ссылкой (адресом блока), в котором хранится список адресов следующих блоков с данными из этого файла. Следующий адрес в поле i_block индексного дескриптора указывает на блок двойной косвенной адресации (double indirect block). Этот блок содержит список адресов блоков, которые, в свою очередь, содержат списки адресов следующих блоков данных того файла, который задается индексным дескриптором.

Последний адрес в поле i_block индексного дескриптора задает адрес блока тройной косвенной адресации, т. е. блока со списком адресов блоков, которые являются блоками двойной косвенной адресации.

Оптимизация производительности

Файловая система Ext2 при операциях ввода/вывода использует буферизацию данных. При считывании блока информации ядро выдает запрос операции ввода/вывода на несколько расположенных рядом блоков. Такие опе-

рации сильно ускоряют извлечение данных при последовательном считывании файлов.

При занесении данных в файл файловая система Ext2, записывая новый блок, заранее размещает рядом до 8 смежных блоков. Такой метод позволяет размещать файлы в смежных блоках, что ускоряет их чтение и дает возможность достичь высокой производительности системы.

Средства управления файловой системы Ext2

Средства управления файловой системы служат для создания, модификации и коррекции любых искажений файловой структуры:

- mke2fs — применяется для установки дискового раздела, содержащего пустую файловую систему Ext2;
- tune2fs — используется для настройки параметров файловой системы;
- e2fsck — предназначена для устранения несоответствий в файловой системе;
- ext2ed — применяется для правки файловой системы;
- debugfs — предназначена для определения и установки состояния файловой системы.

Программа e2fsck спроектирована таким образом, что выполняет проверку с максимально возможной скоростью. В первом проходе e2fsck просматривает все индексные дескрипторы файловой системы и проверяет их как отдельные элементы системы. Также проверяются карты битов, указывающие использование блоков и дескрипторов.

Если e2fsck находит блоки данных, номера которых содержатся более чем в одном дескрипторе, то запускаются проходы с 1B по 1D для устранения несоответствия: либо путем увеличения разделяемых блоков, либо удалением одного или более дескрипторов.

Во втором проходе производится проверка каталогов как отдельных элементов файловой системы. Блок каждого каталога проверяется отдельно, без ссылки на другие блоки каталогов. Для первого блока каталога в каждом дескрипторе каталога, проверяется существование записей "." (ссылка на себя) и ".." (ссылка на родительский каталог), и соответствие номера дескриптора для записи "." текущему каталогу.

В третьем проходе проверяются связи каталогов. Программа e2fsck проверяет пути каждого каталога по направлению к корневому. В этом же проходе проверяется запись ".." для каждого каталога. Все каталоги, не имеющие связи с корневым каталогом, помещаются в каталог /lost+found.

В четвертом проходе e2fsck проверяет счетчики ссылок для каждого индексного дескриптора. Все неудаленные файлы с нулевым счетчиком ссылок также помещаются в каталог /lost+found.

В пятом проходе e2fsck проверяет соответствие всей информации о файловой системе. В этом проходе сравниваются карты битов блоков и дескрипторов, записанных на носителе информации, со значениями, полученными во время проверки файловой системы и, при необходимости, информация на диске корректируется.

Журналируемые файловые системы

Основная цель, которая преследуется при создании журналируемых файловых систем, состоит в том, чтобы обеспечить как можно большую вероятность быстрого восстановления системы после сбоев (например, после потери питания). Дело в том, что если происходит сбой, то часть информации о расположении файлов теряется, поскольку система не успевает записать все изменения из буфера на диск. После сбоя утилита fsck должна проверить все диски, которые не были корректно демонтированы, с целью восстановления потерянной информации. При современных объемах жестких дисков, исчисляемых десятками гигабайт, на проверку двух-трех таких дисков может уйти слишком много времени. Кроме того, нет гарантии, что все данные удастся восстановить.

В журналируемых файловых системах для решения этой проблемы применяют транзакции, которые хорошо известны всем программистам баз данных. Идея транзакции достаточно проста — существует набор связанных операций, называемых транзакцией, и эта группа операций является атомарной (неделимой). Таким образом, транзакция является успешной (завершенной) в том случае, если все операции, составляющие транзакцию, завершились успешно. Но это еще не все. Система ведет журнал, в котором отражаются все действия с данными и все изменения данных протоколируются. В случае сбоя на основании журнала можно вернуть систему в безошибочное состояние.

Основное отличие транзакций из области баз данных от транзакций, применяемых в журналируемых файловых системах, состоит в том, что в базах данных в журнале сохраняются изменяемые данные и вся управляющая информация, а в файловых системах — только мета-данные: индексные дескрипторы изменяемых файлов, битовые карты распределения свободных блоков и свободных индексных дескрипторов.

Файловая система Ext3

По большому счету, файловая система Ext3 не является новой файловой системой. Это похоже на ситуацию с файловой системой FAT 16/FAT 32 — они совместимы, но проблема решена экстенсивным путем. Было необходимо срочно создать журналируемую файловую систему. Если начинать с нуля — долго и накладно, тогда сделали для Ext2 несколько десятков специальных функций и назвали все это Ext3 — получился непонятный гибрид.

Вроде бы добавились журналирующие функции — но не в том объеме, в каком хотелось. И узкие места Ext2 остались: оптимизация использования дискового пространства, ограничение на размер файла и т. п. Пока же общественность (**fido7.ru.linux**) более склоняется к использованию других журналируемых файловых систем.

Файловая система ReiserFS

Кроме проблемы быстрого восстановления после сбоев, в файловой системе Ext2 имеется еще несколько нерешенных проблем. Из самых основных — нерациональное использование дискового пространства, ограничение на размер файла, неоптимальный поиск.

Поскольку в файловой системе используется простой связный список, то время поиска информации линейно зависит от длины списка. Таким образом, чем длиннее список (к примеру, файлов в каталоге), тем дольше идет поиск необходимого элемента.

В системе ReiserFS применяются так называемые "сбалансированные деревья" или "B+Trees", время поиска в которых пропорционально не количеству объектов, а логарифму этого числа. В сбалансированном дереве все ветви имеют одинаковую длину. ReiserFS использует сбалансированные деревья для хранения всех объектов файловой системы: файлов в каталогах, данных о свободных блоках и т. д. Это позволяет существенно повысить производительность обращения к дискам.

Кроме того, система ReiserFS является журналируемой, т. е. в ней решена проблема быстрого восстановления после сбоев. Решена в ReiserFS и проблема с ограничением на размер файла. По всей видимости, именно эта файловая журналируемая система в ближайшее время станет стандартом de-facto для многих дистрибутивов Linux.

Ссылки

- **e2fsprogs.sourceforge.net** — утилиты файловой системы Ext2.
- **ftp.uk.linux.org/pub/linux/sct/fs/jfs/** — код и документация Ext3.
- **www.atnf.csiro.au/~rgooch/linux/docs/vfs.txt** — обзор виртуальной файловой системы.
- **www.osp.ru/pcworld/2000/02/064.htm** — Виктор Хименко. Файлы, файлы, файлы. Обзор файловых систем.
- **www.linux-ve.chat.ru** — виртуальная библиотека Linux.

Глава 5

Дерево каталогов Linux

Эта глава полностью посвящена структуре и размещению каталогов и файлов в Linux. Поскольку для различных дистрибутивов структура может слегка отличаться, для определенности будем рассматривать дистрибутив Red Hat 7.1.

Для того чтобы ориентироваться в Linux, необходимо хорошо представлять себе структуру и размещение каталогов и файлов. Эти параметры для UNIX и Linux описаны в документе "Filesystem Hierarchy Standard — Version 2.2 final", Filesystem Hierarchy Standard Group, edited by Rusty Russell and Daniel Quinlan, редакция от May 23, 2001. Дальнейший текст в основном базируется на этом документе.

Все файлы можно разделить по двум признакам — доступность (shareable, разделяемость) на сетевом уровне и изменяемость/неизменность содержимого.

Соответственно, для каждого признака можно ввести свои понятия:

- *разделяемые* данные — те, которые могут использовать несколько хостов одновременно, т. е. данные, доступные для других хостов через сеть;
- *неразделяемые* данные — как правило, специфичные для каждого хоста, недоступные через сеть для других хостов;
- *статические* данные — включают системные файлы, библиотеки, документацию и другое, что не изменяется без вмешательства администратора;
- *динамические (переменные)* данные — все то, что может изменяться пользователем.

Эти признаки взаимно ортогональны, в табл. 5.1 приведены некоторые каталоги, соответствующие этим признакам.

Таблица 5.1. *Признаки данных и каталоги*

	Разделяемые данные	**Неразделяемые данные**
Статические данные	/usr /opt	/etc /boot
Динамические данные	/var/mail	/var/lock /var/run

Как видно из таблицы, каталог /usr — статический разделяемый, а каталог /var/lock — динамический неразделяемый. По этим признакам можно распределить все каталоги в файловой системе, о чем и будет упоминаться в соответствующих разделах. Однако такое четкое распределение не всегда наблюдается в современных UNIX-системах. Как правило, эта проблема возникает из-за поддержки совместимости со старым программным обеспечением. Каталоги, не удовлетворяющие четкому разделению, будут упомянуты особо.

Иерархия каталогов Linux

В табл. 5.2 приведена иерархия каталогов первого уровня.

Таблица 5.2. *Каталоги первого уровня операционной системы Linux*

Имя каталога	Содержимое каталога
/	Корневой (Root) каталог. Является родительским для всех остальных каталогов в системе
/bin	Содержит важные для функционирования системы файлы
/boot	Содержит файлы для загрузчика ядра
/dev	Хранит файлы устройств
/etc	Содержит Host-специфичные файлы системной конфигурации
/home	Пользовательские домашние каталоги
/lib	Важные разделяемые библиотеки и модули ядра
/lost+found	Содержит файлы, восстановленные при ремонте утилитами восстановления файловых систем
/misc	Каталог для автоматически монтируемых устройств (дисковод, CD-ROM)
/mnt	Точка монтирования временных разделов
/opt	Дополнительные пакеты приложений
/proc	Точка монтирования псевдофайловой системы proc, которая является интерфейсом ядра операционной системы
/root	Домашний каталог для пользователя root
/sbin	Содержит важные системные исполняемые файлы
/tmp	Хранит временные файлы
/usr	Вторичная иерархия
/var	Содержит переменные данные

Рассмотрим подробнее иерархию каталогов.

Корневой (Root) каталог

Точка монтирования всей файловой системы. Играет исключительно важную роль в процессе "жизнедеятельности" операционной системы. Для загрузки системы необходимо, чтобы в корневом разделе (корневой раздел в Linux — это аналог диска C: для DOS/Windows — только на него возможно установить операционную систему. И корневой раздел является точкой монтирования корневого каталога) находились утилиты и конфигурационные файлы, необходимые для монтирования других файловых систем. Кроме того, в корневой файловой системе должны присутствовать утилиты, необходимые для создания, восстановления или ремонта файловых систем, а также для административного восстановления (backup) системы с ленты, CD-ROM, дискет и тому подобных носителей. Каталоги /usr, /opt, /var спроектированы так, что они могут размещаться на файловых системах, отличных от корневой. В дистрибутиве Slackware в корневом каталоге по умолчанию находится ядро операционной системы (что на больших винчестерах иногда вызывало определенные проблемы), в дистрибутиве Red Hat ядро операционной системы перенесено в каталог /boot.

Имеется несколько причин, по которым корневую файловую систему рекомендуется делать минимально возможного размера:

- это позволяет монтировать файловую систему с очень маленьких носителей информации (например дискет);
- корневая файловая система не может быть разделяемой, потому что содержит много системно-зависимых конфигурационных файлов. Создание малой по объему корневой файловой системы позволяет сохранить на серверах больше места для разделяемых ресурсов;
- у маленького по объему корневого каталога меньше вероятность пострадать при крахе системы.

Каталог /bin

Содержит важные исполняемые файлы, которые используются всеми (в том числе и администратором системы) пользователями. Кроме того, в каталоге /bin должны находиться исполняемые файлы, необходимые для функционирования системы в однопользовательском режиме (single mode). Он также может содержать исполняемые файлы, которые напрямую используются в скриптах. Каталог /bin не должен содержать подкаталогов. Исполняемые файлы, от которых напрямую не зависит функционирование системы, рекомендуется размещать во вторичной иерархии — в каталоге /usr/bin.

Таким образом, в каталоге /bin должны находиться следующие файлы или символические ссылки на команды:

- cat — утилита, выдающая на стандартное устройство вывода объединенные файлы;

- chgrp — утилита, позволяющая изменить группу владельца файла;
- chmod — утилита, изменяющая права доступа к файлу;
- chown — утилита, изменяющая владельца и группу файла;
- cp — утилита, позволяющая копировать файлы и каталоги;
- date — утилита, позволяющая вывести или установить системные дату и время;
- dd — утилита, позволяющая конвертировать и копировать файл;
- df — утилита, показывающая использование дискового пространства;
- dmesg — утилита, выводящая или управляющая буфером сообщения ядра;
- echo — утилита, отображающая строку текста;
- false — утилита возвращает значение "Не успешно" (unsuccessfully) ;
- hostname — утилита, показывающая или устанавливающая имя хоста;
- kill — утилита, посылающая управляющие сигналы процессам;
- ln — утилита, создающая линки (связи, ссылки) между файлами;
- login — утилита, начинающая сессию в системе;
- ls — утилита, показывающая содержимое каталога;
- mkdir — утилита, позволяющая создавать каталог;
- mknod — утилита, создающая блочные или символьные специальные файлы;
- more — утилита, позволяющая просматривать текстовые файлы постранично;
- mount — утилита, монтирующая файловую систему;
- mv — утилита, перемещающая или переименовывающая файлы;
- ps — утилита, показывающая статус процессов;
- pwd — утилита, выводящая имя текущего рабочего каталога;
- rm — утилита, удаляющая файлы или каталоги;
- rmdir — утилита, удаляющая пустой каталог;
- sed — редактор;
- setserial — программа настройки последовательных портов;
- sh — командная оболочка Bourne;
- sty — утилита, изменяющая и выводящая установки терминальной линии;
- su — утилита, изменяющая пользовательский идентификатор (user ID);
- sync — утилита, сбрасывающая (flush) буферы файловой системы;
- true — утилита возвращает значение "Успешно" (successfully);

- umount — утилита, размонтирующая файловые системы;
- uname — утилита, выводящая системную информацию.

Если в системе не используется утилита sh, то sh должна быть ссылкой на используемую системой командную оболочку.

Если установлены соответствующие пакеты, в каталоге /bin могут присутствовать следующие программы или символические ссылки:

- csh — командная оболочка C shell;
- ed — редактор;
- tar — архивная утилита;
- cpio — архивная утилита;
- gzip — утилита архивации файлов GNU;
- gunzip — утилита разархивации файлов GNU;
- zcat — утилита разархивации файлов GNU;
- netstat — утилита сетевой статистики;
- ping — ICMP-сетевая утилита.

Каталог /boot

Содержит все, что требуется для процесса загрузки, исключая файлы конфигурации. В каталоге /boot находятся данные, используемые ядром до того, как оно начинает исполнять программы пользовательского режима (user-mode). В этом же каталоге может находиться сохраненный сектор master boot и другие специфичные данные. Конфигурационные файлы загрузчика находятся в каталоге /etc. Ядро операционной системы, как было сказано выше, должно находиться или в корневом каталоге (дистрибутив Slackware), или в каталоге /boot (дистрибутив Red Hat). В некоторых случаях приходится создавать отдельный раздел /boot, находящийся до 1024 цилиндра. Как правило, это зависит от версии загрузчика и от BIOS компьютера. Таким образом, в каталоге /boot версии Linux Red Hat 7.1 должны находиться следующие файлы или символические ссылки на команды:

boot.0300	kernel.h-2.4.3	module-info-2.4.2-2	vmlinux-2.4.2-2*
boot.b	map	os2_d.b	vmlinuz@
chain.b	message	System.map@	vmlinuz-2.4.2-2.
kernel.h@	module-info@	System.map-2.4.2-2	

Каталог /dev

Содержит файлы устройств или специальные файлы. Создание в каталоге /dev файлов устройств осуществляется с помощью предназначенной для

этого утилиты makedev, находящейся в нем же. Также в этом каталоге может находиться утилита makedev.local, предназначенная для создания локальных устройств. Все устройства и специальные файлы описываются в документе Linux Allocated Devices, который поставляется вместе с исходным кодом ядра (см. также соответствующую справочную документацию).

Каталог /etc

Каталог содержит конфигурационные файлы и каталоги, которые специфичны для данной системы. В этом каталоге не должно находиться никаких исполняемых модулей. В каталоге /etc обязательно должен присутствовать каталог /opt, содержащий конфигурационные файлы для программ, установленных в каталоге /opt.

Замечание

Везде, где далее упоминается "... должны присутствовать в каталоге /etc", надо учитывать, что соответствующие файлы и каталоги появляются в /etc только в том случае, если соответствующие программы установлены в системе.

В каталоге /etc также должны присутствовать следующие каталоги:

- /cron.d — конфигурация cron;
- /cron.daily — ежедневно выполняемые операции cron и anacron;
- /cron.hourly — ежечасно выполняемые операции cron и anacron;
- /cron.monthly — ежемесячно выполняемые операции cron и anacron;
- /cron.weekly — еженедельно выполняемые операции cron и anacron;
- /default — в этом каталоге находятся файлы, используемые пакетом shadow при создании новой учетной записи пользователя в системе;
- /gnome — в этом каталоге содержится разнообразная конфигурационная информация, касающаяся графической системы GNOME и ее приложений. Информацию о конфигурации GNOME и ее приложений смотрите в руководстве пользователя GNOME;
- /kde — в этом каталоге содержится разнообразная конфигурационная информация, касающаяся графической системы KDE и ее приложений. Информацию о конфигурации KDE и ее приложений смотрите в руководстве пользователя KDE;
- /locale — настройки локали;
- /opt — в этом каталоге хранятся конфигурационные файлы для пакетов, устанавливаемых в каталоге /opt. Для каждого пакета создается (точно так же, как и в /opt) свой каталог, с точно таким же именем, как и в /opt, в котором содержатся конфигурационные файлы для этого пакета;

- /ppp — в этом каталоге находятся конфигурационные файлы и скрипты, необходимые для функционирования демона pppd. В частности, здесь находятся скрипты, поднимающие и опускающие PPP-интерфейс с поддержкой IPv4 и IPv6, скрипты аутентификации и конфигурационные файлы;
- /rc.d — каталог скриптов, используемых при старте системы;
- samba — этом каталоге находятся конфигурационные файлы для сервера Samba. Список файлов, которые обычно содержатся в этом каталоге:
  - lmhosts — содержит список хостов и соответствующих им адресов;
  - smbpasswd — содержит пароли пользователей сервера Samba;
  - smbusers — файл, предназначенный для хранения конфигурационных файлов пользователей, которым разрешен доступ к ресурсам Samba;
  - smb.conf — главный конфигурационный файл сервера;
- /sgml — содержит конфигурации для SGML и XML;
- /skel — содержит конфигурационные файлы для вновь создаваемых пользователей. В этом каталоге хранятся конфигурационные файлы пользователя, которые при создании нового пользователя в системе копируются в его домашний каталог. Это очень удобно с точки зрения системного администратора — один раз настроив окружение пользователя, мы для вновь созданных пользователей получаем уже готовое окружение. Мы можем определить язык, раскладку клавиатуры, палитру, редактор по умолчанию, графическую оболочку и многое-многое другое. Не следует думать, что этим мы ограничиваем пользователя — наоборот — он получает настроенное рабочее место. Если ему что-то не подходит — он может внести необходимые ему изменения в *свои* конфигурационные файлы. Таким образом, мы получаем с одной стороны — единообразие, а с другой — возможности для индивидуализации рабочего места.

  Обычно в этом каталоге находятся следующие файлы:

.bashrc	.bash_logout	.less	.Xdefaults
.bash_profile	.inputrc	.xinitrc	

  Однако ничто не мешает удалить или, наоборот, добавить файлы в этот каталог;
- /sysconfig — каталог, содержащий файлы системной конфигурации;
- /X11 — содержит конфигурационные файлы для X Window System;

Кроме перечисленных каталогов в каталоге /etc должны находиться следующие файлы:

- aliases — этот файл определяет для программы доставки почтовых сообщений, куда посылать письма, приходящие на адрес псевдопользователей. Большей частью они перенаправляются пользователю root;

- anacrontab — конфигурационный файл для программы anacron. В этом файле задаются периодичность выполнения команд (ежедневно, еженедельно, ежемесячно) и каталоги, в которых содержатся исполняемые модули (как правило — скрипты).

  Программа anacron использует те же каталоги с исполняемыми модулями, что и cron. Однако программа anacron применяется в системах, которые не предназначены для постоянного функционирования (24 часа в сутки). Программа просматривает список задач и запускает текущие в списке или *просроченные*;
- at.allow — с помощью этого файла задается список пользователей, которым разрешено пользоваться командой `at`;
- at.deny — с помощью этого файла задается список пользователей, которым запрещено пользоваться командой `at`;
- bashrc — конфигурационный файл, определяющий поведение bash. Как правило, не требует ручного вмешательства;
- cron.allow — с помощью этого файла задается список пользователей, которым разрешено пользоваться демоном cron;
- cron.deny — с помощью этого файла задается список пользователей, которым запрещено пользоваться демоном cron;
- crontab — конфигурационный файл для программы cron. В этом файле задаются периодичность выполнения команд (ежечасно, ежедневно, еженедельно, ежемесячно) и каталоги, в которых содержатся исполняемые модули (как правило — скрипты);

  Программа cron рассчитана на постоянно функционирующие системы. Поэтому, если во время, когда компьютер был отключен, необходимо было выполнить какую-то операцию — программа cron не поможет. Для выполнения просроченных операций необходимо использовать программу anacron;
- cron.allow — программа cron может разрешать или запрещать конкретным пользователем свое использование. Для разрешения конкретным пользователям использования программы cron необходимо вписать имена соответствующих пользователей в файл cron.allow;
- cron.deny — конфигурационный файл для программы cron, с помощью которого можно запретить использование программы cron конкретным пользователям или всем пользователям кроме тех, которые записаны в файле cron.allow;
- dir_colors — этот файл определяет, каким цветом будет выводить на экран файлы команда `ls`. Для разных типов файлов можно определить свой цвет;
- exports — файл, содержащий управление доступом к файловой системе NFS;

- fstab — файл, содержащий таблицу, в которой определены монтируемые устройства (файлы драйверов), соответствующие им точки монтирования, тип файловой системы и параметры монтирования. Пример файла fstab приведен ниже:

```
LABEL=/           /              ext3      defaults              1 1
LABEL=/boot       /boot          ext2      defaults              1 2
none              /dev/pts       devpts    gid=5,mode=620        0 0
none              /proc          proc      defaults              0 0
none              /dev/shm       tmpfs     defaults              0 0
/dev/hda8         swap           swap      defaults              0 0
/dev/cdrom        /mnt/cdrom     iso9660   noauto,owner,kudzu,ro 0 0
/dev/fd0          /mnt/floppy    auto      noauto,owner,kudzu    0 0
```

- ftpusers — конфигурационный файл FTP-демона, содержащий список пользователей FTP с их правами доступа;
- gateways — файл, содержащий список шлюзов (gateways) для демона маршрутизации routed;
- gettydefs — файл, содержащий терминальные установки, используемые getty;
- group — в этом файле содержатся пользователи и группы, членами которых они являются. Файл состоит из строк, в каждой строке — 4 поля:
  - имя пользователя;
  - пароль;
  - GUID — числовой идентификатор группы;
  - список имен групп, к которым принадлежит пользователь.

  Пример файла group приведен ниже:

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
wheel:x:10:root
mail:x:12:mail
news:x:13:news
nobody:x:99:
users:x:100:
slocate:x:21:
```

```
floppy:x:19:
utmp:x:22:
mysql:x:27:
alst:x:500:
```

- gshadow — содержимое этого файла напоминает содержимое файла group.

```
root:::root
bin:::root,bin,daemon
daemon:::root,bin,daemon
sys:::root,bin,adm
adm:::root,adm,daemon
disk:::root
lp:::daemon,lp
mem:::
kmem:::
wheel:::root
utmp:x::
mailnull:x::
mysql:x::
alst:!::
```

- host.conf — конфигурационный файл, который определяет порядок разрешения символического имени хоста в IP-адресе. Обычно содержимое этого файла имеет вид:

```
order hosts,bind
```

- hostname — в этом файле содержится (обычно) имя хоста. Текущее имя хоста можно посмотреть с помощью команды `hostname`;
- hosts — содержимое этого файла используется для определения пары IP-адрес — символическое имя хоста. Очень рекомендуется, чтобы в этом файле была следующая запись:

```
127.0.0.1localhost.localdomain localhost
```

Замечание

Если она отсутствует — возникнут проблемы, связанные с сетью (в частности, возможно зависание программы sendmail).

- hosts.allow — файл, определяющий, каким хостам разрешено подключаться к системе;
- hosts.deny — файл, определяющий, каким хостам запрещено подключаться к системе;

- hosts.equiv — файл, содержащий список доверенных хостов для rlogin, rsh, rcp;
- hosts.lpd — файл, содержащий список доверенных хостов для lpd;
- inetd.conf — конфигурационный файл для демона inetd;
- inittab — конфигурационный файл для процесса init. Этот файл описывает, как процесс init должен настроить операционную систему в соответствующем уровне исполнения. Более подробную информацию *см. в гл. 6.*
- issue — в этом файле содержится сообщение, выдаваемое системой до приглашения "login:".

  Для дистрибутива Red Hat Linux 7.2 этот файл содержит следующее сообщение:

```
Red Hat Linux release 7.2 (Enigma)
Kernel \r on an \m
```

- ld.so.conf — файл, содержащий список каталогов для поиска разделяемых библиотек;
- lilo.conf — конфигурационный файл для загрузчика lilo. Более подробную информацию об этом конфигурационном файле вы можете прочитать в справочных страницах man.

Внимание

После внесения изменений в файл lilo.conf необходимо выполнить команду `lilo`. В противном случае внесенные в конфигурационный файл изменения не воспримутся загрузчиком.

Пример файла lilo.conf приведен ниже:

```
prompt
timeout=50
default=DOS
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
message=/boot/message
lba32

image=/boot/vmlinuz-2.4.7-10
    label=linux
    initrd=/boot/initrd-2.4.7-10.img
    read-only
    root=/dev/hda7
```

```
other=/dev/hda2
    optional
    label=DOS
```

- localtime — бинарный файл, определяющий временную зону компьютера, правила перехода на летнее/зимнее время и другую информацию, связанную с местной временной зоной. Обычно берется один из файлов, находящихся в каталоге /usr/share/zoneinfo/, и копируется в каталог /etc с именем localtime. В том случае, если для вас не существует готового файла localtime, его можно создать с помощью утилиты zic;
- man.config — конфигурационный файл, содержащий настройки для справочных страниц man;
- modules.conf — файл, используемый операционной системой для загрузки по требованию программ некоторых модулей ядра. Обычно используется для модулей звуковых карт и плат TV-тюнеров, или в том случае, если в системе установлено несколько сетевых плат;
- motd — сообщение, выдаваемое системой после входа пользователя в систему;
- mtab — файл, содержащий динамическую информацию о файловых системах;
- mtools.conf — конфигурационный файл для mtools;
- networks — файл, содержащий статическую информацию о сетевых именах;
- passwd — файл содержит информацию обо всех пользователях системы, в том числе и псевдопользователях, которые необходимы для правильного функционирования некоторых сервисов. Типичный файл passwd имеет следующий вид:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
```

```
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/dev/null
rpm:x:37:37::/var/lib/rpm:/bin/bash
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
ntp:x:38:38::/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/bin/false
gdm:x:42:42::/var/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/bin/false
ident:x:98:98:pident user:/:/sbin/nologin
radvd:x:75:75:radvd user:/:/bin/false
apache:x:48:48:Apache:/var/www:/bin/false
squid:x:23:23::/var/spool/squid:/dev/null
pcap:x:77:77::/var/arpwatch:/bin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
vasya:x:500:500::/home/vasya:/bin/bash
```

Этот файл содержит строки, где каждая определяет одного пользователя. В строке есть семь полей, отделенных друг от друга двоеточием. Рассмотрим более подробно эти поля:

- имя пользователя;
- пароль пользователя; в современных системах не используется (стоит символ x). Для хранения пароля используется файл shadow;
- UID — числовой идентификатор пользователя;
- GID — числовой идентификатор группы пользователя;
- поле описания пользователя (телефон, адрес и т. п.). Обычно не используется;
- домашний каталог пользователя;
- командный интерпретатор, используемый пользователем.

Никто, кроме пользователя root, не имеет доступа на запись в файл passwd. Таким образом, если вам необходимо запретить пользователю регистрироваться в системе, можно назначить ему командный интерпретатор /sbin/nologin или /dev/null. Другой способ — отредактировать пароль (смотри shadow).

- printcap — база совместимых принтеров для lpd;

- profile — общесистемный конфигурационный файл для оболочки sh. Все настройки, добавленные в этот файл, отражаются на переменные окружения вашей системы;
- protocols — файл, содержащий список IP-протоколов;
- resolv.conf — конфигурационный файл Resolver;
- rpc — файл, содержащий RPC-список протоколов;
- securetty — управление доступом к консоли для администратора (root);
- sendmail.cf — конфигурационный файл программы sendmail — программы передачи почтовых сообщений;
- services — файл, содержащий имена портов для сетевых сервисов. Описание сервиса представляет собой строку, которая содержит четыре поля:
  - имя сервиса;
  - номер порта/протокол;
  - псевдонимы;
  - комментарии.

  Пример записи в файле services:

```
tcpmux          1/tcp           # TCP port service multiplexer
tcpmux          1/udp           # TCP port service multiplexer
```

- shadow — файл паролей, по структуре напоминающий passwd. Содержит полный список пользователей системы. Интересны первые два поля — имя пользователя и хэш пароля. Алгоритм создания хэша пароля работает таким образом, что *никогда* первым символом хэша не может быть символ *. Таким образом, для того, чтобы запретить пользователю вход в систему, достаточно первым символом пароля поставить символ `*`.

  Пример файла shadow:

```
root:$1$є¦zA¦+NЎ$ZL.87fvylY.:11689:0:99999:7:::
bin:*:11689:0:99999:7:::
daemon:*:11689:0:99999:7:::
adm:*:11689:0:99999:7:::
lp:*:11689:0:99999:7:::
```

- shells — файл, содержащий пути для установленных командных оболочек;
- sysctl.conf — файл конфигурации ядра операционной системы. Позволяет производить настройку различных свойств ядра — сетевых и не только.

  Пример файла sysctl.conf:

```
# Disables packet forwarding
net.ipv4.ip_forward = 0
```

```
# Enables source route verification
net.ipv4.conf.default.rp_filter = 1
# Disables the magic-sysrq key
kernel.sysrq = 0
```

- syslog.conf — конфигурационный файл для демона syslogd.

/etc/rc.d — инициализационные скрипты системы

В каталоге содержатся следующие файлы и каталоги, необходимые для загрузки операционной системы Linux и запуска необходимых сервисов:

- /init.d — каталог содержит управляющие скрипты для загружаемых при старте операционной системы сервисов;
- /rc0.d — каталог содержит скрипты, отвечающие за запуск и остановку сервисов при переходе на нулевой уровень выполнения;
- /rc1.d — каталог содержит скрипты, отвечающие за запуск и остановку сервисов при переходе на первый уровень выполнения;
- /rc2.d — каталог содержит скрипты, отвечающие за запуск и остановку сервисов при переходе на второй уровень выполнения;
- /rc3.d — каталог содержит скрипты, отвечающие за запуск и остановку сервисов при переходе на третий уровень выполнения;
- /rc4.d — каталог содержит скрипты, отвечающие за запуск и остановку сервисов при переходе на четвертый уровень выполнения;
- /rc5.d — каталог содержит скрипты, отвечающие за запуск и остановку сервисов при переходе на пятый уровень выполнения;
- /rc6.d — каталог содержит скрипты, отвечающие за запуск и остановку сервисов при переходе на шестой уровень выполнения;
- rc — файл предназначен для запуска и останова сервисов при переходе в указанный уровень выполнения;
- rc.local — файл предназначен для команд, добавляемых администратором для запуска в процессе начальной загрузки;
- rc.sysinit — файл предназначен для выполнения начальных действий, необходимых для корректного функционирования операционной системы;

/etc/rc.d/init.d — управляющие скрипты для сервисов

Каталог содержит управляющие скрипты для сервисов, которые выполняются (или могут выполняться) при старте системы или при переходе с одного уровня выполнения на другой.

Если соответствующие сервисы установлены, в этом каталоге находятся следующие файлы:

anacron	ipchains	nfslock	sendmail
apmd	iptables	nscd	single
arpwatch	isdn	portmap	snmpd
atd	kdcrotate	pppoe	sshd
autofs	keytable	random	syslog
crond	killall	awdevices	tux
functions	kudzu	rhnsd	windows
gpm	lpd	rstatd	xfs
halt	netfs	rusersd	xinetd
httpd	network	rwalld	ypbind
identd	nfs	rwhod	ppasswdd
			ypserv

/etc/rc.d/rc0.d ... rc6.d — каталоги для соответствующего уровня выполнения

Эти каталоги содержат стартовые и стоповые скрипты сервисов, используемых операционной системой при переходе в нужный уровень выполнения. К примеру, каталог /rc3.d конкретного компьютера может содержать следующие файлы:

K03rhnsd	K50tux	S13portmap	S56xinetd
K15httpd	K65identd	S14nfslock	S60lpd
K20nfs	K73ypbind	S17keytable	S80isdn
K20rstatd	K74nscd	S20random	S80pppoe
K20rusersd	K74ypserv	S25netfs	S80sendmail
K20rwalld	S05kudzu	S26apmd	S85gpm
K20rwhod	S08ipchains	S28autofs	S90crond
K34yppasswdd	S08iptables	S40atd	S90xfs
K45arpwatch	S10network	S55sshd	S95anacron
K50snmpd	S12syslog	S56rawdevices	S99local

S99windows/etc/sysconfig — конфигурационные файлы для процессов

Каталог содержит различные конфигурационные файлы и скрипты, используемые операционной системой во время загрузки и останова сервисов.

В частности, в нем находятся следующие файлы и каталоги:

- /etc/sysconfig/apm-scripts — в этом каталоге находятся скрипты, относящиеся к демону apmd, предназначенному для управления питанием системы. Чаще всего используется в системах, установленных на ноутбуках;
- /etc/sysconfig/cbq — каталог для конфигурирования программы cbq — так называемого траффик-шейпера. Принцип действия — искусственно ограничивает полосу пропускания сетевого устройства с заданной шириной канала;
- /etc/sysconfig/console — каталог для конфигурирования консоли. В частности, /etc/sysconfig/console/default.kmap — файл раскладки клавиатуры по умолчанию;
- /etc/sysconfig/network/ — каталог хранит различные настройки сети, а также скрипты, отвечающие за старт и останов сетевой подсистемы;
- /etc/sysconfig/clock — файл используется для конфигурирования системных часов (временная зона, формат хранения времени, переход на летнее/зимнее время и т. п.);
- /etc/sysconfig/i18n/ — каталог содержит файлы, связанные с локализацией системы, в частности, шрифты.

В самом каталоге /sysconfig находятся следующие файлы:

- apmd — этот файл отвечает за конфигурацию демона управления электропитанием;
- arpwatch — файл, отвечающий за конфигурацию программы arpwatch;
- clock — файл, отвечающий за конфигурацию часовой зоны и некоторых других параметров. Например:

  ```
  ZONE="Europe/Kiev"
  UTC=false
  ARC=false
  ```

  Как видно из примера, системные часы не используют универсальное представление времени, а система находится в Киевском часовом поясе (Гринвич + 2 часа);
- gpm — файл предназначен для конфигурирования gpm — программы, осуществляющей поддержку мыши в консоли;
- harddisks — этот файл предназначен для тонкой настройки производительности жестких дисков. Так же смотрите описание программы hdparm;
- hwconf — этот файл содержит базу обнаруженных и сконфигурированных устройств программой kudzu. Например:

  ```
  -
  class: OTHER
  bus: PCI
  ```

```
detached: 0
driver: agpgart
desc: "Intel Corporation|82815 815 Chipset Host Bridge and Memory Con-
troller Hub"
vendorId: 8086
deviceId: 1130
subVendorId: 8086
subDeviceId: 1130
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: unknown
desc: "Intel Corporation|unknown device 8086:1131"
vendorId: 8086
deviceId: 1131
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: unknown
desc: "Intel Corporation|82820 820 (Camino 2) Chipset PCI"
vendorId: 8086
deviceId: 244e
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: i810-tco
desc: "Intel Corporation|82820 820 (Camino 2) Chipset ISA Bridge
(ICH2)"
vendorId: 8086
```

```
deviceId: 2440
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: unknown
desc: "Intel Corporation|82820 820 (Camino 2) Chipset IDE U100"
vendorId: 8086
deviceId: 244b
subVendorId: 8086
subDeviceId: 244b
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: unknown
desc: "Intel Corporation|82820 820 (Camino 2) Chipset SMBus"
vendorId: 8086
deviceId: 2443
subVendorId: 8086
subDeviceId: 244b
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: btaudio
desc: "Brooktree Corporation|Bt878"
vendorId: 109e
deviceId: 0878
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: OTHER
```

```
bus: USB
detached: 0
driver: unknown
desc: "USB UHCI Root Hub"
usbclass: 9
usbsubclass: 0
usbprotocol: 0
usbbus: 1
usblevel: 0
usbport: 0
vendorId: 0000
deviceId: 0000
productrevision: unknown
-
class: OTHER
bus: USB
detached: 0
driver: unknown
desc: "USB UHCI Root Hub"
usbclass: 9
usbsubclass: 0
usbprotocol: 0
usbbus: 2
usblevel: 0
usbport: 0
vendorId: 0000
deviceId: 0000
productrevision: unknown
-
class: MOUSE
bus: PSAUX
detached: 0
device: psaux
driver: generic3ps/2
desc: "Generic 3 Button Mouse (PS/2)"
-
class: AUDIO
bus: PCI
detached: 0
```

```
driver: i810_audio
desc: "Intel Corporation|82801BA/BAM (ICH2) AC'97 Audio Controller"
vendorId: 8086
deviceId: 2445
subVendorId: 11d4
subDeviceId: 5360
pciType: 1
-
class: CDROM
bus: IDE
detached: 0
device: hdc
driver: ignore
desc: "SONY CD-ROM CDU4821"
-
class: VIDEO
bus: PCI
detached: 0
driver: Card:NVIDIA GeForce 2 MX (generic)
desc: "nVidia Corporation|NV11"
vendorId: 10de
deviceId: 0110
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: FLOPPY
bus: MISC
detached: 0
device: fd0
driver: unknown
desc: "3.5" 1.44MB floppy drive"
-
class: HD
bus: IDE
detached: 0
device: hda
driver: ignore
desc: "FUJITSU MPG3409AT E"
```

```
physical: 79428/16/63
logical: 4983/255/63
-
class: CAPTURE
bus: PCI
detached: 0
driver: bttv
desc: "Brooktree Corporation|Bt878"
vendorId: 109e
deviceId: 036e
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: USB
bus: PCI
detached: 0
driver: usb-uhci
desc: "Intel Corporation|82820 820 (Camino 2) Chipset USB (Hub A)"
vendorId: 8086
deviceId: 2442
subVendorId: 8086
subDeviceId: 244b
pciType: 1
-
class: USB
bus: PCI
detached: 0
driver: usb-uhci
desc: "Intel Corporation|82820 820 (Camino 2) Chipset USB (Hub B)"
vendorId: 8086
deviceId: 2444
subVendorId: 8086
subDeviceId: 244b
pciType: 1
-
class: MODEM
bus: SERIAL
detached: 1
```

```
device: ttyS0
driver: ignore
desc: "IDC|5620 IDC 5614BXL VR PnP"
pnpmfr: IDC
pnpmodel: 5620
pnpcompat: *PNPC107
pnpdesc: IDC 5614BXL VR PnP
```

- i18n — файл, отвечающий за локализацию системы. Название расшифровывается как internationalization — между i и n восемнадцать букв. Пример:

```
LANG="ru_RU.koi8r"
SUPPORTED="ru_RU.koi8r:ru_RU:ru"
SYSFONT="cyr-sun16"
SYSFONTACM="koi8-u"
```

  В примере определено, что у нас русская локаль с кодировкой KOI8-R, использующая шрифт cyr-sun16;

- identd — конфигурационный файл демона identd, реализующего поддержку протокола идентификации пользователя;

- keyboard — этот файл, находящийся в каталоге /etc/sysconfig, отвечает за конфигурирование клавиатуры. Пример файла keyboard приведен ниже.

  Для настройки клавиатуры (раскладки и скорости повтора) необходимо произвести следующие операции:

  - определиться с раскладкой клавиатуры (описания раскладки клавиатуры находятся в каталоге /usr/lib/kbd/keytables/ в файлах с расширением map);
  - внести изменения в файл keyboard таким образом, чтобы он содержал строку: `KEYTABLE="/usr/lib/kbd/keytables/zzz.map"`, где `zzz` — имя раскладки клавиатуры;
  - для настройки скорости повтора нажатия и время задержки необходимо добавить следующую строку в файл /etc/rc.d/rc.sysinit или, если у вас Caldera, к /etc/rc.d/rc.boot: `/sbin/kbdrate -s -r 16 -d 500` — где `-r 16` — количество символов, а `-d 500` — задержка в миллисекундах.

  Пример стандартного файла:

```
KEYBOARDTYPE="pc"
KEYTABLE="ru"
```

- kudzu — файл, отвечающий за поведение программы kudzu при перезапуске системы — как она будет себя вести при обнаружении нового оборудования;

- mouse — этот файл определяет параметры мыши, эмуляцию нажатия третьей кнопки и файл устройства. Пример файла mouse:

```
MOUSETYPE="imps2"
XMOUSETYPE="IMPS/2"
FULLNAME="Microsoft IntelliMouse (PS/2)"
XEMU3=no
DEVICE=/dev/mouse
```

- sendmail — файл, определяющий как стартует программа sendmail и через сколько времени отправляется почтовая очередь;
- squid — этот файл отвечает за настройки программы squid;
- syslog — этот файл отвечает за настройку демона syslog;
- xinetd — этот файл отвечает за настройку демона xinetd, который в Linux заменяет inetd.

/etc/X11 — конфигурационные файлы для X Window System

Каталог содержит конфигурационные файлы X11, специфичные для данного хоста. Если соответствующие пакеты установлены, в каталоге должны находиться следующие файлы или символические связи:

- Xconfig — конфигурационный файл для ранних версий XFree86;
- XF86Config — конфигурационный файл для XFree86 версии 3 и 4;
- Xmodmap — глобальный файл клавиатурных раскладок X11.

/etc/sgml — конфигурационные файлы для SGML и XML

Каталог содержит базовые конфигурационные файлы для определения параметров высокого уровня для SGML или XML. Файлы с именами *.conf обозначают базовые конфигурационные файлы. Файлы с именами *.cat — DTD-специфичные централизированные каталоги, содержащие руководства по всем остальным каталогам.

Каталог /home — пользовательские домашние каталоги

В каталоге находятся домашние каталоги пользователей. Как правило, каждый пользователь каталога в небольшой системе имеет свой домашний каталог, и имя домашнего совпадает с именем (login) пользователя. Например, у пользователя frozzy домашний каталог — /home/frozzy. Типичное содер-

жимое каталога пользователя, только что зарегистрированного в системе, включает следующие файлы и каталоги:

/cedit	/.gnome-desktop	/.netscape	.bash_profile
/Desktop	/.gnome_private	/nsmail	.bashrc
/.dia	/.gnp	/.sawfish	.ICEauthority
/.gimp-1.2	/kde	.bash_history	.screenrc
/.gnome	/.mc	.bash_logout	

Как можно видеть, это, в основном, конфигурационные файлы программ, установленных в операционной системе.

Каталог /lib — важные разделяемые библиотеки и модули ядра

Каталог содержит разделяемые библиотеки, необходимые для загрузки системы и запуска команд в корневой файловой системе, т. е. только для файлов, находящихся в каталоге /bin и /sbin. По меньшей мере одна из групп файлов должна находиться в каталоге /lib:

- libc.so.* — динамически подключаемая (линкуемая) библиотека C;
- ld* — линкер/загрузчик (linker/loader) времени выполнения.

Следующие каталоги также должны находится в каталоге /lib:

- /modules — загружаемые модули ядра;
- /security — модули PAM.

Каталог /lost+found

Каталог, который обязательно должен присутствовать на каждом разделе. (Если, к примеру, винчестер разбит на три раздела, которые монтируются в /, /home, /var, то в корневой файловой системе, в каталоге /home и в каталоге /var будет присутствовать /lost+found.) Назначение этого каталога достаточно очевидно — при аварийных ситуациях возможна потеря информации. Специальная утилита chkfsk восстанавливает (если, конечно, это возможно) утерянную информацию. Однако иногда невозможно достоверно определить принадлежность восстановленных данных какому-нибудь определенному файлу. В этом случае восстановленные данные помещаются в каталог /lost+found.

Каталог /misc — точка монтирования автоматически монтируемых устройств

Каталог предназначен для использования в качестве точки монтирования дискет и CD-ROM программой automount.

Каталог /mnt — точка монтирования для временно монтируемой файловой системы

Каталог предназначен для того, чтобы системный администратор мог временно монтировать файловую систему (например, дискету или CD-ROM). В различных дистрибутивах Linux в каталоге /mnt могут находиться каталоги, являющиеся точками монтирования дискет, разделов жесткого диска, CD-ROM и т. п. Например, в только что установленном Red Hat 7.1 в каталоге /mnt находятся каталоги /cdrom и /floppy, которые являются точками монтирования для CD-ROM и дискет. Если в каталоге /mnt находятся какие-то файлы и к каталогу /mnt монтируется некий раздел, то файлы, находящиеся в каталоге /mnt, становятся недоступны до тех пор, пока не размонтируют раздел, подмонтированный к /mnt.

Каталог /opt — дополнительные программные пакеты

Каталог зарезервирован для инсталляции дополнительного программного обеспечения. Пакет, который устанавливается в каталог /opt, должен хранить свои неизменяемые файлы в каталоге /opt/<имя_пакета>, где <имя_пакета> — имя устанавливаемого пакета. Структура поддерева каталогов в каталоге <имя_пакета>:

- /bin;
- /doc;
- /lib;
- /man и т. д.

Исполняемые модули надо размещать в каталоге /bin, а если пакет включает в себя документацию, ее надо сохранить в каталоге /doc. При наличии страниц справочной системы, размещать их в /opt/<имя_пакета>/man и использовать подструктуру каталогов, как в /usr/share/man. Специфичные для конкретного пакета библиотеки размещаются в /opt/<имя_пакета>/lib и т. д. Файлы пакета, которые могут изменяться, должны быть установлены в каталоге /var/opt. Хост-специфичные конфигурационные файлы должны быть установлены в /etc/opt.

Каталог /proc — точка монтирования виртуальной файловой системы procfs

Procfs является псевдофайловой системой, обеспечивающей интерфейс с ядром Linux. Эта система позволяет получить доступ к определенным структурам данных ядра, в частности, к списку процессов (отсюда и название). Все эти структуры выглядят как файловая система, и ими можно оперировать обычными средствами работы с файловой системой.

Структура каталогов в /proc:

- /1 — подкаталог процесса, имя каталога соответствует номеру PID-процесса;
- /2;
- /3;
- /4;
- /5;
- /6;
- /7;
- /384;
- /389;
- /403;
- /418;
- /490;
- /5196;
- /bus — каталог содержит специфичную информацию, касающуюся шин (PCI, ISA);
- /driver — здесь сгруппированы различные драйверы;
- /fs — каталог содержит параметры файловых систем;
- /ide — каталог содержит информацию о IDE-подсистеме;
- /irq — маски для управления аппаратными прерываниями;
- /net — сетевая информация;
- /sys — системная информация;
- /sysvipc — информация о SysVIPC-ресурсах (msg, sem, shm);
- /tty — информация о TTY-драйверах;
- apm — расширенная информация управлением питанием;
- cmdline — командная строка ядра операционной системы;
- cpuinfo — информация о микропроцессоре;
- devices — доступные устройства (блочные и символьные);
- dma — используемые каналы DMA;
- execdomains — используемые домены;
- fb — Frame Buffer-устройства;
- filesystems — поддерживаемые файловые системы;
- interrupts — используемые прерывания;

- iomem — карта памяти;
- ioports — используемые порты ввода/вывода;
- isapnp — информация о ISA-устройствах;
- kcore — образ ядра операционной системы;
- kmsg — сообщения ядра;
- ksyms — таблица символов ядра;
- loadavg — средняя загрузка за последние 1, 5 и 15 минут;
- locks — "защелки" ядра;
- mdstat— файл, сообщающий о конфигурации RAID-массива системы;
- meminfo — информация о памяти;
- misc — различная информация, не попавшая не в одну из категорий;
- modules — список загруженных модулей;
- mounts — смонтированные файловые системы;
- mtrr — управление использованием памяти;
- partitions — список разделов, известных системе;
- pci — устаревшая информация о PCI-шине (см. /proc/bus/pci/);
- rts — часы реального времени;
- scsi — информация о SCSI-устройствах;
- self — символическая ссылка к каталогу процесса, пытающегося получить информацию из /proc;
- slabinfo — информация о Slab;
- stat — разнообразная статистика;
- swaps — использование разделов и файлов подкачки;
- uptime — время работы системы без перезагрузки;
- version — версия ядра;
- video — BTTV-информация о видеоресурсах.

/proc/№процесса_PID-процесса

Каталог имеет имя, соответствующее номеру PID-процесса. Каждый процесс в системе имеет соответствующий ему каталог в /proc. В этом каталоге обязательно находятся следующие файлы:

- cmdline — файл, содержащий аргументы командной строки процесса;
- cpu — текущий и последний использовавшийся микропроцессор (только для мультипроцессорных систем);
- /cwd — ссылка на текущий рабочий каталог;

- environ — содержит значения переменных окружения;
- exe — ссылка на исполняемый файл этого процесса;
- /fd — каталог, содержащий все файловые дескрипторы данного процесса;
- maps — карты памяти исполняемых и библиотечных файлов;
- mem — память, занятая этим процессом;
- /root — ссылка на корневой каталог этого процесса;
- stat — статус процесса;
- statm — информация об использовании процессом памяти;
- status — статус процесса в форме, воспринимаемой человеком.

/proc/ide — IDE-устройства, установленные в системе

В каталоге содержится информация обо всех установленных в системе IDE-устройствах, в том числе используемые драйверы.

/proc/net — сетевая информация

В этом каталоге содержится информация, относящаяся к сети. Следующие файлы являются общими как для протокола IPv4, так и IPv6:

- arp — ARP-таблица ядра;
- dev — сетевые устройства со своей статистикой;
- dev_stat — статус сетевого устройства;
- ip_fwchains — связи цепочки Firewall;
- ip_fwnames — имена цепочек Firewall;
- /ip_masq — каталог содержит таблицы маскарадинга[1];
- ip_masquerade — главная таблица маскарадинга;
- netstat — сетевая статистика;
- raw — статистика сетевых устройств;
- route — таблица маршрутизации ядра;
- /rpc — каталог содержит RPC-информацию;
- rt_cache — кэш маршрутизации;
- snmp — данные SNMP;
- sockstat — статистика сокетов;

[1] Маскарадинг — подмена реального IP-адреса любого исходящего пакета на другой (специальный), а для входящего пакета — замена IP-адреса (специального) с помощью таблицы соответствия на реальный адрес компьютера, к которому адресован сетевой пакет.

- tcp — TCP-сокеты;
- tr_rif — таблица маршрутизации Token ring RIF;
- udp — UDP-сокеты;
- unix — UNIX-сокеты;
- wireless — данные беспроводного интерфейса (Wavelan и т. п.);
- igmp — IP-адреса, которые хост принимает;
- psched — параметры глобального администратора пакетов;
- netlink — список PF_NETLINK-сокетов;
- ip_mr_vifs — список виртуальных интерфейсов;
- ip_mr_cache — список кэша маршрутизации.

Файлы, приведенные ниже, используются протоколом IPv6:

- udp6 — UDP-сокеты (IPv6);
- tcp6 — TCP-сокеты (IPv6);
- raw6 — статистика устройств (IPv6);
- igmp6 — IP-адреса, принимаемые хостом (IPv6);
- if_inet6 — список IPv6-интерфейсных адресов;
- ipv6_route — таблица маршрутизации для IPv6;
- rt6_stats — общая статистика IPv6-таблиц маршрутизации;
- sockstat6 — статистика сокетов (IPv6);
- snmp6 — SNMP-данные (IPv6).

/proc/parport — параллельные порты

Каталог содержит информацию обо всех параллельных портах, установленных в системе.

/proc/scsi — SCSI-устройства, установленные в системе

Если в компьютере установлены SCSI-устройства, то должен существовать каталог /proc/scsi. В нем содержится информация обо всех установленных в системе SCSI-устройствах, в том числе используемые драйверы.

/proc/sys — системная информация

Этот каталог содержит файлы, изменением которых можно, не перегружая системы, изменять различные параметры ядра.

/proc/sys/dev — информация, специфическая для устройств

На сегодняшний день поддерживаются только устройства CD-ROM.

/proc/sys/fs — данные файловой системы

Каталог содержит различную информацию о файловой системе.

/proc/sys/kernel — основные параметры ядра операционной системы

В этом каталоге находятся файлы, с помощью которых можно изменять настройки ядра операционной системы.

/proc/sys/net — сетевая "начинка"

Этот каталог содержит интерфейс по управлению различными сетевыми протоколами. В этом каталоге могут находиться следующие подкаталоги:

- /802 — протокол E802;
- /appletalk — Appletalk-протокол;
- /ax25 — AX25;
- /bridge — Bridging;
- /core — основные параметры;
- /decnet — DEC-net;
- /ethernet — Ethernet-протокол;
- /ipv4 — IP версии 4;
- /ipv6 — IP версии 6;
- /ipx — IPX;
- /netrom — NET/ROM;
- /rose — X.25 PLP layer;
- /token-ring — IBM token ring;
- /unix — UNIX domain sockets;
- x25 — протокол X.25.

/proc/sys/sunrpc — удаленные вызовы процедур

Каталог содержит файлы, которые разрешают или запрещают отладку удаленно вызываемых процедур.

/proc/sys/vm — виртуальная подсистема памяти

Файлы в этом каталоге используются для настройки виртуальной подсистемы памяти ядра Linux.

/proc/tty — терминалы

Здесь содержится информация о доступных и используемых терминалах.

Каталог /root — домашний каталог для пользователя root (администратора)

Существенных причин для вынесения домашнего каталога /root в корневой уровень нет. Однако существует практика выделения отдельного раздела для каталога /home, который при аварийных ситуациях может не подмонтироваться. По-видимому, по этой причине каталог /root вынесли на корневой уровень.

Каталог /sbin — системные исполняемые файлы

Утилиты, используемые для системного администрирования, и другие, используемые только администратором (пользователем root), хранятся в каталогах /sbin, /usr/sbin и /usr/local/sbin. Каталог /sbin содержит исполняемые файлы, необходимые для загрузки, восстановления, починки системы в добавление к файлам, находящимся в каталоге /bin. Программы, используемые после монтирования файловых систем, в основном помещаются в каталог /usr/sbin. Административные программы, используемые только на локальной системе, помещаются в каталог /usr/local/sbin.

Обычные пользователи не должны иметь доступа в каталоги /sbin. Если обычный пользователь (не администратор) может запускать команду, она должна находиться в одном из каталогов /bin. В каталоге /sbin должны присутствовать следующие файлы:

- badblocks — утилита для проверки жестких дисков;
- ctrlaltdel — программа для перезагрузки операционной системы;
- dumpe2fs — утилита для работы с файловой системой;
- e2fsck — утилита для проверки файловой системы;
- fastboot — утилита, перезагружающая систему без проверки дисков;
- fasthalt — утилита, останавливающая систему без проверки дисков;
- fdisk — утилита, позволяющая производить различные действия с таблицей разделов (создавать, редактировать, удалять раздел и т. д.);
- fsck — утилита, проверяющая и восстанавливающая файловую систему;
- fsck.* — утилита, проверяющая и восстанавливающая файловую специфичную систему (например, Ext2);
- getty — программа getty;
- halt — команда, останавливающая систему;
- ifconfig — утилита конфигурации сетевого интерфейса;
- init — Init-процесс;

- kbdrate — утилита для настройки клавиатуры;
- lilo — загрузчик операционной системы;
- mke2fs — утилита создания файловой системы;
- mkfs — команда, создающая файловую систему;
- mkfs.* — команда, создающая специфичную файловую систему;
- mkswap — команда, устанавливающая своп-область;
- reboot — команда, перегружающая систему;
- route — утилита для таблицы IP-маршрутизации;
- swapon — утилита, разрешающая свопирование;
- swapoff — утилита, запрещающая свопирование;
- tune2fs — утилита тонкой настройки файловой системы;
- update — демон, периодически сбрасывающий буферы файловой системы.

Каталог /tmp — временные файлы

Каталог должен быть доступен для программ, которые нуждаются во временных файлах.

При загрузке системы файлы, находящиеся в /tmp, должны удаляться (по крайней мере, рекомендуется).

Каталог /usr — иерархия

Каталог /usr — это вторая основная секция файловой системы, разделяемая, только для чтения. В каталоге /usr должны находиться следующие каталоги:

- /bin — содержит большую часть утилит, используемых пользователем;
- /include — файлы заголовков, включаемых в C-программы;
- /lib — библиотеки;
- /local — локальная иерархия;
- /sbin — содержит не жизненно необходимые системные исполняемые файлы;
- /share — архитектурно-независимые данные;
- /X11R6 — X Window System, версия 11, выпуск 6;
- /games — игры и образовательные программы;
- /src — исходные коды.

/usr/bin — пользовательские программы

В каталоге содержится большинство программ, предназначенных для пользователей. В частности, здесь должны находиться следующие программы (если установлены соответствующие пакеты):

- perl — интерпретатор языка Perl;
- python — интерпретатор языка Python;
- tclsh — интерпретатор Tcl;
- wish — простая оконная оболочка Tcl/Tk;
- expect — программа для интерактивного диалога.

/usr/include — каталог для стандартных include-файлов

В этом каталоге хранится большинство включаемых файлов, используемых компилятором C/C++.

/usr/lib — библиотеки для программирования и пакетов

Каталог содержит объектные файлы, библиотеки и другие файлы, которые не используются напрямую пользователем или скриптами командных оболочек. Если программа создает подкаталог в /usr/lib, все архитектурно-зависимые данные должны помещаться в этот каталог. Для примера: подкаталог /perl5 содержит в себе модули и библиотеки для Perl 5.

/usr/local — локальная иерархия

Каталог предназначен для системного администратора под установку локального программного обеспечения. Это необходимо для предотвращения перезаписи программного обеспечения при обновлении системного программного обеспечения. Содержит следующие каталоги:

- /bin — локальные исполняемые файлы;
- /games — локальные исполняемые файлы игр;
- /include — локальные файлы C-заголовков;
- /lib — локальные библиотеки;
- /sbin — локальные системные исполняемые файлы;
- /share — локальная архитектурно-независимая иерархия;
- /src — локальный исходный код.

/usr/sbin — не жизненно необходимые стандартные системные программы

Каталог содержит любые не жизненно необходимые для функционирования системы исполняемые файлы, используемые исключительно системным ад-

министратором. Программы и утилиты, используемые при восстановлении работоспособности системы, должны находиться в каталоге /sbin.

/usr/share — архитектурно-независимые данные

Каталог предназначен для всех архитектурно-независимых файлов данных, предназначенных только для чтения (неизменяемых). Содержит следующие каталоги:

- /dict — списки слов (словари);
- /doc — разнообразная документация;
- /games — неизменяемые файлы данных для /usr/games;
- /info — основной каталог информационной системы GNU;
- /locale — информация для локализации системы;
- /man — файлы справочной системы;
- /misc — разнообразные архитектурно-независимые данные;
- /terminfo — каталог для базы данных terminfo;
- /zoneinfo — информация и конфигурация временной зоны (Timezone).

Любая программа или пакет, который содержит или требует данных, не нуждающихся в модификации должны хранится в /usr/share (или /usr/local/share, если программное обеспечение установлено локально).

/usr/share/dict — списки слов (словари)

Каталог содержит словари, находящиеся в системе. Традиционно в этом каталоге находится только файл с английскими словами, которые используются программой look и многими программами проверки правописания. В этот каталог можно установить свои файлы, например, с русскими словами.

/usr/share/man — страницы справочной системы

Каталог предназначен для хранения данных справочной системы. Вся справочная информация разделена на восемь больших тем, для каждой существует свой отдельный каталог — от /man1 до /man8. Содержит следующие каталоги:

- /man1 — справочные страницы, описывающие доступные пользователям программы;
- /man2 — раздел, описывающий все системные вызовы (для взаимодействия с ядром);
- /man3 — библиотечные функции и подпрограммы. Описывает программные библиотеки, напрямую не взаимодействующие с ядром операционной системы. Этот и второй разделы справочной системы представляют интерес только для программистов;

- /man4 — описывает специальные файлы, осуществляющие функции драйверов и сетевой поддержки в системе. В основном эти файлы находятся в каталоге /dev;
- /man5 — документация по множеству файловых форматов;
- /man6 — содержится документация по разнообразным играм;
- /man7 — разное. Содержит документацию, которую трудно классифицировать;
- /man8 — системное администрирование. Программы, используемые системным администратором для администрирования и сопровождения системы.

Система справочной информации должна поддерживать несколько языков одновременно, поэтому для исключения конфликтов в каталоге /usr/share/man файлы справочной системы принято хранить следующим образом:

- для каждого языка, установленного в системе (locale, локаль), в каталоге /usr/share/man создается подкаталог, носящий имя своей локали;
- в этом подкаталоге создаются каталоги /man<раздел>, причем только те, в которых есть справочная информация;
- в каталоге /man<раздел> хранятся справочные файлы, отдельные для каждой установленной программы, причем стандартом de-facto является то, что справочные файлы хранятся в архивированном виде (никто, однако, не запрещает хранить их в распакованном виде, но для экономии места на жестком диске их упаковывают).

Программа man при обращении к ней с целью получения справочной информации по какой-то программе сначала пытается получить справочную информацию на языке, соответствующем текущей локали. Если ей это не удается, то берется информация, хранящяяся в /usr/share/man/man<раздел>. По умолчанию в этих каталогах содержится англоязычная справочная информация.

Наименование языковых подкаталогов в /usr/share/man основывается на приложении E стандарта POSIX 1003.1, который описывает строку-идентификатор локали. Строка-идентификатор локали согласно этому стандарту имеет вид:

```
<язык>[_<территория>][.<кодовая страница символов>][,<версия>]
```

- поле `<язык>` берется из стандарта ISO 639. Это должны быть два символа исключительно в нижнем регистре;
- поле `<территория>` должно быть двухсимвольным кодом только в верхнем регистре (согласно стандарту ISO 3166);

- поле <кодовая страница символов> должно быть представлено в стандартном описании кодовой страницы. Если <кодовая страница символов> содержит числовую спецификацию, она соответствует интернациональному стандарту, описывающему эту страницу;
- поле <версия> рекомендуется не использовать без крайней необходимости. Реальное его применение — например, для страны, имеющей один язык и кодировку, но разные диалекты.

Пример формирования каталогов локализованной справочной системы приведен в табл. 5.3.

Таблица 5.3. *Пример формирования каталогов локализованной справочной системы*

Язык	Территория	Кодовая страница символов	Каталог
Английский	—	ASCII	/usr/share/man/en
Английский	Великобритания	ASCII	/usr/share/man/en_GB
Английский	США	ASCII	/usr/share/man/en_US
Французский	Франция	ISO 8859-1	/usr/share/man/fr_FR
Французский	Канада	ISO 8859-1	/usr/share/man/fr_CA
Русский	СНГ	KOI8-R	/usr/share/man/ru_RU

Архитектурно-зависимые справочные файлы можно помещать в отдельные каталоги, соответствующие архитектуре. Например, /usr/share/man/<locale>/man8/i386/ctrlaltdel.8. Однако проще написать общее справочное руководство, в котором особо отметить архитектурно-зависимые случаи, чем разрабатывать справочные файлы для каждой архитектуры.

Справочная информация для программ и данных, находящихся в /usr/local, размещается в каталоге /usr/local/man. Справочная информация, касающаяся X11R6, размещается в каталоге /usr/X11R6/man.

Правило размещения справочных руководств на различных языках в отдельные подкаталоги также распространяется и на справочные руководства, хранящиеся в каталогах /usr/local/man и /usr/X11R6/man.

/usr/share/misc — различные архитектурно-независимые данные

Каталог содержит различные архитектурно-независимые файлы, которые не требуют отдельного каталога в /usr/share/. Если соответствующие пакеты установлены в системе, в каталоге должны находиться следующие файлы:

- ascii — ASCII-таблица символов;

- magic — список "магических" цифр;
- termcap — база данных совместимости терминалов.

/usr/src — исходные тексты программ

Любой исходный код нелокальной программы должен помещаться в этот каталог.

/usr/src/Linux-2.4.3 — каталог исходного кода ядра Linux

В каталоге хранятся файлы и каталоги, содержащие исходный код ядра Linux, модулей, различная документация. Имя каталога меняется в зависимости от того, исходный код какой версии ядра Linux находится в каталоге.

/usr/src/Linux-2.4.3/Documentation — документация к ядру и модулям операционной системы Linux

В каталоге содержится документация, которая тем или иным образом касается ядра операционной системы Linux или загружаемых модулей. Типичное содержимое каталога приведено ниже:

/arm	/sysctl	dnotify.txt	kmod.txt
/cdrom	/telephony	exception.txt	locks.txt
/cris	/video4linux	floppy.txt	logo.gif
/DocBook	/vm	ftape.txt	logo.txt
/fb	/usb	hayes-esp.txt	LVM-HOWTO
/filesystems	00-INDEX	highuid.txt	magic-number.txt
/i2c	binfmt_misc.txt	ide.txt	mandatory.txt
/i386	BUG-HUNTING	initrd.txt	mca.txt
/ia64	cachetlb.txt	ioctl-number.txt	md.txt
/isdn	cciss.txt	IO-mapping.txt	memory.txt
/kbuild	Changes	IRQ-affinity.txt	mkdev.cciss
/m68k	CodingStyle	isapnp.txt	mkdev.ida
/mips	computone.txt	java.txt	modules.txt
/networking	Configure.help	joystick-api.txt	moxa-smartio
/parisc	cpqarray.txt	joystick-parport.txt	mtrr.txt
/powerpc	devices.txt	joystick.txt	nbd.txt
/s390	digiboard.txt	kernel-doc-nano-HOWTO.txt	nfsroot.txt
/sound	digiepca.txt	kernel-docs.txt	nmi_watchdog.txt
/sparc	DMA-mapping.txt	kernel-parameters.txt	oops-tracing.txt

paride.txt	README.moxa	sgi-visws.txt	SubmittingPatches
parport-lowlevel.txt	README.nsp_cs.eng	smart-config.txt	svga.txt
parport.txt	riscom8.txt	smp.tex	sx.txt
pci.txt	rtc.txt	smp.txt	sysrq.txt
pcwd-watchdog.txt	SAK.txt	specialix.txt	unicode.txt
pm.txt	scsi-generic.txt	spinlocks.txt	VGA-softcursor.txt
ramdisk.txt	scsi.txt	stallion.txt	watchdog.txt
README.DAC960	serial-console.txt	SubmittingDrivers	xterm-linux.xpm
			zorro.txt

/usr/X11R6 — X Window System, Version 11 Release 6

В каталоге X11R6 содержится иерархия каталогов X Window. Информацию о структуре и назначении каталогов следует искать в документации на X Window.

Каталог /var

Каталог содержит изменяемые файлы. Сюда входят spool-каталоги и файлы, административные и журнальные данные, временные файлы. Некоторые каталоги, входящие в иерархию /var, такие как /var/log, /var/lock и /var/run, не должны быть разделяемыми между различными системами. Другие каталоги, такие как /var/mail, /var/cache/man, /var/cache/fonts и /var/spool/news, могут быть разделяемыми.

Рекомендуется для каталога /var выделить отдельный раздел на жестком диске. В том случае, если это невозможно, не следует размещать его в корневой файловой системе. Это позволит избежать некоторых проблем, возникающих при переполнении диска. Приложения не должны создавать каталоги в верхнем уровне иерархии /var. В каталоге /var должны присутствовать следующие каталоги:

- /cache — каталог кэша программ;
- /db — каталог для файлов баз данных;
- /games — файлы для игровых программ;
- /lib — библиотеки;
- /local — изменяемые данные для /usr/local;
- /lock — Lock-файлы (файлы-защелки);
- /log — Log-файлы и каталоги (файлы журналов);
- /lost+found — каталог для файлов, восстановленных после краха системы;

- /mail — каталог, содержащий почтовые ящики пользователей;
- /named — файлы DNS-сервера;
- /opt — переменные данные для /opt;
- /run — данные о запущенных процессах;
- /spool — spool-данные приложений;
- /state — состояние приложений;
- /tmp — временные файлы, сохраняемые между перезагрузками системы.

/var/cache — кэш программ

Каталог используется для хранения временных "короткоживущих" данных, создаваемых программами. Это могут быть буферы ввода/вывода или файлы, содержащие какие-нибудь промежуточные данные. Подкаталоги в /var/cashe создаются при установке пакетов и обычно носят имя соответствующей программы.

Если соответствующие пакеты установлены в системе, в каталоге должны находиться следующие файлы:

- /fonts — каталог для хранения динамически создаваемых шрифтов;
- /man — сформатированные страницы руководств. Справочные страницы в /usr/man хранятся в специальном виде и перед тем, как показать справочное руководство пользователю, страницы необходимо сформатировать;
- /www — файлы или кэш-данные прокси-сервера WWW;
- /<пакет> — кэш соответствующего пакета.

/var/games — файлы для игровых программ

В этом каталоге должны храниться файлы, которые могут изменяться, например файлы, содержащие таблицы результатов, файлы сохраненных игр и т. п.

/var/lib — библиотеки

Немного неверное наименование раздела. В этом каталоге содержатся различные файлы, входящие в какие-либо пакеты, которые можно отнести к системным. Обычно каждый пакет, который сохраняет какие-то файлы в каталог /var/lib, создает свой каталог, имеющий вид /var/lib<имя_пакета>. Если соответствующие пакеты установлены в системе, в каталоге должны находиться следующие файлы:

- /misc — разные несистематизированные файлы;
- /<редактор> — каталог соответствующего редактора, в котором хранятся резервные копии файлов и файлы состояния;

- /rpm — каталог для менеджера пакетов RPM. В нем содержатся базы установленных в системе пакетов и другая служебная информация;
- /<пакет> — файлы соответствующего пакета;
- /xdm — данные X-менеджера.

/var/lock — lock-файлы (файлы-защелки)

Lock-файлы (файлы-защелки) — файлы, которые "закрепляют" какое-либо оборудование или файлы для использования только программой, создающей файл-защелку. Обычно уничтожаются по окончании работы программы или если файл или оборудование не нужны в данный момент программе. В каталоге /var/lock могут находиться, например, следующие подкаталоги:

- /console — данные, относящиеся к консоли системы;
- /samba — данные, связанные с программой Samba.

/var/log — файлы и каталоги журналов (log-файлов)

Каталог содержит разнообразные файлы журналов. Также для некоторых пакетов используются каталоги, в которых хранятся соответствующие файлы журналов. Если соответствующие пакеты установлены в системе, в каталоге должны находиться следующие файлы:

- /httpd — каталог для журнальных файлов Web-сервера;
- /samba — каталог для журнальных файлов сервера Samba;
- /squid — каталог для журнальных файлов SQUID;
- /uucp — каталог для журнальных файлов UUCP.

Также в каталоге /var/log должны находиться следующие файлы:

- cron — события демона cron;
- dmesg — сообщения в течение дня;
- lastlog — записи о последней регистрации в системе каждого пользователя;
- maillog — регистрация событий, связанных с почтовыми сообщениями;
- messages — системные сообщения от syslogd;
- secure — сообщения, связанные с безопасностью;
- statistics — файл статистики;
- usracct — файл активности пользователей;
- wtmp — записи всех logins и logouts;
- boot.log — журнал загрузки системы;
- htmlaccess.log — журнал доступа к Web-серверу;
- XFree86.0.log — журнал XFree86.

/var/mail — пользовательские почтовые ящики

Этот каталог хранит пользовательские почтовые ящики, сохраненные в стандартном формате UNIX mailbox.

/var/opt — изменяемые данные для каталога /opt

В этом каталоге должны храниться изменяемые данные пакетов, устанавливаемые в каталог /opt. Рекомендуется для каждого пакета создать свой каталог вида /opt/<имя_пакета>.

/var/run — переменные файлы времени исполнения

Каталог содержит системную информацию, описывающую состояние системы. Файлы в этом каталоге при загрузке системы должны быть удалены или усечены до нулевого размера. Программы, если это им необходимо, могут иметь подкаталоги, при условии, что эти программы создают во время функционирования более чем один файл (однако, например, демон FTP создает следующие файлы: ftp.pids-all, ftp.pids-local, ftp.pids-other, а отдельного каталога не имеет).

В каталоге, в основном, содержатся файлы-идентификаторы процессов (PID, Process identifie file), имеющие имя <имя_программы>.pid. К примеру, /var/run/named.pid. Pid-файл должен содержать символы, соответствующие номеру PID и символ перевода строки.

Каталог /var/run должен быть недоступен для записи непривилегированными пользователями, поскольку запись информации или ее удаление из каталога /var/run может привести к печальным последствиям, вплоть до краха системы.

/var/spool — spool-данные приложений

Каталог /var/spool — содержит данные, которые ожидают какой-либо обработки. После обработки (программой, пользователем, администратором) они должны быть удалены из каталога. Если соответствующие пакеты установлены в системе, в каталоге должны находиться следующие файлы:

- /at — spool-каталог программы at;
- /cron — spool-каталог программы cron;
- /lpd — spool-каталог программы печати;
- /mail — каталог входящей почты;
- /mqueue — исходящая почтовая очередь;
- /news — spool-каталог сервера новостей;
- /samba — spool-каталог сервера Samba;

- /squid — spool-каталог SQUID;
- /uucp — spool-каталог для UUCP.

/var/tmp — временные файлы, сохраняемые между перезагрузками

Каталог /var/tmp используется для того, чтобы временные файлы, необходимые для программ, сохранялись при перезагрузке системы. Файлы, находящиеся в /tmp, при перезагрузке системы могут быть удалены.

/var/yp — файлы баз данных Network Information Service (NIS) (опционально)

Если в системе установлена сетевая информационная служба (Network Information Service, NIS), так же известная, как Желтые страницы (Sun Yellow Pages, YP), то в этом каталоге хранятся ее базы данных.

Ссылки

- **http://www.pathname.com/fhs/** — Filesystem Hierarchy Standard в различных текстовых форматах.
- **http://www.kernel.org/pub/linux/docs/device-list/devices.txt** — список устройств и специальных файлов.
- proc.txt — документация по файловой системе procfs. Входит в состав документации к ядру Linux.
- Соответствующие man-страницы.
- Соответствующие HOWTO *(см. гл. 13)*:
  - Networking-HOWTO;
  - SMB-HOWTO;
  - DNS-HOWTO;
  - LILO-HOWTO.

Глава 6

Процесс загрузки Linux

Для того чтобы достичь полного контроля над операционной системой крайне важно представлять себе, как происходит процесс ее загрузки.

Вот нажатием кнопки Power вы включили компьютер. Сначала специальная программа, зашитая в ПЗУ материнской платы, производит тестирование установленного в компьютере оборудования. В случае неудачи вы либо услышите из встроенного динамика компьютера серию гудков, либо программа тестирования оборудования выведет на дисплей предупреждающее сообщение.

Если система успешно прошла тестирование, на дисплее можно будет увидеть перечень установленного оборудования, емкость оперативной памяти и жесткого диска. После этого программа BIOS (Basic Input/Output System — базовая система ввода/вывода), хранящаяся в ПЗУ материнской платы, определит, с какого устройства будет происходить загрузка (например, с жесткого диска C:), и считает из первого сектора загрузочного диска короткую программу-загрузчик. Эта программа (LILO, GRUB) загружает с жесткого диска ядро Linux, которое имеет имя vmlinuz-x.y.z-a (где x.y.z — это номер версии ядра, например, 2.4.3, а строка a — признак сборки, может быть каким-то числом или словом) и находится в каталоге /boot (для Red Hat-подобных дистрибутивов) или в корне файловой системы (для дистрибутива Slackware). Во время загрузки ядру можно передать различные параметры, позволяющие более тонко настроить систему (об этом немного позже). Сразу после загрузки ядро производит инициализацию устройств, установленных в компьютере. Затем пробует загрузить и монтировать корневую (root) файловую систему. Ядру необходимо тем или иным образом сообщить, где искать корневую файловую систему. Если ядро Linux не может ее найти, оно выдает соответствующее сообщение и останавливается.

Во многих дистрибутивах и практически всегда при загрузке с дискеты в оперативной памяти создается псевдодиск (RAM-disk, виртуальный диск), который и выступает в роли корневой файловой системы. Для этого есть две причины. Во-первых, оперативная память на несколько порядков быстрее, чем дискета, и во-вторых — на виртуальный диск ядро может загрузить с

дискеты и распаковать сжатую файловую систему, что позволяет поместить на дискете намного больше файлов.

После того как ядро Linux успешно смонтирует корневую файловую систему, оно запускает процесс init. Процесс init — это программа, которая, собственно, и осуществляет переход от начального состояния системы в стандартный многопользовательский режим (или тот, который установлен администратором по умолчанию). Помимо этого, процесс init выполняет множество различных операций, необходимых для корректной работы системы: проверку и монтирование файловых систем, запуск различных сервисов, запуск системы входа пользователя и т. п. А теперь подробнее разберемся с каждым шагом загрузки системы.

Программы-загрузчики

Используются для загрузки ядра операционной системы Linux, передачи параметров ядру и организации загрузки нескольких операционных систем, установленных на компьютере.

LILO — LInux LOader

Программа-загрузчик, на сегодняшний день является стандартом de-facto практически для любого дистрибутива Linux. LILO (LInux LOader — загрузчик Linux) без проблем может загружать DOS, OS/2, Linux, FreeBSD, Windows и множество других операционных систем.

Стандартно сконфигурированная программа LILO после запуска приостановит свое выполнение и выведет на экран графическое изображение с меню, пунктами которого являются варианты загрузки. Несколько секунд LILO ожидает ввода пользователем варианта загрузки (или специальных команд) и, в случае их отсутствия, запускает вариант загрузки, выбранный при конфигурировании по умолчанию. Обычные варианты загрузки в LILO носят название linux и dos (если в системе установлены одновременно операционная система Windows 9*x* и Linux).

Конфигурационный файл LILO — /etc/lilo.conf, формат его можно найти в соответствующей справочной документации.

GRUB

GRand Unified Bootloader (Главный унифицированный загрузчик) — универсальный загрузчик, разработан в Фонде свободного программного обеспечения. Имеет больше возможностей по сравнению с LILO, а также избавлен от некоторых ограничений. В последнее время стал очень популярен, и по распространенности скоро догонит LILO.

LoadLin

Еще одна программа запуска Linux, которая, правда, не используется при старте компьютера, а позволяет загрузить ядро Linux из командной строки DOS (с параметрами загрузки). Применение LoadLin оправдано в том случае, если мы не хотим устанавливать загрузчик типа LILO в MBR (Master Boot Record) винчестера. Этот загрузчик также необходим, если у нас имеется оборудование, использующее драйвер DOS для установки в определенное состояние.

Есть еще несколько других программ, которые можно использовать для загрузки Linux, однако они не получили широкого распространения.

Параметры ядра

Обзор параметров строки загрузки

Программы-загрузчики, описание которых приведено выше, способны также, помимо загрузки самого ядра, передавать ему необходимые параметры загрузки.

В параметрах загрузки недопустимо использовать пробелы, кроме как между отдельными аргументами. Список значений для одного аргумента должен разделяться запятыми между значениями. К примеру:

```
ether=9,0x300,0xd0000,0xd4000,eth0  root=/dev/hda1
```

Посмотреть параметры командной строки, заданные при загрузке, можно набрав `/proc/cmdline`.

Утилита rdev

Есть несколько параметров загрузки ядра Linux, хранящих свои значения по умолчанию в его образе. Эти параметры задаются при компиляции ядра, и для того, чтобы не перекомпилировать каждый раз ядро, используется утилита rdev.

Утилита rdev может изменять следующие параметры:

- rdev — устройство, с которого производится загрузка;
- swapdev — устройство, содержащее раздел подкачки (swap);
- ramsize — параметры RAM-диска;
- vidmode — видеорежим по умолчанию;
- rootflags — установка режима монтирования корневого устройства ("только для чтения" или "чтение/запись").

Более подробную информацию по rdev можно найти в соответствующей справочной документации.

Разбор параметров ядром Linux

Большая часть параметров загрузки имеет вид:

```
имя[=значение_1][,значение_2]...[,значение_11]
```

где `имя` — уникальное ключевое слово, идентифицирующее часть ядра, которому передаются связанные значения, но не более одиннадцати параметров. Бóльшая часть разбора параметров загрузки происходит в linux/init/main.c. Сначала ядро проверяет, не являются ли параметры одним из специальных параметров `root=`, `ro`, `rw` или `debug`. Затем ядро просматривает список функций установки (находящийся в массиве bootsetups) в поиске совпадения заданной строки параметра с функцией установки конкретного устройства или части ядра. Если мы передаем ядру строку `foo=3,4,5,6,bar`, то ядро будет искать, присутствует ли `foo` в массиве bootsetups. Если присутствует, то ядро вызовет функцию установки, связанную с `foo` (`foo_setup()`) и передаст ей целочисленные значения `3`, `4`, `5` и `6`, указанные в командной строке ядра, и также строковый параметр `bar`.

Если строка не подходит ни для одной функции установки, то этот случай считается установкой переменной окружения. Примером может служить указание переменных окружения `TERM=vt100` или `BOOT_IMAGE=vmlinuz.bak` в качестве параметров загрузки. Как правило, переменные окружения проверяются скриптами инициализации для разрешения или запрещения большого диапазона параметров.

Любые оставшиеся параметры, не выбранные ядром и не интерпретированные в качестве переменных окружения, будут переданы в дальнейшую обработку, которую обычно выполняет программа init. Чаще всего процессу init в качестве параметра передается слово `single`, которое сообщает init о необходимости загрузить компьютер в однопользовательском режиме. Список параметров программы init можно найти в соответствующей справочной документации.

Общие неаппаратные параметры загрузки

В этом разделе рассматриваются параметры загрузки, не связанные с каким-либо оборудованием или периферией, а с параметрами ядра, такими как управление памятью, RAM-диском, корневой системой и т. п.

Опции корневой файловой системы

Параметр *root*

Этот параметр сообщает ядру, какое устройство будет использовано в качестве корневой файловой системы во время загрузки. По умолчанию эта ус-

тановка имеет значение корневого устройства системы, на котором было скомпилировано ядро. Например, на одном компьютере корневая файловая система находится на /dev/hda2, а на другом — на /dev/hda6. Если скомпилировать ядро на втором компьютере, перенести его на первый и не указать в параметре `root=/dev/hda2`, то ядро будет думать, что оно загружается с /dev/hda6. А такого устройства на этом компьютере нет! Допустимыми корневыми устройствами могут быть следующие:

- /dev/hdaN, /dev/hdbN, /dev/hdcN, /dev/hddN, которые являются разделами N на IDE-диске;
- /dev/sdaN, /dev/sdbN, /dev/sdcN, /dev/sddN, /dev/sdeN, которые являются разделами N на SCSI-диске;
- /dev/fd0, /dev/fd1 — привод флоппи-диска с номером N;
- /dev/nfs, не являющееся флагом, заставляющим ядро получить корневую файловую систему по сети.

Это один из немногих параметров загрузки ядра, которые хранятся в его образе и могут быть изменены утилитой rdev.

Параметры *ro и rw*

Параметр `ro` сообщает ядру о необходимости монтирования корневой файловой системы в режиме "только для чтения". Парный ему параметр `rw` указывает ядру монтировать корневую файловую систему в режиме "чтение/запись". Сразу после загрузки ядра и запуска процесса init система должна осуществить проверку подмонтированных файловых систем на отсутствие ошибок. Однако, если корневая файловая система смонтирована в режиме "чтение/запись" надежно проверить целостность файловой системы невозможно. Существует два способа решения проблемы:

- Смонтировать изначально корневую файловую систему в режиме "только для чтения".
- Смонтировать изначально корневую файловую систему в режиме "чтение/запись", а перед проверкой перемонтировать ее в режим "только для чтения".

Это одни из немногих параметров загрузки ядра, которые хранят значение в образе ядра и могут быть изменены утилитой rdev.

Опции управления RAM-диском

Все следующие опции сообщают ядру, как управлять устройством RAM-диска, обычно используемым для загрузки машины.

Параметр ramdisk_start

Чтобы разрешить образу ядра находиться на флоппи-диске со сжатым образом RAM-диска, необходимо добавить команду `ramdisk_start=<смещение>`.

Параметр load_ramdisk

Этот параметр сообщает ядру, нужно загружать образ RAM-диска или нет. При `load_ramdisk=1` ядро будет загружать RAM-диск. По умолчанию значение параметра равно нулю, т. е. ядро не должно в этом случае загружать RAM-диск.

Параметр prompt_ramdisk

Этот параметр сообщает ядру о необходимости вывести пользователю приглашение вставить флоппи-диск с образом RAM-диска. В однодисковой конфигурации образ RAM-диска находится на той же дискете, с которой только что закончилась загрузка ядра, поэтому приглашение не нужно. В этом случае нужно использовать команду `prompt_ramdisk=0`. В двухдисковой конфигурации может потребоваться заменить диски, поэтому нужно указать команду `prompt_ramdisk=1`. По умолчанию значение равно единице.

Параметр ramdisk_size

Поскольку RAM-диск размещается в оперативной памяти, необходимо каким-то способом указать занимаемый им объем оперативной памяти. По умолчанию это 4096 Кбайт.

Параметр *noinitrd* (начальный RAM-диск)

В ядрах, начиная с версии 2.*x*, корневой файловой системой изначально может быть RAM-диск. Эта возможность обычно используется для загрузки модулей, необходимых для монтирования реальной корневой файловой системы (например, загрузка модулей драйвера SCSI, хранящихся в образе RAM-диска, а затем монтирование реальной файловой системы на SCSI-диске).

Параметр `noinitrd` определяет, что будет происходить с данными initrd после загрузки ядра.

Параметры загрузки для управления памятью

Следующие параметры определяют действия ядра Linux по обнаружению или управлению физической и виртуальной памятью системы.

Параметр *mem*

Используется для указания объема установленной памяти (или меньшего значения, если требуется ограничить объем памяти, доступный Linux). Старые версии BIOS не могли корректно возвращать количество оперативной памяти, если оно превышало 64 Мбайт. Поэтому приходилось вручную передавать в ядро реально установленное количество памяти.

Ядро воспримет любое значение параметра `mem=xx`, которое будет указано, однако если указать *больший* размер памяти, чем физически установлено в компьютере, то при определенном количестве процессов система попытается использовать несуществующий участок памяти и в общем возникнут проблемы, и что самое неприятное, такое может произойти и через месяц-другой после конфигурирования системы.

Количество памяти можно указывать как в шестнадцатеричном представлении, так и в десятичном. Например, если в компьютере установлено 96 Мбайт оперативной памяти, можно указать `mem=0x6000000` или `mem=96M`.

Параметр *swap*

Позволяет пользователю настраивать некоторые параметры виртуальной памяти (Virtual Memory), относящиеся к разделу подкачки. Он может иметь следующие значения:

- `MAX_PAGE_AGE`;
- `PAGE_ADVANCE`;
- `PAGE_DECLINE`;
- `PAGE_INITIAL_AGE`;
- `AGE_CLUSTER_FRACT`;
- `AGE_CLUSTER_MIN`;
- `PAGEOUT_WEIGHT`;
- `BUFFEROUT_WEIGHT`.

В каталоге /usr/src/Linux-2.4.3/Documentation/vm/ содержится полезная документация по этой теме, подставляемая с ядром операционной системы.

Параметр *buff*

Параметр, похожий на `swap`, позволяет пользователю настроить некоторые параметры, связанные с управлением буферной памятью. Он может иметь следующие значения:

- `MAX_BUFF_AGE`;
- `BUFF_ADVANCE`;

- BUFF_DECLINE;
- BUFF_INITIAL_AGE;
- BUFFEROUT_WEIGHT;
- BUFFERMEM_GRACE.

Параметры загрузки для файловой системы NFS

Linux поддерживает и бездисковые рабочие станции, загружаемые по локальной сети. Для этого необходимо настроить корневую файловую систему бездисковой станции как NFS (Network File System, сетевая файловая система). Чтобы сообщить бездисковой рабочей станции, с какой машины она будет получать операционную систему, используются указанные в этом разделе параметры. Также необходимо установить параметр `root=/dev/nfs`. Подробная информация по использованию NFS в качестве корневой файловой системы содержится в файле /usr/src/Linux-2.4.3/Documentation/nfsroot.txt. Эта тема также будет рассмотрена в *гл. 33*.

Параметр *nfsroot*

Параметр сообщает ядру, какую машину, какой каталог и с какими опциями NFS использовать в качестве корневой файловой системы. Формат этого параметра следующий:

```
nfsroot=[<server-ip>:]<root-dir>[,<nfs-options>]
```

Если параметр `nfsroot` не был дан в командной строке, то по умолчанию будет использовано значение `/tftpboot/%s`.

Другие опции:

- `<server-ip>` — задает IP-адрес сервера NFS. Если это поле не задано, по умолчанию адрес будет определен переменной `nfsaddrs`;
- `<root-dir>` — имя каталога на сервере, монтируемого как корневой. Если в строке имеется фраза `"%s"`, она будет заменена на ASCII-представление IP-адреса клиента;
- `<nfs-options>` — стандартные опции NFS. Все опции разделены запятыми. Если поле опций не задано, будут использованы следующие параметры:
  - `port` = указывается демоном portmap-сервера
  - `rsize = 1024`
  - `wsize = 1024`
  - `timeo = 7`
  - `retrans = 3`

- acregmin = 3
- acregmax = 60
- acdirmin = 30
- acdirmax = 60
- flags = hard, nointr, noposix, cto, ac

Параметр *nfsaddrs*

Параметр загрузки устанавливает параметры сетевого интерфейса. Если параметр опущен, то для выяснения этих значений ядро попытается использовать RARP и/или BOOTP. Формат параметра следующий:

```
nfsaddrs=<my-ip>:<serv-ip>:<gw-ip>:<netmask>:<name>:<dev>:<auto>
```

- <my-ip> — IP-адрес клиента. Если параметр опущен, адрес определяется с помощью RARP или BOOTP. Выбор протокола будет зависеть от того, как было сконфигурировано ядро, и от параметра <auto>. Если параметр указан, ни RARP, ни BOOTP использоваться не будут;
- <serv-ip> — IP-адрес сервера NFS. Если это поле опущено, будет использован адрес сервера, ответившего на запрос RARP или BOOTP;
- <gw-ip> — IP-адрес шлюза. Если поле опущено, шлюзы использоваться не будут;
- <netmask> — маска сети для сетевого интерфейса;
- <name> — имя клиента;
- <dev> — имя используемого сетевого устройства. Если поле опущено, для RARP-запросов будут использованы все устройства, а для BOOTP — первое найденное. Для NFS будет использовано устройство, на котором были получены ответы RARP или BOOTP;
- <auto> — автоконфигурирование. Можно использовать следующие значения:
  - rarp — использовать протокол RARP;
  - bootp — использовать протокол BOOTP;
  - both — будут применены оба протокола;
  - none — означает отсутствие автоконфигурирования. В этом случае следует указать все необходимые значения в предыдущих полях.

Дополнительные параметры загрузки

Эти параметры начальной загрузки позволяют пользователю настраивать некоторые внутренние параметры ядра.

Параметр *debug*

Ядро Linux имеет возможность выводить важные сообщения на консоль (ошибки ввода/вывода, проблемы с оборудованием и т. п.). Пороговое значение важности сообщения задается переменной `console_loglevel`. По умолчанию на консоль отправляется практически все, кроме отладочной информации. Использование параметра `debug` позволит *всем* сообщениям ядра попадать на консоль.

Параметр *init*

Во время загрузки ядро Linux запускает программу init, которая затем подготавливает операционную систему для полноценной работы. Сначала ядро Linux ищет программу init в каталоге /sbin, а при неудаче попробует запустить ее из каталога /bin/sh. Если программа init повреждена и загрузить операционную систему не удается, можно использовать командную строку загрузки `init=/bin/sh`, которая даст возможность заменить поврежденную программу или выполнить какие-то другие программы.

Параметр *kbd-reset*

Обычно на компьютерах семейства *x*86 ядро Linux не сбрасывает при загрузке контроллер клавиатуры, предполагая, что это делает BIOS. Однако такое предположение не всегда соответствует действительности. Применение этой опции заставляет во время загрузки Linux делать сброс контроллера клавиатуры.

Параметр *maxcpus*

Параметр ограничивает максимальное количество процессоров, используемое в режиме SMP. Указание в параметре `0` эквивалентно опции `nosmp`.

Параметр *mca-pentium*

Параметр, специфичный для компьютеров IBM модели 95 с шиной MCA (Microchannel), которые зависают во время теста, выполняемого Linux для обнаружения типа математического сопроцессора. Эту проблему можно решить с помощью параметра загрузки `mca-pentium`.

Параметр *md*

Если корневая система компьютера расположена на составном (Multiple) устройстве (как правило, это RAID-массив дисков), то можно использовать параметр `md`, чтобы сообщить ядру конфигурацию составного устройства.

Подробная информация по этой теме содержится в файле /usr/src/Linux-2.4.3/Documentation/md.txt.

Параметр *no387*

Параметр актуален только для старых компьютеров на базе процессора i386. В некоторых сопроцессорах i387 есть ошибки, например, ранние чипы ULSI-387 вызывают зависание при вычислениях с плавающей запятой. Параметр загрузки `no387` заставляет Linux игнорировать сопроцессор, даже если он имеется.

Параметр *no-hlt*

Параметр актуален только для старых компьютеров на базе процессора i486. У процессоров Intel есть инструкция hlt, заставляющая процессор ничего не делать, пока внешнее устройство (клавиатура, винчестер и т. п.) не вызовут его для выполнения задачи. Некоторые чипы i486 имели проблемы с командой hlt, после которой они не могли вернуться в рабочий режим. С помощью параметра `no-hlt` можно заставить ядро Linux при отсутствии активности вместо остановки процессора выполнять бесконечный цикл.

Параметр *no-scroll*

Параметр запрещает при загрузке функцию прокрутки. Актуально только для некоторых устаревших терминалов.

Параметр *noapic*

Параметр позволяет ядру Linux с поддержкой мультипроцессорности не использовать расширенные возможности контроллера прерываний в многопроцессорных машинах. Подробную информацию можно найти в файле /usr/src/Linux-2.4.3/Documentation/IO-APIC.txt.

Параметр *nosmp*

Позволяет ядру Linux с поддержкой мультипроцессорности на SMP-машинах работать только с одним процессором. Обычно используется для отладки.

Параметр *panic*

В крайне редком случае "паники" ядра (обнаруженная ядром внутренняя ошибка, которую ядро считает достаточно серьезной, что приводит к выдаче сообщения `kernel panic` и полной остановке системы) по умолчанию

компьютер остается в этом состоянии, пока администратор его не перезагрузит. Однако иногда необходимо, чтобы машина автоматически перезагрузила себя, чтобы восстановить нормальную работу системы. Используя этот параметр, можно установить время (в секундах), по прошествии которого система попытается перезагрузиться. Например, при установке параметра `panic=20` ядро Linux попытается перегрузиться через 20 секунд после выдачи сообщения `kernel panic`. Нулевое значение соответствует стандартному поведению — ждать вмешательства администратора.

Также время тайм-аута можно прочитать и изменить через интерфейс /proc/sys/kernel/panic.

Параметр *pirq*

Эта опция передает мультипроцессорному ядру информацию об установках IRQ-слота PCI для некоторых материнских плат SMP. Подробную информацию можно найти в файле /usr/src/Linux-2.4.3/Documentation/IO-APIC.txt.

Параметр *profile*

Разработчики ядер могут разрешать опции, позволяющие им с целью оптимизации быстродействия ядра определять, как и где ядро может использовать циклы процессора. Эта опция позволяет указать номер конфигурационного файла при загрузке. Можно также скомпилировать ядро с конфигурацией, разрешенной по умолчанию.

Параметр *reboot*

Параметр задает тип перезагрузки, выполняемой ядром Linux. Стандартно ядро Linux выполняет так называемую "холодную" перезагрузку (полная инициализация аппаратного обеспечения, BIOS проверяет память и т. д.). Существует также "теплая" перезагрузка, при которой не происходит первоначального тестирования оборудования, что несколько убыстряет загрузку операционной системы.

Параметр *reserve*

Используется для защиты диапазона портов ввода/вывода от тестирования (I/O probe). Формат команды:

```
reserve=iobase,extent[,iobase,extent]...
```

В некоторых машинах бывает необходимо защитить драйверы устройств от поиска устройств (auto-probing) в определенном диапазоне. Причиной могут послужить устройства, идентифицирующиеся ошибочно, или устройства, инициализация которых ядром нежелательна.

Параметр загрузки `reserve` устраняет проблему, указывая диапазон портов ввода/вывода, который необходимо исключить из тестирования. При этом диапазон резервируется в таблице ядра регистрации портов как уже определенный. Такой механизм необходимо использовать только при наличии проблем или в специальных случаях.

Параметр *vga*

Опция, интерпретируемая LILO, а не ядром, однако ее применение стало настолько обычным, что заслуживает упоминания. Также может быть установлена с помощью команды `rdev -v`. Лучший способ применения этой опции — стартовать с `vga=ask`. Тогда до загрузки ядра будет предложен список различных режимов, допустимых для имеющейся в системе видеокарты. Более подробная информация содержится в файле /usr/src/Linux-2.4.3/Documentation/svga.txt.

Загрузочные параметры, определяющие поведение шины PCI

Параметр `pci=` можно использовать для изменения способа поиска устройств на шине PCI и поведения этих устройств. Как правило, это необходимо либо для старого оборудования, не совсем корректно использующего технологию Plug and Play, либо для специфических PCI-устройств.

Аргументы *pci=bios* и *pci=nobios*

Используются для установки или сброса флага индикации тестирования (probing) PCI через PCI BIOS. По умолчанию используется BIOS.

Аргументы *pci=conf1* и *pci=conf2*

Разрешают тип конфигурации 1 или 2. Также они неявно сбрасывают флаг PCI BIOS probe (т. е. `pci=nobios`).

Аргумент *pci=io=*

Если получено сообщение типа

```
Unassigned IO space for.../
```

то может потребоваться указать значение ввода/вывода этой опцией.

Аргумент *pci=nopeer*

Специфический аргумент, исправляющий погрешности некоторых версий BIOS.

Аргумент *pci=nosort*

Использование этого аргумента заставляет ядро не сортировать PCI-устройства в процессе проверки.

Аргумент *pci=off*

Использование этой опции запрещает все проверки PCI-шины. Любые драйверы устройств, использующих функции PCI для поиска и инициализации оборудования, скорее всего, потеряют работоспособность.

Аргумент *pci=reverse*

Эта опция меняет на обратный порядок PCI-устройств на шине PCI.

Аргументы загрузки для драйверов буфера видеофреймов

Аргумент `video=` используется, когда уровень абстракции устройства буфера фреймов встроен в ядро. Это означает, что вместо наличия отдельных программ для каждого семейства видеокарт (VOODOO, TNT, S3 и пр.) ядро имеет встроенный драйвер для каждой видеокарты и экспортирует единственный (единый) интерфейс для видеопрограмм. Типичный формат этого аргумента:

```
video=name:option1,option2,...
```

где `name` — название универсальной опции или драйвера буфера фреймов. Как только найдено совпадающее имя драйвера, то список параметров, разделенных запятыми, передается в этот конкретный драйвер для окончательной обработки.

Информацию по опциям, поддерживаемым каждым драйвером, можно найти в файле /usr/src/Linux-2.4.3/Documentation/fb/.

Аргумент *video=map:...*

Эта опция используется для установки консоли отображения устройства буфера фреймов.

Аргумент *video=scrollback:...*

Число после двоеточия устанавливает размер памяти, выделенной для буфера прокрутки. Суффикс `k` или `K` после числа указывает, что число представляет килобайты.

Аргумент *video=vc:...*

Число или диапазон чисел определяют первую или первую и последнюю виртуальную консоли буфера фреймов.

Аргументы загрузки для SCSI-периферии

Этот раздел содержит описание аргументов загрузки, используемых для передачи информации об установленных SCSI-контроллерах и устройствах.

Аргументы для драйверов Mid-level

Драйверы уровня Mid управляют такими устройствами, как винчестеры, CD-ROM и стримеры без учета специфики SCSI-контроллера.

Максимальный LUN (*max_scsi_luns=*)

Каждое SCSI-устройство может иметь несколько псевдоустройств внутри себя. К примеру, SCSI CD-ROM, обслуживающий более чем один диск одновременно. Каждый CD-ROM адресуется номером логического устройства (Logical Unit Number, LUN). Но большинство SCSI-устройств являются одним устройством, и им назначается нулевой LUN.

Старые SCSI-устройства не могут обработать запросы поиска с LUN, не равным нулю. Зачастую это приводит к зависанию устройства. Чтобы избежать указанной проблемы, по умолчанию пробуется только нулевой LUN.

Для определения количества пробуемых LUN при загрузке, в качестве аргумента загрузки вводится `max_scsi_luns=`*n*, где *n* — номер от 1 до 8.

Регистрация SCSI (*scsi_logging=*)

Ненулевое значение этого загрузочного аргумента включает регистрацию всех SCSI-событий.

Параметры для ленточного накопителя SCSI (*st=*)

При загрузке ядра Linux можно изменить конфигурацию ленточного накопителя SCSI, используя

```
st=buf_size[,write_threshold[,max_bufs]]
```

Первые два числа указываются в килобайтах. По умолчанию `buf_size` равен 32 Кбайт. `write_threshold` — значение, при котором буфер сбрасывается на ленту, по умолчанию 30 Кбайт. Максимальное количество буферов зависит от количества обнаруженных ленточных накопителей, по умолчанию равно 2.

Аргументы для контроллеров SCSI

Понятия, используемые в данном разделе:

- `iobase` — первый порт ввода/вывода, занимаемый контроллером SCSI. Указывается в шестнадцатеричной нотации и обычно лежит в диапазоне от `0x200` до `0x3ff`;
- `irq` — аппаратное прерывание, установленное на карте. Допустимые значения зависят от конкретного контроллера, но обычно это 5, 7, 9, 10, 11, 12 и 15;
- `dma` — используемый картой канал DMA (Direct Memory Access — прямой доступ к памяти). Обычно применяется только для карт с управлением шиной (bus-mastering);
- `scsi-id` — идентификатор, используемый контроллером для идентификации себя на SCSI-шине. Только некоторые контроллеры позволят изменить это значение. Типичное значение по умолчанию — 7.
- `parity` — ожидает ли SCSI-контроллер поддержку всеми подсоединенными устройствами четности при всех информационных обменах. Единица разрешает проверку четности, ноль — запрещает.

К сожалению, большей неразберихи, чем в настройках SCSI-контроллеров и устройств, наверное, не существует. До недавнего времени любая попытка улучшить поддержку SCSI-устройств в Linux оборачивалась тем, что какие-то новые контроллеры работали, а старые (казалось, уже давно отлаженные) теряли свою работоспособность.

В качестве примера ниже приведена конфигурация некоторых семейств контроллеров. Подробную информацию следует искать в документации на конкретные контроллеры.

Adaptec *aha154x* (*aha1542=*)

Карты серии aha154x с управлением шиной. Аргументы загрузки выглядят следующим образом:

```
aha1542=iobase[,buson,busoff[,dmaspeed]]
```

Допустимые значения `iobase`: `0x130`, `0x134`, `0x230`, `0x234`, `0x330`, `0x334`. Клоны карты могут допускать другие значения.

Значения `buson`, `busoff` указывают количество микросекунд, на которое карта может захватить ISA-шину.

Параметр `dmaspeed` указывает скорость в мегабайтах в секунду, с которой происходит DMA-доступ. По умолчанию — 5 Мбайт/с.

Adaptec *aha274x*, *aha284x*, *aic7xxx* (*aic7xxx=*)

Эти контроллеры принимают следующие аргументы:

```
aic7xxx=extended,no_reset
```

Здесь:

- `extended` — значение, используемое с винчестерами большой емкости;
- `no_reset` — значение, запрещающее сброс SCSI-шины во время загрузки.

Если SCSI-контроллер не желает нормально функционировать, следует обратится к SCSI-HOWTO или к документации ядра. Возможно, там присутствует данный SCSI-контроллер и описано решение этой проблемы.

Жесткие диски

В этом разделе приводится список аргументов загрузки для стандартных жестких дисков (винчестеров) MFM/RLL, ST-506, XT и устройств IDE.

Параметры драйвера IDE — винчестера/CD-ROM

Драйвер IDE допускает множество параметров, от определения геометрии диска до поддержки расширенных или дефектных микросхем контроллера. Подробная информация по конфигурации драйвера содержится в файле /usr/src/Linux-2.4.3/Documentation/ide.txt.

- `hdx=` — распознается от `a` до `h`, например HDD;
- `idex=` — распознается от 0 до 3, например IDE1;
- `hdx=noprobe` — привод может присутствовать, но он не тестируется;
- `hdx=none` — жесткий диск отсутствует, CMOS игнорируется и тестирование не производится;
- `hdx=nowerr` — игнорируется бит `WRERR_STAT` на этом приводе;
- `hdx=cdrom` — привод присутствует и является приводом CD-ROM;
- `hdx=cyl,head,sect` — принудительное указание геометрии жесткого диска;
- `hdx=autotune` — привод попытается настроить скорость интерфейса на самый быстрый поддерживаемый режим PIO, который только возможен для этого привода. На старых материнских платах не гарантируется полная поддержка такого режима;
- `idex=noprobe` — не тестировать данный интерфейс;

- idex=base — задать адрес указанному интерфейсу, где base обычно 0x1f0 или 0x170, а ctl подразумевается base+0x206;
- idex=base,ctl — указывает как base, так и ctl;
- idex=base,ctl,irq — указывает base, ctl и номер IRQ;
- idex=autotune — будет произведена попытка настроить скорость интерфейса на самый быстрый поддерживаемый режим PIO для всех приводов на этом интерфейсе. На старых материнских платах не гарантируется полная поддержка такого режима;
- idex=noautotune — привод не будет пытаться настроить скорость интерфейса;
- idex=serialize — не выполнять операции overlap на idex.

Нижеследующее допустимо только на IDE0, и умолчания для base- и ctl-портов не должны меняться. Используется для старых чипсетов времен процессоров i386 и i486:

- ide0=dtc2278 — поддерживать контроллер DTC2278;
- ide0=ht6560b — поддерживать контроллер HT6560B;
- ide0=cmd640_vlb — необходим для карт VLB с чипом CMD640;
- ide0=qd6580 — поддерживать контроллер qd6580;
- ide0=ali14xx — поддерживать чипсеты ALI14xx (ALI M1439/M1445);
- ide0=umc8672 — поддерживать чипсет UMC8672.

Опции драйвера диска стандарта ST-506 (*hd*)

Устаревший стандарт. Сегодня вряд ли можно где-нибудь столкнуться с жестким диском этого стандарта. Допустим только аргумент hd=.

Формат: hd=cyls,heads,sects

Если установлено два диска, точно так же задается второй диск.

Опции драйвера диска XT (*xd*)

Устаревший стандарт. Аргумент загрузки для жесткого диска:

xd=type,irq,iobase,dma_chan

Значение type указывает конкретного производителя карты и обозначается: 0=generic; 1=DTC; 2,3,4=Western Digital, 5,6,7=Seagate; 8=OMTI. Единственное отличие между разными типами от одного и того же производителя — строка BIOS, используемая для обнаружения, которая не активизируется, если указан тип.

CD-ROM (не-SCSI/ATAPI/IDE)

Первоначально, когда только появились приводы CD-ROM, для этих приводов не было единого интерфейса. Последние 4—5 лет выпускаются приводы CD-ROM только с SCSI или IDE-интерфейсом. Однако иногда приходится встречаться с CD-ROM с интерфейсом Sony и Mitsumi. Ниже приведены параметры для различных контроллеров приводов CD-ROM.

Более подробная информация содержится в каталоге /usr/src/Linux-2.4.3/Documentation/cdrom.

Интерфейс Aztech (*aztcd*)

Синтаксис для этого типа интерфейса:

```
aztcd=iobase[,magic_number]
```

Если установить `magic_number` равным `0x79`, драйвер опробует устройство и в случае неизвестной ему версии оборудования (firmware) отключится. Все другие значения игнорируются.

Интерфейс Sony CDU-31A и CDU-33A (*cdu31a*)

Синтаксис:

```
cdu31a=iobase,[irq[,is_pas_card]]
```

Указав значение IRQ, равное нулю, сообщаем драйверу, что аппаратные прерывания не поддерживаются. Если имеющаяся карта поддерживает прерывания, следует использовать их для уменьшения загрузки центрального процессора.

Интерфейс Sony CDU-535 (*sonycd535*)

Синтаксис для этого интерфейса:

```
sonycd535=iobase[,irq]
```

Если необходимо прописать значение IRQ, то в качестве адреса ввода/вывода следует указать 0.

Интерфейс GoldStar (*gscd*)

Синтаксис для интерфейса CD-ROM:

```
gscd=iobase
```

Интерфейс ISP16 (*isp16*)

Синтаксис для этого интерфейса:

```
isp16=[port[,irq[,dma]]][[,]drive_type]
```

Использование нуля для IRQ или DMA означает, что они не используются. Допустимые значения для `drive_type` — `noisp16`, `Sanyo`, `Panasonic`, `Sony` и `Mitsumi`. Применение `noisp16` полностью запрещает драйвер.

Интерфейс Mitsumi Standard (*mcd*)

Синтаксис для этого интерфейса CD-ROM:

```
mcd=iobase,[irq[,wait_value]]
```

`wait_value` используется как значение внутреннего тайм-аута.

Интерфейс Optics Storage (*optcd*)

Синтаксис для этого типа карт:

```
optcd=iobase
```

Интерфейс Phillips CM206 (*cm206*)

Синтаксис для этого типа карт:

```
cm206=[iobase][,irq]
```

Драйвер предполагает значения IRQ между 3 и 11, а значения портов ввода/вывода — между `0x300` и `0x370`. Также допускается `cm206=auto` для разрешения автоматического определения параметров.

Интерфейс Sanyo (*sjcd*)

Синтаксис для этого типа карт:

```
sjcd=iobase[,irq[,dma_channel]]
```

Интерфейс SoundBlaster Pro (*sbpcd*)

Синтаксис для этого типа карт:

```
sbpcd=iobase,type
```

где `type` — один из следующих (чувствителен к регистру) значений: `SoundBlaster`, `LaserMate`, или `SPEA`. `iobase` — адрес интерфейса CD-ROM, а не звуковой части карты.

Последовательные и ISDN-драйверы

В разделе приведены параметры для некоторых ISDN-карт и так называемых мультипортовых последовательных контроллеров. Как обычно, первоначально единых стандартов не существовало, и из-за этого приходится иногда использовать параметры, передаваемые при загрузке ядра.

Драйвер PCBIT ISDN (*pcbit*)

Параметры:

```
pcbit=membase1,irq1[,membase2,irq2]
```

где `membaseN` — база разделяемой памяти для `N`-ой карты, а `irqN` — установленное прерывание для `N`-ой карты. По умолчанию IRQ 5 и membase `0xD0000`.

Драйвер Teles ISDN (*teles*)

ISDN-драйвер требует аргументы загрузки в следующем виде:

```
teles=iobase,irq,membase,protocol,teles_id
```

где `iobase` — адрес порта ввода/вывода карты, `membase` — базовый адрес разделяемой памяти карты, `irq` — прерывание, используемое картой, `teles_id` — уникальная строка идентификатора.

Драйвер DigiBoard (*digi*)

Драйвер мультипортового последовательного контроллера DigiBoard принимает строку из шести идентификаторов или целых чисел, разделенных запятыми. Значения по порядку:

- `Enable/Disable` — разрешить/запретить использование контроллера;
- тип карты — `PC/Xi(0), PC/Xe(1), PC/Xeve(2), PC/Xem(3)`;
- `Enable/Disable` — разрешить/запретить альтернативное расположение контактов;
- количество портов на этой карте;
- порт ввода/вывода, на который сконфигурирована карта;
- база окна памяти.

Пример аргумента загрузки:

```
digi=E,PC/Xi,D,16,200,D0000
```

Более подробную информацию можно прочитать в файле /usr/src/Linux-2.4.3/Documentation/digiboard.txt.

Последовательный/параллельный радиомодем Baycom (*baycom*)

Формат аргумента загрузки для этого устройства:

```
baycom=modem,io,irq,options[,modem,io,irq,options]
```

Использование `modem=1` означает, что у вас устройство `ser12`; `modem=2` — устройство `par96`. Значение `options=0` предписывает использование аппаратного DCD, а `opton=1` — программного DCD. Параметры `io` и `irq` — базовый порт ввода/вывода и прерывание.

Драйверы других устройств

В разделе приведены параметры загрузки других устройств, не вошедших ни в одну из упомянутых выше категорий.

Устройства Ethernet (*ether*)

Драйверы для различных видов сетевых контроллеров поддерживают разные параметры, но они все используют значения прерывания, базовый адрес порта ввода/вывода и имя. В наиболее универсальной форме это выглядит так:

```
ether=irq,iobase[,param_1[,param_2,…]]],name
```

Первый нецифровой аргумент воспринимается как имя. Обычно значения `param_n` имеют различные назначения для разных сетевых контроллеров. Чаще всего этот параметр используют для второй сетевой карты, поскольку по умолчанию автоматически определяется только одна сетевая карта. Это можно сделать, указав:

```
ether=0,0,eth1
```

Обратите внимание, что нулевые значения IRQ и базы ввода/вывода в примере заставляют драйвер сделать автоопределение параметров сетевой карты.

Данный пример не будет автоматически определять параметры второй сетевой карты, если вместо вкомпилированных в ядро использовать загружаемые модули. Большинство современных дистрибутивов Linux используют ядро операционной системы в комбинации с загружаемыми модулями. Параметр `ether=` применяется только для драйверов, вкомпилированных непосредственно в ядро.

Полная информация по конфигурации и использованию нескольких сетевых карт и описание особенностей настройки конкретных типов сетевых карт содержится в ETHERNET-HOWTO.

Драйвер флоппи-диска (*floppy*)

Существует большое количество опций драйвера флоппи-диска, и все они перечислены в файле /usr/src/Linux-2.4.3/drivers/block/README.fd. Использование параметров загрузки для дисковода зачастую вызывает откровенное непонимание — казалось бы, более стандартное устройство трудно найти. Однако достаточно много проблем вносят ноутбуки. Почти треть параметров загрузки для дисковода так или иначе касаются только ноутбуков. Ниже приведены только основные опции:

- `floppy=0,daring` — сообщает драйверу дисковода о необходимости запрета всех рискованных операций;
- `floppy=thinkpad` — сообщает драйверу дисковода, что у вас ноутбук фирмы IBM;
- `floppy=nodma` — указывает драйверу дисковода не использовать DMA для передачи данных. Необходима при установке Linux на ноутбук HP Omnibooks, у которого нет работающего DMA-канала для дисковода. Эта опция также необходима, если вы часто получаете сообщения Unable to allocate DMA memory ("Не могу распределить память DMA");
- `floppy=nofif` — полностью запрещает буфер FIFO (First Input First Output, первый вошел — первый вышел) при операциях записи/чтения. Применение этого параметра необходимо, если при доступе к дисководу вы получаете сообщения Bus master arbitration error ("Ошибка разделения шины");
- `floppy=broken_dcl` — указывает драйверу не использовать сигнал смены диска (Disc Change Line, DCL), однако при этом каждый раз при повторном открытии узла устройства (device node) операционная система предполагает, что диск был заменен. Необходима для компьютеров, где сигнал замены диска поврежден или не поддерживается. В основном касается ноутбуков. Однако, если вдруг возникли проблемы с определением замены дискеты — это первый признак того, что дисковод скоро выйдет из строя;
- `floppy=debug` — установка этой опции указывает драйверу выводить отладочную информацию;
- `floppy=message` — указывает драйверу выводить информационные сообщения для некоторых дисковых операций.

Драйвер звуковой карты (*sound*)

Драйвер звуковой карты также может принимать аргументы загрузки для изменения вкомпилированных в ядро значений. Делать этого не рекомендуется, поскольку в связи с отсутствием внятной документации такие действия сильно смахивают на шаманство. Намного надежнее использовать загружаемые модули.

Тем более что за последнее время заметно улучшилось качество драйверов для звуковых карт и заметно увеличился ассортимент поддерживаемых драйверами устройств. Принимается аргумент загрузки в следующем виде:

```
sound=device1[,device2[,device3...]]
```

где каждое значение `deviceN` имеет формат `0xDTaaaId`. Расшифруем формат `deviceN`:

- `D` — второй канал DMA (ноль не применяется);
- `T` — тип устройства (список звуковых карт до типа 26 находится в файле /usr/src/Linux-2.4.3/include/linux/soundcard.h, а от 27 до 999 — в файле /usr/src/Linux-2.4.3/drivers/sound/dev_table.h.):
  - `1=FM`
  - `2=SB`
  - `3=PAS`
  - `4=GUS`
  - `5=MPU401`
  - `6=SB16`
  - `7=SB16-MIDI`

  и т. д.;
- `aaa` — адрес ввода/вывода в шестнадцатеричном представлении;
- `I` — номер прерывания в шестнадцатеричном представлении;
- `d` — первый канал DMA.

Применение параметра загрузки `sound=0` полностью запрещает драйвер звуковой карты.

Драйвер Bus Mouse (*bmouse*)

Этот драйвер поддерживает только один параметр, который является значением используемого аппаратного прерывания.

Драйвер MS Bus Mouse (*msmouse*)

Этот драйвер поддерживает только один параметр, который является значением используемого аппаратного прерывания.

Драйвер принтера (*lp*)

Используя этот аргумент загрузки можно сообщить драйверу принтера, какие порты можно использовать, а какие — нет. Этот параметр удобен для запрета захвата драйвером принтера всех доступных параллельных портов.

Формат аргумента — несколько пар адресов ввода/вывода, прерываний. Например,

```
lp=0x3bc,0,0x378,7
```

Драйвер принтера будет использовать порт на `0x3bc` без прерывания и порт `0x378` с седьмым прерыванием. Порт `0x278` (если он присутствует в компьютере) не будет использоваться, поскольку автоопределение выполняется при отсутствии аргумента `lp=`. Для полного отключения драйвера принтера можно использовать параметр `lp=0`.

Процесс init

После того как ядро Linux полностью загрузилось, считало конфигурационные параметры и настроило оборудование (по крайней мере то, которое упоминалось в конфигурационных параметрах, и то, драйверы которого присутствуют в ядре), оно приступает к монтированию разделов жесткого диска. Монтирование всегда начинается с корневой файловой системы. Как только корневая файловая система окажется загружена и смонтирована, будет выведено сообщение:

```
VFS: Mounted root (ext2 filesystem) readonly
```

В этой точке система находит на корневой файловой системе программу init и выполняет ее.

Процесс init — это программа, ответственная за продолжение процедуры загрузки и перевод операционной системы из начального состояния, возникающего после загрузки ядра, в стандартное состояние. Во время этого процесса init выполняет множество операций, необходимых для нормального функционирования операционной системы: монтирование и проверку файловых систем, запуск различных служб и т. п. Список производимых действий помимо конфигурации системы зависит от так называемого уровня выполнения (run level).

Достаточно простой аналогией уровня выполнения является обычный распорядок дня человека — пробуждение, приведение себя в порядок, завтрак, "выход в свет" — общение с окружающим миром, ужин, приведение себя в порядок, сон. Так изо дня в день, одни и те же операции, в одной и той же последовательности. Не умывшись, вы на работу не пойдете, завтрак обязательно идет перед ужином и т. д.

Точно так же разбиты уровни выполнения. Каждый уровень выполнения однозначно (по крайней мере, в пределах дистрибутива) определяет перечень действий, выполняемых процессом init, и конфигурацию запущенных процессов. К сожалению (а может, и к счастью) четкого разделения на уровни выполнения, их количество, действия, выполняемые на каждом уровне, нет. Так, в некоторых UNIX-системах уровней выполнения всего

два. Некоторые дистрибутивы Linux таким же образом конфигурируют свою операционную систему (в дистрибутиве Slackware, например, два уровня выполнения). В других дистрибутивах (Red Hat Linux) уровней выполнения восемь. Поскольку эта книга базируется на дистрибутиве Red Hat, дальнейшее описание на нем и основано.

В операционной системе Linux существует восемь уровней выполнения:

- 0 — останов системы;
- 1 — однопользовательский режим для специальных случаев администрирования. Отсутствует поддержка сети, практически нет сервисов;
- 2 — многопользовательский режим без поддержки сети;
- 3 — многопользовательский режим с поддержкой сети;
- 4 — использование не регламентировано;
- 5 — обычно по умолчанию стартует X Window System;
- 6 — перезагрузка системы;
- S или s — практически то же, что и однопользовательский режим, но уровень выполнения S используется, в основном, в скриптах.

Как можно заметить, существует определенное логическое нарушение в следовании уровней выполнения. Было бы более логично нулевой уровень выполнения вставить перед шестым. Однако здесь проявили себя исторические традиции — как повелось много лет назад в UNIX, так ради совместимости и остается.

К сожалению, не существует единого мнения, как использовать уровни со второго по пятый. По большей части это определяется идеологами дистрибутива или пристрастиями системного администратора. Приведенная выше схема уровней выполнения достаточно оптимальна, и, в конечном итоге, только вы сами решаете, как использовать уровни выполнения.

Конфигурационный файл init — /etc/inittab

Как всякая программа, после старта init сразу считывает свой конфигурационный файл /etc/inittab. Это обычный текстовый файл, состоящий из отдельных строк. Если строка начинается со знака `#` (стандартный признак комментария в конфигурационных файлах и скриптах) или пуста, она игнорируется. Все остальные строки состоят из 4 разделенных двоеточиями полей, имеющих вид:

```
id:runlevels:action:process
```

где:

- `id` — идентификатор строки. Выбирается произвольно, но в файле не может быть двух строк с одинаковыми идентификаторами. Если конфи-

гурационный файл модифицируется достаточно часто, имеет смысл использовать неписаное правило нумерации строк в BASIC — номера строкам назначать кратно пяти или десяти;

- `runlevels` — уровни выполнения, на которых эта строка будет задействована. Уровни задаются цифрами (без разделителей);
- `process` — команда, которая должна быть запущена;
- `action` — действие. В этом поле стоит ключевое слово, которое определяет, что должен делать процесс init, пока выполняется (или после выполнения) команда, заданная полем `process`:
  - `wait` — ожидать завершения процесса. Соответственно, пока не закончится данный процесс, init не запускает никаких других процессов. Как правило, такого типа процессы используются для разнообразных проверочных действий (проверка и восстановление файловых систем), а так же для запуска различных служб (демонов);
  - `once` — выполнять процесс только один раз;
  - `respawn` — перезапустить процесс в случае его "смерти". Актуально для некоторых служб, которые должны постоянно присутствовать в системе;
  - `off` — игнорировать данный элемент. Можно использовать при отладке конфигурационного файла;
  - `boot` — процесс должен быть выполнен при загрузке операционной системы, поле `runlevels` (уровни выполнения) при этом игнорируется;
  - `bootwait` — то же, что и предыдущая опция, но init должен ожидать окончания работы процесса;
  - `initdefault` — указывает init, в какой уровень выполнения необходимо перейти системе после загрузки;
  - `sysinit` — процесс должен быть выполнен во время загрузки операционной системы до выполнения любой строки с `boot` или `bootwait`;
  - `powerwait` — позволяет процессу init остановить систему при пропадании электроэнергии. Применение этого ключевого слова предполагает, что используется источник бесперебойного питания (UPS), имеющий специальный интерфейс, с помощью которого источник бесперебойного питания может посылать в компьютер и принимать из него различные управляющие сигналы (например "нет питания", "выключить источник бесперебойного питания", "аккумуляторы разряжены" и т. п.), а также программное обеспечение, которое отслеживает состояние источника бесперебойного питания и информирует init о том, что питание отключилось;
  - `ctrlaltdel` — разрешает init перезагрузить систему, когда пользователь нажимает комбинацию <Ctrl>+<Alt>+<Del> на клавиатуре. Однако

системный администратор может определить действия по <Ctrl>+ +<Alt>+<Del>, например, игнорировать нажатие этой комбинации.

Этот список не является исчерпывающим. Подробную информацию о файле inittab можно узнать из man-страниц init, inittab.

В качестве примера приведем файл inittab, который находится в только что установленной системе Red Hat 7.1.

```
# inittab    Этот файл описывает как процесс INIT должен настроить
#  операционную систему в соответствующем уровне выполнения
#
# Author:  Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#       Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#   0 — halt (Do NOT set initdefault to this)
#   1 — Single user mode
#   2 — Multiuser, without NFS (The same as 3, if you do not have
#       networking)
#   3 — Full multiuser mode
#   4 — unused
#   5 — X11
#   6 — reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud::once:/sbin/update
```

```
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few
# minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting
Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Can-
celled"



# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
# xdm is now a separate service
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Сразу после запуска процесс init считывает свой конфигурационный файл /etc/inittab и производит его разбор. Сначала он определяет, какой уровень по умолчанию установлен в системе. Как видно из приведенного конфигурационного файла id:3:initdefault уровень выполнения, в котором будет функционировать операционная система после загрузки, равен трем (то есть предполагается многопользовательский режим с поддержкой сетевых функций). Дистрибутив Red Hat по умолчанию предлагает установить вход в систему в графическом режиме — пятый уровень выполнения.

Затем процесс init принимает к сведению строки, содержащие специальные команды, такие как:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting
Down"
```

```
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Can-
celled"
```

После этого процесс init инициирует команду, которую необходимо запустить при старте системы, но перед тем как перейти к какому-нибудь уровню выполнения. Эта команда содержится в строке с ключевым словом `sysinit`.

```
si::sysinit:/etc/rc.d/rc.sysinit
```

После этого процесс init запускает скрипты, которые должны действовать в любом уровне выполнения:

```
ud::once:/sbin/update
```

а затем команды, соответствующие уровню, заданному по умолчанию:

```
l3:3:wait:/etc/rc.d/rc 3
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Как можно заметить, есть несколько строк, запускающих скрипт rc, которые отличаются только уровнем выполнения и аргументом командной строки, передаваемой в скрипт rc. Функции, выполняемые скриптами rc.sysinit и rc, будут рассмотрены *в разд. "Основные конфигурационные файлы"*.

После запуска скрипта rc процесс init выполняет запуск шести виртуальных консолей (процессов mingetty или, в более старом варианте, — gettty), что дает пользователям возможность регистрироваться в системе с терминалов (или виртуальных консолей, поскольку терминал вы вряд ли где-нибудь встретите). Для переключения между виртуальными консолями необходимо нажимать комбинацию одной из русифицированных клавиш верхнего ряда клавиатуры: клавиши <Alt> с номером, соответствующим номеру виртуальной консоли. После инициализации виртуальных консолей можно считать, что система полностью перешла в соответствующий уровень выполнения, загрузка завершилась, операционная система ожидает регистрации пользователя.

После окончания загрузки init продолжает функционировать в фоновом режиме. Поэтому, с помощью команды `telinit`, которая взаимодействует с процессом init, можно произвести перевод системы с одного уровня выполнения на другой или указать init перечитать свой конфигурационный файл.

Когда пользователь останавливает систему (командой `shutdown`, `halt`, `poweroff` или `reboot`), процесс init завершает все исполняющиеся процессы, размонтирует все файловые системы и останавливает процессор или производит перезагрузку системы.

Основные конфигурационные файлы

Таким образом, в итоге рассмотрения предыдущего раздела мы установили, что процесс init выполняет три основных действия:

- запускает скрипт rc.sysinit из каталога /etc/rc.d;
- запускает скрипт rc из того же каталога /etc/rc.d с опцией, равной уровню выполнения (обычно третий или пятый уровни выполнения);
- запускает процессы getty.

Как следует из материала *гл. 5*, в каталоге /etc находится каталог rc.d, содержимое которого непосредственно касается процесса загрузки системы. Вот оно:

/init.d	/rc2.d	/rc5.d	rc.local
/rc0.d	/rc3.d	/rc6.d	rc.sysinit
/rc1.d	/rc4.d	rc	

Опираясь на предыдущую информацию, нетрудно заметить, что существует семь каталогов для каждого уровня выполнения, какой-то каталог /init.d и три исполняемых файла, два из которых нам уже знакомы — rc и rc.sysinit. Третий файл — rc.local — вызывается по окончании исполнения файла rc и предназначен для команд, добавляемых администратором для запуска в процессе начальной загрузки. Редактировать файл rc не возбраняется, однако вероятность ошибки в файле, содержащем сотню-другую строк, очень велика, поэтому настоятельно рекомендуется использовать только файл rc.local.

rc.sysinit

Вернемся к процессу загрузки. Файл rc.sysinit предназначен для выполнения начальных действий, необходимых для корректного функционирования операционной системы. Ниже приведен список действий, выполняемых скриптом rc.sysinit. Конечно, он зависит от дистрибутива и от конфигурации системы, но в большей части он неизменен.

Действия скрипта:

- установка путей;
- установка имени хоста;
- чтение конфигурационных данных из /etc/sysconfig/network;
- вывод баннера;
- монтирование файловой системы /proc;
- конфигурирование параметров ядра системы, используя файл /etc/sysctl.conf;
- установка системных часов, используя конфигурацию из /etc/sysconfig/clock;

- установка параметров клавиатуры консоли программой loadkeys в соответствии с файлами /etc/sysconfig/console/default.kmap или /etc/sysconfig/keyboard;
- загрузка системного шрифта из /etc/sysconfig/i18n и файлов с расширением pcf.gz или gz из каталогов /etc/sysconfig/console, /usr/lib/kbd/consolefonts или /lib/kbd/consolefonts;
- активация области подкачки;
- инициализация USB-контроллера;
- запуск программы fsck для корневой системы, при обнаружении серьезных проблем выполняется немедленная перезагрузка;
- старт PNP-устройств в соответствии с /etc/isapnp.conf;
- перемонтирование корневой файловой системы в режим чтения/записи;
- перенастройка таблицы монтирования /etc/mtab;
- проверка квот для корневой файловой системы;
- проверка необходимости загрузки модулей, нахождение зависимостей, загрузка и конфигурирование модулей;
- подключение RAID-устройств;
- запуск fsck для других систем;
- монтирование локальных файловых систем;
- включение механизма квот;
- удаление тригерных файлов загрузки;
- очистка каталогов /var/lock и /var/run;
- очистка файлов /var/run/utmp и /var/run/utmpx;
- удаление файлов-защелок из /tmp;
- включение подкачки;
- инициализация последовательных устройств, используя скрипт /etc/rc.d/rc.serial;
- загрузка модулей для SCSI-стримера;
- генерация файла заголовка для определения загружаемого ядра командой /sbin/mkkerneldoth;
- установка ссылки /boot/System.map;
- проверка использования интерактивного режима загрузки и, в случае необходимости, создание файла /var/run/confirm.

Запуск проверки файловой системы командой `fsck` может быть принудительно отключен при наличии файла /fastboot, а также включен при наличии /forcefsck. Создать эти файлы можно выполнением команды `shutdown` с

соответствующими ключами. Однако не рекомендуется злоупотреблять этими возможностями.

Sysctl позволяет зафиксировать ряд параметров и обеспечить (через /etc/sysctl.conf) их установку после перезагрузки. Вот как выглядит /etc/sysctl.conf сразу после инсталяции системы:

```
# Disables packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
net.ipv4.conf.all.rp_filter = 1
# Disables the magic-sysrq key
kernel.sysrq = 0
```

rc

Прежде чем приступить к разбору скрипта rc, необходимо упомянуть о каталогах /rcX.d и /init.d. Уточним еще раз — иерархия /rcX.d характерна для дистрибутивов Red Hat и базирующихся на нем, в других дистрибутивах и в UNIX-системах их может и не быть. Эти каталоги играют исключительную роль в процессе загрузки, поскольку они содержит основные скрипты, необходимые для организации процесса загрузки.

Подкаталог /init.d содержит по одному скрипту для каждой из служб, установленных в системе (sendmail, HTTP, Samba, FTP и т. п.). Этот скрипт отвечает за запуск, остановку или перезагрузку соответствующей службы. В каталоге /rcX.d находятся ссылки на файлы скриптов, как правило расположенные в каталоге /etc/rc.d/init.d. Названия этих ссылок имеют имена, начинающиеся либо с буквы K, либо с буквы S, после которой идет двухзначное число и имя соответствующей службы. Буквы S и K — первые буквы слов start и kill соответственно. Из этого следует, что файл, начинающийся с буквы S, отвечает за старт соответствующего процесса, а файл, начинающийся с буквы K, отвечает за остановку соответствующего процесса. Цифры, идущие после S или K в именах ссылок, задают порядок запуска скриптов.

Заглянем в файл rc. Первым делом скрипт пытается определить текущий уровень выполнения и уровень выполнения, в который необходимо перевести систему. После этого он проверяет, нажимал ли пользователь букву I для перехода в режим пошаговой загрузки процессов. Затем скрипт останавливает запущенные на предыдущем уровне выполнения процессы, отсутствующие на новом уровне выполнения, а потом запускает необходимые службы для нового уровня выполнения. Как правило, одна и та же служба нужна на нескольких уровнях. Поэтому не имеет смысла эту службу при переходе с одного уровня выполнения на другой останавливать и тут же запускать. В Linux для этой цели используются специальные флаги.

Рис. 6.1. Программа ntsysv

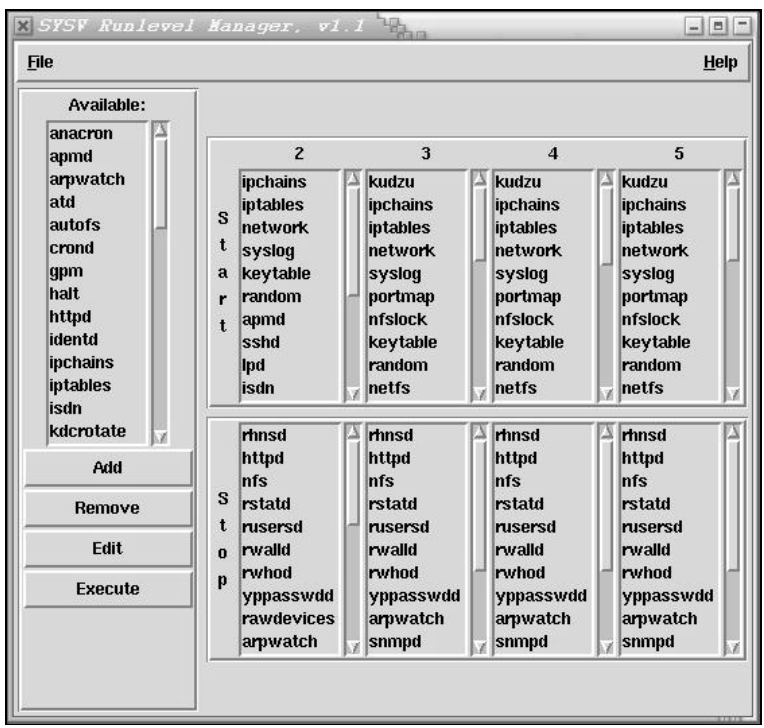

Рис. 6.2. Программа Control-panel

В качестве флагов служат файлы в каталоге /var/lock/subsys/${subsys} или /var/lock/subsys/${subsys}.init, где subsys — имя соответствующей службы. Если файлов нет, то данный процесс считается незапущенным (запуск S-файла имеет смысл), а если есть — запущенным (запуск K-файла имеет смысл). Так же для программы linuxconf создается специальный флаг /var/run/runlevel.dir, из которого можно узнать текущий уровень выполнения системы.

Для управления набором доступных служб в текущем уровне выполнения можно использовать программу конфигурирования linuxconf, программу ntsysv (рис. 6.1), /usr/sbin/setup или графическую программу Control-panel (рис. 6.2).

Можно сконфигурировать набор доступных сервисов и вручную. Для запрета старта какого-либо сервиса достаточно просто удалить соответствующую ссылку (SXXlalala) из необходимого каталога /rcX.d, а для разрешения — создать соответствующую ссылку в нужном каталоге /rcX.d. Однако не следует забывать помимо стартовой ссылки создавать стоповую, иначе возможны проблемы, когда система некорректно завершит функционирование сервиса, для которого забыли создать стоповую ссылку. А как же корректно установить порядковый номер у соответствующей ссылки? Конечно, можно чисто эмпирически подобрать номер, исходя из функций, выполняемых сервисом. Но давайте заглянем в любой файл в каталоге /etc/rc.d/init.d/, к примеру, в файл anacron:

```
#!/bin/sh
# Startup script for anacron
# chkconfig: 2345 95 05
# description: Run cron jobs that were left out due to downtime

# Source function library.
. /etc/rc.d/init.d/functions
[ -f /usr/sbin/anacron ] || exit 0
prog="anacron"
start() {
    echo -n $"Starting $prog: "
    daemon anacron
    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/anacron
    echo
    return $RETVAL
}
stop() {
    if test "x"pidof anacron"" != x; then
      echo -n $"Stopping $prog: "
```

```
      killproc anacron
      echo
    fi
    RETVAL=$?
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/anacron
    return $RETVAL
}
case "$1" in
      start)
      start
      ;;
      stop)
      stop
      ;;
      status)
      status anacron
      ;;
      restart)
      stop
      start
      ;;
      condrestart)
      if test "x"pidof anacron"" != x; then
        stop
        start
      fi
      ;;
      *)
      echo $"Usage: $0 {start|stop|restart|condrestart|status}"
      exit 1
esac
exit 0
#####################################################################
```

Обратите внимание на часть заголовка файла:

```
#!/bin/sh
# Startup script for anacron
# chkconfig: 2345 95 05
# description: Run cron jobs that were left out due to downtime
```

Помимо указания, какой командной оболочкой необходимо воспользоваться, там есть строчка

```
# chkconfig: 2345 95 05
```

из которой следует, что этот скрипт может использоваться во втором, третьем, четвертом и пятом уровнях выполнения, а цифры 95 и 05 — порядковый номер для стартового (95) и стопового (05) скриптов. Обратите внимание — в сумме эти две цифры составляют 100. Таким образом, достаточно просто добиться того, чтобы порядок останова сервисов был в точности обратный стартовому. Description в данном файле — комментарий, который linuxconf выдает на экран для объяснения роли данного сервиса.

Если внимательно посмотреть скрипт, то сразу видно, что опций у него больше, чем стандартные start и stop. Имеются еще restart, condrestart и status. Старт, останов и проверка состояния демона выполняются рядом функций типа daemon, killproc, status. Процедуры daemon, killproc, status — определяются в файле /etc/rc.d/init.d/functions (а тот пользуется определениями из /etc/sysconfig/init). Они предназначены для старта, останова и проверки статуса демона (сервиса).

Функция daemon обеспечивает старт сервиса. При этом можно учесть особенности поведения демона. Перед стартом сервиса всегда делается проверка наличия его в системе. Так как появление дампа (дамп — моментальный снимок памяти, используемой зависшей программой на момент ее краха) памяти сервиса может привести к проблемам с безопасностью, то все демоны запускаются в режиме без создания дампа памяти.

Останов сервиса выполняется процедурой killproc. Данная функция предполагает один аргумент в виде имени демона и, при необходимости, еще один для указания сигнала, который будет послан сервису. Сигнал SIGKILL часто может быть нежелателен для останова сервиса, поэтому, если сигнал назначен, то используется только он, в противном случае сперва посылается SIGTERM и, если данный сигнал не произвел на процесс впечатления, то посылается сигнал SIGKILL. В последнюю очередь скрипт подчищает различные блокировочные файлы.

Функция status позволяет проверить текущее состояние сервиса. Если сервис нормально функционирует, то просто сообщается об этом факте. В противном случае осуществляется проверка на наличие флаговых файлов (/var/run/подсистема.pid и /var/lock/subsys/подсистема), которые должны блокировать повторный запуск сервиса. Таким образом, несложно самому создать скрипт для управления сервисом.

rc.local

Файл /etc/rc.d/rc.local выполняется после скрипта rc. В него рекомендуется помещать дополнительные сервисы или персональные настройки. Однако обычный пользователь редко использует эту возможность.

Другие файлы, влияющие на процесс загрузки

Все файлы конфигурации, задействованные при загрузке системы, находятся в каталоге /etc:

- /etc/fstab — содержит информацию об автоматически монтируемых при старте файловых системах;
- /etc/skel — образцы файлов конфигурации, используются при создании учетных записей новых пользователей;
- /etc/bashrc — общесистемный файл для командной оболочки;
- /etc/initscript — файл, позволяющий задать специфические действия для каждой команды из файла /etc/inittab (подробную информацию следует искать в справочной системе).

Второстепенные файлы конфигурации:

- /etc/issue — сообщение, выдаваемое системой до приглашения "login:";
- /etc/motd — сообщение, выдаваемое системой после регистрации пользователя.

Процессы, происходящие при регистрации пользователя

Последовательность событий при регистрации пользователя:

1. Пользователь вводит свое регистрационное имя (login) по приглашению "login:" процесса getty;
2. Процесс getty выполняет программу login, передавая программе login в качестве аргумента регистрационное имя пользователя;
3. Программа login запрашивает пароль и сверяет регистрационное имя и пароль пользователя с записанными в файле /etc/passwd (login) и /etc/shadow (пароль). При этом введенный пароль пользователя шифруется по специальному алгоритму (в последнее время чаще всего используется алгоритм MD5), и полученный результат сравнивается с зашифрованным паролем, хранящимся в /etc/shadow;
4. В случае, если регистрационное имя пользователя или пароль не совпали с хранящимся в системе, то после паузы (около 3 секунд, задержка настраивается) выводится сообщение `Password incorrect`. Программа login завершает свою работу, а процесс getty снова выводит приглашение "login:";
5. Если проверка регистрационного имени и пароля пользователя прошла успешно, login выводит на экран из файла /etc/motd так называемое "сообщение дня";

6. После этого login запускает командную оболочку (shell), указанную в бюджете пользователя, и устанавливает переменную среды TERM;
7. Оболочка shell выполняет файлы, исполняемые при регистрации пользователя в системе, сперва общесистемные, а затем пользовательские (если это Bourne-shell, выполняется файл .profile, если C-shell — .login и .cshrc, если Korn-shell — .profile и .kshrc). В этих файлах можно указать специфические настройки пользователя, переменные окружения, запустить какие-то приложения. После этого shell выводит на экран приглашение и ожидает ввода информации.

Основные файлы, участвующие в регистрации пользователя

В регистрации пользователя участвуют следующие файлы:

- /etc/profile — общесистемный профильный файл, устанавливает пути и другие важнейшие переменные;
- /etc/passwd — различная регистрационная информация, такая как имя пользователя, группа пользователя, домашний каталог и командный интерпретатор;
- /etc/shadow — в определенной мере дублирует файл passwd, но его основное назначение — хранить пароли пользователей;
- /etc/bashrc — общесистемный файл конфигурации bash;
- /домашний каталог/.* — пользовательские файлы конфигурации.

Если требуется, чтобы при регистрации пользователя выполнялся какой-то скрипт или устанавливались переменные окружения, то для этого надо вызов данного скрипта поместить в ~/.profile или в .bash_profile.

Если необходимо, чтобы пользователь не мог отменить выполнение какого-то скрипта или команды при его регистрации в системе, то для этого следует вписать в ./etc/profile следующее:

```
if test $USER = petya; then
        echo Hello Petya!
#здесь запускается ваш скрипт
fi
```

Эти команды будут исполняться только при регистрации в системе пользователя petya.

Загрузка в однопользовательском режиме

Помимо стандартной загрузки, описанной ранее, можно заставить процесс init загрузить систему на уровне выполнения, отличном от загружаемого по

умолчанию. Эта возможность крайне необходима в случае, когда у операционной системы есть проблемы. Это могут быть чьи-то неудачные эксперименты с файлом inittab или неправильно сконфигурированный процесс (например, если сеть нормально не настроена, sendmail пытается найти хост, которого нет. Это может привести к задержке при старте системы в 10 минут и более). При возникновении такой ситуации необходимо перевести систему в однопользовательский режим и решить проблему. Для этого в строку приглашения `LILO (boot:)` надо ввести аргументы `single` или `emergency`. Это позволит загрузить систему в однопользовательском режиме (уровень выполнения 1), при котором в системе работает только суперпользователь (root) и запускается очень небольшое число самых необходимых системных служб, включая login. Другим способом перевода системы в однопользовательский режим является применение команды `telinit`, которая, собственно, является символической ссылкой на сам init. Команда `telinit`, помимо перевода системы с одного уровня выполнения в другой, может заставить процесс init перечитать файл inittab без перезагрузки операционной системы.

Однопользовательский режим необходим для выполнения административных задач, таких как проверка и восстановление файловой системы. К примеру, запуск fsck в разделе /usr. Обычно необходимость перехода в однопользовательский режим возникает тогда, когда fsck не может автоматически восстановить файловую систему при загрузке. Такое случается редко, обычно при пропадании электроэнергии во время работы компьютера или выходе из строя жесткого диска.

Однако бывают случаи, когда в однопользовательском режиме загрузиться с жесткого диска невозможно. В этом случае можно загрузиться в однопользовательском режиме с дискеты или с CD-ROM. Такое может произойти в случае серьезного краха системы (что иногда бывает смертельно для информации на жестком диске) или когда, к примеру, при установке Windows переписывает MBR, и в результате уничтожается загрузчик LILO (разделы Linux и система жизнеспособны, только загрузить ее нечем). Более подробную информацию *см. в гл. 40*.

Из соображений безопасности правильно сконфигурированная система при загрузке в однопользовательском режиме запросит пароль пользователя root.

Существует еще один способ вмешаться в процесс загрузки. В процессе загрузки системы выдается сообщение `Press "I" to enter interactive startup`. Если вы нажмете клавишу <I>, система перейдет в режим пошагового выполнения загрузки сервисов (подобно нажатию клавиши <F8> в Windows и выбора режима загрузки step by step).

Это позволит отказаться от загрузки подозрительных (с вашей точки зрения) процессов и определить, при запуске какого процесса возникают неприятности.

Утилиты

Подводя итог, перечислим утилиты, участвующие в процессе загрузки системы:

- init — программа, управляющая загрузкой операционной системы;
- telinit — утилита для управления процессом init;
- runlevel — выводит текущий уровень выполнения;
- linuxconf — утилита конфигурации операционной системы Linux. В том числе позволяет редактировать список сервисов, запускаемых в текущем уровне выполнения;
- ntsysv — консольная утилита для редактирования списка сервисов, запускаемых в текущем уровне выполнения;
- /usr/sbin/setup — консольная утилита для конфигурирования операционной системы;
- control-panel — графическая утилита для конфигурирования операционной системы.

Ссылки

- **www.osp.ru/os/2001/02/073.htm** — И. Облаков. Восход солнца вручную.
- /usr/src/Linux-2.4.3/Documentation/ — много информации, так или иначе связанной с ядром операционной системы, драйверами, файловыми системами и т. п.
- Справочные страницы man — init, inittab, telinit, initscript.
- Соответствующие HOWTO *(см. гл. 13)*:
  - Ethernet-HOWTO — различные тонкости настройки сетевых адаптеров;
  - The Linux BootPromt HOWTO — справочник по аргументам начальной загрузки, передаваемым ядру Linux во время загрузки системы;
  - The Linux Bootdisk HOWTO — создание загрузочной дискеты.

Глава 7

Безопасная работа в Linux

В этой главе мы в концептуальном плане, особо не вдаваясь в подробности, рассмотрим вопросы увеличения безопасности операционной системы Linux. Подробности вы всегда найдете в соответствующей литературе и главах, описывающих конкретное программное обеспечение.

Основные положения

Зачем вам безопасность?

Безопасность компьютеров и компьютерных сетей сейчас, в связи с повсеместным распространением Интернета и электронной коммерции, все больше выходит на первый план. В любой серьезной организации при приеме на работу системного администратора одним из основных требований к нему является умение организовать безопасность системы и сетей.

Даже если вы устанавливаете один-единственный сервер, не исключена возможность, что кто-то попробует взломать его просто от скуки. Поэтому особо нельзя надеяться, что на ваш сервер или Web-страницу никто не покусится.

Надежность защиты системы

Следует помнить: "все, что один человек построил, другой всегда сможет поломать". Надежность защиты, в идеале, должна соответствовать следующему правилу: затраты взломщика на преодоление защиты должны существенно превышать стоимость поврежденных или украденных данных. Это, конечно, не означает, что домашнюю систему, на которой ничего важного нет, защищать не надо. Просто необходимо соразмерять затраченные усилия — для домашнего пользователя их надо потратить гораздо меньше, чем для банковской сети.

У защиты есть одна особенность — чем более безопасна система, тем больше усилий необходимо затрачивать на поддержание ее в рабочем состоянии, и тем более навязчивой становится сама система безопасности. Всегда необ-

ходимо соблюдать золотую середину между безопасностью системы и неудобствами пользователей, связанными с режимом безопасности.

Защита не бывает идеальной. Вы должны представлять, сколько времени и усилий необходимо потратить на восстановление или воссоздание данных при их потере.

Определение приоритетов защиты

До того, как начать настраивать безопасность системы, необходимо определить, от чего и как будет защищена ваша система, какие службы должны быть защищены в первую очередь и особо. Следует четко представлять, какая информация или какое оборудование наиболее ценно, что ни при каких обстоятельствах не должно пропасть или попасть к постороннему человеку, а что имеет минимальную ценность (было бы нелепо выстроить супермощную систему для защиты сочинений вашего ребенка и установить минимальную безопасность для банковских документов, стоящих миллионы).

Кроме того, всегда следует помнить — никаких поблажек для пользователей вне зависимости, насколько высокое положение они занимают. Пользователь, который имеет привилегии, неизбежно является брешью в безопасности.

Политика безопасности

Если ваша задача — администрирование средней или большой сети, необходимо разработать документ, который называется "Политика безопасности" и определяет права и обязанности пользователей, меры по защите системы и действия, применяемые в случае нарушения безопасности системы.

Этот документ должен быть утвержден руководством фирмы, и с ним следует ознакомить каждого сотрудника, причем желательно под расписку. Чем проще и понятнее составлен документ, тем больше вероятность, что его поймут и будут им руководствоваться. Главное правило, которое должно быть четко зафиксировано: "То, что не разрешено — запрещено".

Основные направления защиты

Прежде чем что-то защищать, необходимо знать, что, от кого и от чего.

- Физическая защита:
  - защита от физического проникновения в компьютер;
  - защита от аппаратных сбоев.
- Защита рабочей станции (локальная безопасность).
- Защита сервера:
  - защита от пользователей;
  - защита от внешнего мира.

- Защита сети:
  - защита от проникновения извне;
  - защита от взломов изнутри.

Физическая безопасность

Первое, что следует сделать, это позаботиться о физической безопасности системы. Нет никакого прока от программной защиты, если из сервера можно просто изъять жесткий диск и переписать данные. Необходимо выяснить, кто имеет прямой физический доступ к системе, а также можно (и нужно ли) защитить систему от их потенциально вредного воздействия.

Степень физической безопасности напрямую зависит от ситуации и финансовых возможностей. Домашнему пользователю, скорее всего, не нужна сильная физическая защита. Офисный компьютер может понадобиться обезопасить на время отсутствия пользователя. А если это сервер, то его физическая защищенность должна быть максимальна, в идеальном случае он должен находиться в бронированной комнате без окон, с сейфовым замком на стальной двери.

Замки

Практически любой серверный корпус имеет специальный замок или отверстия для установки навесного замка. Как правило, наружные отсеки для винчестеров тоже имеют замки.

Для корпуса обычного компьютера дело обстоит похуже. Сейчас редко встретишь компьютерный корпус, у которого есть замок блокировки клавиатуры, хотя еще лет пять назад он был обязательным элементом любого компьютерного корпуса. Но даже при наличии такого замка эта защита является крайне слабой — как правило, к замку подходит любой ключ от аналогичных корпусов, а с помощью обыкновенной канцелярской скрепки можно открыть его за несколько секунд. Тем не менее, для неопытного взломщика замок является непреодолимой преградой.

Хуже дело обстоит с ноутбуками — его можно просто взять со стола. Но любой современный ноутбук имеет специальное отверстие, куда крепится замок, похожий на велосипедный.

Охрана жесткого диска

Если на жестком диске хранится информация, которая ни в коем случае не должна попасть в чужие руки, и лучше ее уничтожить, чем допустить пропажу, то для этого необходимо предпринять дополнительные меры. Если у вас небольшой сервер или рабочая станция, имеет смысл жесткий диск, на

котором хранится информация, установить в специальную съемную корзину, так называемый Rack Mount. Это позволит по окончании работы извлекать жесткий диск из компьютера и прятать в сейф. Для больших серверов такое сделать несколько затруднительно, но иногда имеет смысл. В случае, если к вам заявляются гости, которым "невозможно отказать", можно практически моментально извлечь жесткий диск и уничтожить его, разбив об пол. (Автору известна фирма, возле сервера которой стоит охранник, в случае попытки прорыва в серверную обязанный выстрелом из пистолета уничтожить жесткие дискию.)

BIOS

BIOS является самым первым звеном, от которого зависят настройки компьютера, и попытка проникновения в BIOS будет произведена взломщиком в 99% случаев. Практически любая BIOS имеет возможность установить два типа паролей — на вход в систему и на вход в BIOS. Это не дает стопроцентной защиты (установки BIOS могут быть обнулены, если кто-то имеет доступ вовнутрь корпуса), но может быть хорошим сдерживающим фактором. Однако следует помнить, что в случае установки загрузочного пароля на сервере система не сможет загрузиться без вмешательства администратора. (Очень неприятно, когда в три часа ночи вас выдергивают из постели для того, чтобы вы ввели пароль в сервер.)

Загрузочные устройства

Если не установлен пароль в BIOS, велика вероятность, что взломщик попытается загрузить систему с помощью системной дискеты, CD-ROM-диска, Zip-дисковода и тому подобных съемных устройств. Это дает ему полный доступ к жестким дискам компьютера. Поэтому, помимо установки пароля на вход в BIOS, необходимо запретить в BIOS загрузку со всех устройств, кроме жесткого диска. Так же рекомендуется либо отключить в BIOS сервера дисководы съемных устройств, либо (при наличии такой возможности) физически отключить флоппи-, Zip- и тому подобные дисководы и CD-ROM, а иногда даже вообще не устанавливать их в сервер.

Безопасность загрузчика операционной системы

Загрузчики Linux также имеют возможность установки стартового пароля. Это позволяет предотвратить загрузку операционной системы без ввода пароля.

Программы xlock и vlock

Если вы отходите на какое-то время от своего рабочего компьютера и не хотите выключать его, используйте программы xlock и vlock. Эти программы

заблокируют доступ к системе, причем как визуально (черный экран или какая-то надпись типа "Консоль заблокирована"), так и с помощью клавиатуры:

- xlock предназначена для X Window. Она "запирает" дисплей и для продолжения работы запрашивает пароль;
- vlock — консольная программа, которая позволяет "запереть" часть или все виртуальные консоли системы.

Конечно, "запирание" консоли не позволит нанести прямой вред работе, однако не помешает перезагрузить машину (кнопку Reset или отключение питания еще никто не отменял).

Определение нарушений физической безопасности

При подозрении на попытку нарушения физической безопасности системы первым делом проверьте, что сразу бросается в глаза — наличие физических повреждений. Второе — определите, была ли произведена перезагрузка системы. Операционная система Linux очень надежна, и непроизвольные перезагрузки по ее вине исключены. Если операционная система непроизвольно перезагружается, причина, скорее всего, в аппаратной части. Известны случаи, когда установленный на столе компьютер перезагружался, если кто-то задевал стол. Причиной была микротрещина в материнской плате. Во всех остальных случаях перезагрузка системы производится администратором (пользователем) или по команде источника бесперебойного питания. Если компьютер перезагружен без вас — это проблема, требующая изучения.

Вот часть того, что следует проверить:

- короткие или незаконченные системные журналы;
- системные журналы, которые содержат неверные права доступа или права собственности;
- системные журналы, в которых присутствуют записи перегрузки или перезапуска сервисов;
- отсутствие системных журналов;
- подключение пользователя с нетипичного для него места или использование программы su пользователем, который никогда этого не делал.

Так же крайне желательно вести журнал перезагрузки системы с указанием даты, времени и причины перезагрузки. Время, прошедшее с момента перезагрузки, можно узнать из системного журнала или командой `uptime`.

Локальная безопасность

Сразу после установки системы необходимо позаботиться о защите от локальных пользователей. Достаточно много методов взлома основано на раз-

личных недочетах или ошибках программного обеспечения, доступного только им. Локальный пользователь и сам по себе потенциально опасен. Предоставьте ему чуть больше прав, чем следует — и одной простой командой `rm` он может снести половину операционной системы.

Регистрация новых пользователей

Следует разработать правила регистрации новых пользователей в системе, которые должны неукоснительно выполняться. Существует несколько правил, которых необходимо придерживаться при работе с пользователями:

- предоставлять минимальное количество привилегий;
- отслеживать, когда и откуда происходит регистрация пользователя;
- не забывать удалить пользователя, если он больше не работает в фирме;
- стараться максимально ограничить количество пользователей, имеющих нестандартную конфигурацию или привилегии.

Безопасность пользователя root

Наиболее желанным приобретением для взломщика является пароль суперпользователя (root). Поскольку этот пользователь в системе "царь и бог", проблемы, связанные с ним, для системы катастрофические. Следует заходить в систему под именем пользователя root как можно реже и, желательно, не через сеть. Существует несколько правил, которых необходимо придерживаться при работе в системе в качестве пользователя root:

- старайтесь избегать использования сложных комплексных команд или длинных одиночных команд. Велика вероятность того, что вы ошибетесь и выполните не то, что надо;
- используйте потенциально опасные команды (удаление, переименование, перенос файлов) со специальным ключом, который заставляет команду спрашивать, действительно ли вы хотите совершить эту операцию с файлами. Помните, удаленные файлы в операционной системе Linux восстановить невозможно;
- регистрируйтесь в системе как пользователь root только в экстраординарных случаях. Для выполнения отдельных команд вполне можно воспользоваться программами su и sudo;
- исключите из своего и пользовательского обихода r-утилиты — rlogin, rsh, rexec и тому подобные, а также программу telnet. Лучше просто удалите их из всех систем. Эти программы были хороши лет двадцать пять назад. Теперь же каждый второй взломщик пытается ими воспользоваться в корыстных целях. В качестве замены используйте пакет SSH;
- сначала хорошенько подумайте, что собираетесь делать, потом проверьте, правильно ли ввели команду. И только затем нажмите клавишу <Enter>.

Безопасность файлов и файловой системы

Множество взломов операционной системы увенчались успехом из-за неправильной установки прав доступа к файлам или проблем с файловой системой. Существует несколько правил, которых необходимо придерживаться при работе с установкой прав:

- ограничьте возможность запуска пользователем специальных команд или файлов. Для разделов, на которые может записывать данные любой пользователь, в файле /etc/fstab поставьте опцию `nosuid`. Так же можно использовать `nodev` (запрещает создание символьных и блочных устройств), `noexec` (запрет выполнения программ) и `ro` (монтировать раздел только для чтения);
- не рекомендуется использовать NFS. Если все же NFS установлена, применяйте максимальные ограничения;
- настройте маску для создания пользователями файлов в максимально ограничивающем режиме. Идеальный вариант — маска 077;
- установите квоты на использование файловой системы для пользователей. Также желательно запретить приложениям пользователя создавать дамп памяти программы на диске;
- старайтесь свести к минимуму количество SUID- и SGID-файлов в системе. Поскольку эти программы предоставляют пользователям, которые их запускают, специальные привилегии, необходимо убедиться, что небезопасные программы не установлены;
- обнаруживайте и удаляйте файлы .rhosts;
- прежде чем изменить права доступа для системных файлов, убедитесь, что понимаете, что делаете. Никогда не изменяйте права доступа файла только потому, что это является простым способом заставить что-то работать;
- периодически проверяйте права доступа ко всем важнейшим файлам системы. Изменение прав доступа к файлам — один из основных признаков взлома системы.

Проверка целостности файлов

Хороший способ обнаружения атаки на операционную систему — проверка целостности файлов. Существует несколько пакетов, с помощью которых можно организовать проверку системы. Простейший случай — программа rpm, с помощью которой можно сверить все файлы установленных в систе-

ме пакетов. Однако эта программа не может сверить файлы, попавшие в систему минуя `rpm`.

Программа Tripwire вычисляет контрольные суммы для всех важных бинарных и конфигурационных файлов в системе и сравнивает их с предыдущими записями, хранящимися в базе данных. Никто не мешает написать скрипт, автоматизирующий эту процедуру. Рекомендуется так же хранить базу данных во избежание подмены ее взломщиком либо на дискете, либо на другом компьютере.

Особенности безопасности файловой системы Ext2

В файловой системе Ext2 присутствует поддержка дополнительных флагов для файлов, используемых для повышения безопасности системы. Ядро Linux версии 2.4 позволяет работать со следующим набором атрибутов:

- `A` — `Atime`. Система не модифицирует access time для данного файла;
- `S` — `Sync`. Система фиксирует все изменения, происходящие в данном файле на физическом диске синхронно с приложением, изменяющим данный файл;
- `a` — `append`. Система позволяет открывать данный файл с целью его дополнения и не позволяет никаким процессам перезаписывать или усекать его. Если данный атрибут применяется к каталогу — процесс может создавать или модифицировать файлы в этом каталоге, но не удалять их;
- `i` — `immutable`. Система запрещает любые изменения данного файла. Если данный атрибут применяется к каталогу — процессы могут модифицировать файлы, уже содержащиеся в данном каталоге, но не могут удалять файлы или создавать новые;
- `d` — `no dump`. Программе, создающей дампы системы, дается указание игнорировать данный файл во время создания резервной копии;
- `c` — `compress`. Система использует прозрачную компрессию для данного файла;
- `s` — `secure deletion`. Удаление такого файла сопровождается записью блоков диска, на которых он располагался, нулями;
- `u` — `undelete`. Когда приложение запрашивает файл на удаление, система должна сохранить его блоки на диске, чтобы потом его можно было восстановить.

Несмотря на то, что файловая система поддерживает данный набор атрибутов, у ядра и различных приложений остается выбор, учитывать или не учитывать их.

К сожалению, ядро Linux версии 2.4 игнорирует флаги `c`, `s` и `u`.

Флаг `A` или `Atime` для определенных файлов может дать некоторую прибавку производительности, т. к. избавляет систему от необходимости постоянно обновлять поле access time для этих файлов каждый раз, когда их открывают на чтение. Атрибут `S` или `Sync` увеличивает надежность сохранения данных ценой некоторой потери производительности системы.

Команды для установки и чтения атрибутов в Ext2

Есть две утилиты, специально предназначенные для установки и чтения данных атрибутов: chattr и lsattr.

Команда `chattr` используется для установки и снятия флагов:

- `chattr +Si test.txt` — установить флаги `sync` и `immutable` для файла test.txt;
- `chattr -ai test.txt` — убрать флаги `append-only` и `immutable` у test.txt;
- `chattr =aiA test.txt` — установить ограничение на использование только флагов `a`, `i` и `A`.

Команда `lsattr` выводит список файлов и каталогов с атрибутами и функционально напоминает команду `ls`.

Команда `lsattr -a test*`, например, выдаст на экран:

```
---i-------- test.conf
----a------- test.log
------------ test.txt
```

Использование для защиты файлов атрибутов файловой системы не является стопроцентной гарантией защищенности системы. Конечно, атрибуты `a` и `i` запрещают изменение защищенных файлов даже процессами, владельцем которых является root, однако в обычных обстоятельствах ничто не мешает пользователю root снять эти флаги. Тем не менее, есть возможность решить эту проблему.

Утилита lcap позволяет конфигурировать параметры ядра, и в том числе те, которые определяют работу файловой системы Ext2 с расширенными атрибутами. Вот наиболее важные вызовы lcap, которые нас интересуют:

- `lcap CAP_LINUX_IMMUTABLE`;
- `lcap CAP_SYS_RAWIO`.

Первый параметр запрещает процессам root изменять флаги `a` и `i`, а второй параметр запрещает низкоуровневый доступ к блочным устройствам, таким как диски, чтобы предотвратить изменение флагов, используя прямой доступ к файлам.

Пароли и шифрование

Стандартным атрибутом безопасности системы в наше время является пароль. Наиболее общие рекомендации по выбору паролей приведены ниже:

- длина пароля не должна быть менее, чем 8 символов;
- пароль должен состоять из букв, набираемых в разных регистрах, символов типа `# $ @ / . ,` и цифр;
- не рекомендуется использовать что-либо обозначающие слова;
- желательно периодически изменять пароли.

Семейство Linux для шифрования паролей использует односторонний алгоритм шифрования, называемый DES (Data Encription Standard, стандарт шифрования данных). Хэши паролей затем сохраняются в файле /etc/shadow. Использование одностороннего алгоритма шифрования исключает возможность провести расшифровку /etc/shadow для получения паролей. Однако наличие у взломщика файла /etc/shadow значительно облегчает подбор пароля пользователя программами типа John the Ripper. На современных машинах шестисимвольный пароль подбирается этой программой за пару часов. PAM-модули (такие как MD5 или подобные) позволяют использовать различные алгоритмы шифрования для паролей.

Протоколы шифрования трафика

Для надежной (в плане безопасности) передачи данных в Интернете используются следующие методы и протоколы:

- SSL — Secure Sockets Layer, метод шифрования, разработанный Netscape для обеспечения безопасности в сети. Он поддерживает несколько различных протоколов шифрования и обеспечивает идентификацию как на уровне клиента, так и на уровне сервера. SSL работает на транспортном уровне и создает безопасный шифрованный канал данных. Чаще всего он используется при посещении пользователем защищенного Web-узла;
- S-HTTP — интернет-протокол, реализующий сервис безопасности;
- S/MIME — Secure Multipurpose Internet Mail Extension, стандарт шифрования, используемый в электронной почте или других типах сообщений в Интернете.

SSH

SSH (Secure Shell) — программа, позволяющая зарегистрироваться на удаленном сервере и иметь шифрованное соединение. SSH используется вместо устаревших и небезопасных утилит rlogin, rsh и rcp. Применяемый протокол использует шифрование с помощью открытого ключа как для шифрования соединения

между двумя машинами, так и для опознавания пользователей. Существует также несколько бесплатных реализаций SSH-клиентов для Windows.

PAM

PAM (Pluggable Authentication Modules) — унифицированный метод идентификации. Практически все современные приложения, которые используют идентификацию пользователя, имеют соответствующий модуль PAM. Это позволяет пользователю "на лету" изменять методы идентификации, требования, инкапсулировать все локальные методы идентификации без перекомпиляции программ.

Вот что можно делать с PAM:

- использовать различные алгоритмы шифрования для своих паролей;
- устанавливать лимиты на ресурсы для пользователей;
- "на лету" активизировать теневые пароли (shadow password);
- разрешать определенным пользователям регистрироваться только в определенное время и/или из определенного места.

CIPE

CIPE — криптографическая IP-инкапсуляция, шифрует данные на сетевом уровне. Шифруются пакеты, которые передаются между компьютерами в сети. CIPE можно также использовать при тунелировании (tunnelling) для создания виртуальных частных сетей (VPN, Virtual Private Networks). Преимущество низкоуровневого шифрования состоит в том, что оно разрешает прозрачную работу между двумя сетями, соединенными в VPN, без каких-либо изменений в программном обеспечении.

Kerberos

Kerberos является идентификационной системой, разработанной по проекту Athena в Массачусетском технологическом институте (MIT). Kerberos представляет сервер идентификации, услугами которого могут пользоваться компьютеры, подключенные к сети. Таким образом нет необходимости заводить на всех компьютерах учетную запись пользователя. Очень часто используется при модемном соединении провайдерами, имеющими несколько удаленных площадок.

CFS и TCFS

CFS — криптографическая файловая система. Это метод шифрования всей файловой системы, позволяющий пользователям сохранять в ней зашифро-

ванные файлы. Метод использует NFS-сервер, запущенный на локальной машине.

TCFS — прозрачная криптографическая файловая система, является улучшенным вариантом CFS, поскольку более интегрирована с файловой системой и, таким образом, прозрачна для всех пользователей, использующих зашифрованную файловую систему.

Безопасность ядра

Поскольку ядро контролирует поведение компьютера в сети, очень важно, чтобы ядро было максимально защищено от взломов. С этих позиций крайне желательно использовать последние стабильные версии ядер, а при компиляции ядра максимально исключить ненужные вам опции и драйверы устройств.

Устройства ядра

Устройства /dev/random и /dev/urandom служат для получения случайных чисел в любой момент времени. Эти числа используются в генераторах PGP-ключей (Pretty Good Privacy — общедоступная система кодирования информации с открытым ключом), SSH-вызовах и других аналогичных приложениях.

Устройство /dev/random является высококачественным генератором случайных чисел, основанным на временно-зависимых параметрах системы.

Работа устройства /dev/urandom подобна, однако она более быстра, но менее надежна.

Сетевая безопасность

Все наверняка слышали, как кто-то по сети взломал Web-сервер и испортил его содержимое или украл номера кредиток. Таких случаев становится все больше, поэтому крайне важно обеспокоиться сетевой безопасностью. Не следует, однако, забывать, что атака может проистекать в равной мере как из Интернета, так и из внутренней сети фирмы, поэтому крайне неразумно защищать сеть от атак снаружи и ничего не предпринимать для защиты от атак из внутренней сети.

Packet Sniffers

Одним из наиболее общих методов взлома сетевых машин является применение сниферов (Packet Sniffer — в дальнейшем просто снифер — програм-

ма, позволяющая перехватывать сетевые пакеты, предназначенные для других компьютеров. Первоначально использовалась для анализа сетевого трафика) с уже взломанного компьютера вашей сети. Эта программа перехватывает все Ethernet-пакеты сети и сканирует их на наличие слов Password, Login или su. Таким образом без особых усилий взломщик получает множество паролей для систем, которые он даже и не пробовал пока взламывать. Поэтому крайне нежелательно использовать сетевые сервисы, передающие пароли в незашифрованном виде. Этот способ взлома напрямую связан с обеспечением физической безопасности, т. к. посторонний может просто принести с собой ноутбук и подключиться с его помощью к внутренней сети фирмы.

Использование SSH или других методов шифрования паролей сводит к нулю эффективность этого способа взлома.

Системные сервисы

Прежде чем подключить систему к сети, следует подумать, какие сервисы будут предоставляться системой. Чем меньше запущенных сервисов, тем меньше вероятность взлома системы. Можно также ограничить список компьютеров, для которых разрешено использование сервисов вашего компьютера. Для этого в файле /etc/hosts.allow необходимо прописать только те компьютеры, которым разрешено иметь доступ к сервисам на вашем компьютере. Для запрещения доступа "подозрительных" систем следует использовать файл /etc/hosts.deny. Также проверьте ваши каталоги /etc/rc.d/rcN.d на предмет наличия запуска сервисов, которые вам не нужны.

Однако нельзя огульно выбросить все, что, как вам кажется, не используется. К примеру, удаление сервиса в файле /etc/services приводит к тому, что локальный клиент также не сможет использовать этот сервис.

DNS

Поддержка достоверной DNS-информации о всех компьютерах в сети также помогает повысить безопасность. Зачастую при атаке производится подмена DNS-информации, что облегчает взломщику проникновение в систему.

identd

Программа identd фиксирует информацию о том, какой пользователь запускает какой TCP-сервис. С точки зрения повседневной жизни задача, вроде бы, бесполезная, однако выдаваемая ей информация может пригодиться при анализе взлома системы.

Сетевые сканеры

Существует целый класс программных пакетов, которые выполняют сканирование портов и сервисов в компьютерных сетях. Сетевые сканеры используются администратором системы для определения уязвимых мест системы, однако никто не мешает воспользоваться ими и злоумышленнику. Наиболее известными, хотя уже и устаревшими, представителями этого класса программ являются SATAN и ISS. SATAN (инструмент администратора безопасности для анализа сетей) является сканером портов с Web-интерфейсом. Он может быть полезен для выполнения легкой, средней или тщательной проверки машины или сети машин. ISS (сканер безопасности Интернета) также является сканером портов. Он быстрее чем SATAN, но SATAN предоставляет больше информации.

Электронная почта

Одним из наиболее важных сетевых сервисов является сервер электронной почты. К сожалению, это наиболее часто атакуемый взломщиками сервис, поскольку он уязвим просто из-за огромного числа задач, которые выполняет, и привилегий, которые ему обычно нужны. Поэтому крайне желательно обновлять свой сервер электронной почты.

"Отказ в предоставлении доступа"

Очень популярный в последнее время вид атаки. Смысл ее напоминает пословицу — "Сам не гам, и другому не дам". Взломщик пытается искусственно загрузить некоторые сервисы настолько, чтобы они не могли отвечать на запросы или запрещали доступ к вашей машине законным пользователям. Имеется несколько разновидностей такой атаки:

- SYN flooding — сетевая атака "отказ в предоставлении доступа". Использует преимущества "лазейки" (loophole) в методе создания TCP-соединения. Последние версии ядер Linux имеют несколько конфигурационных настроек для предотвращения SYN Flooding-атак;
- Ping flooding — простая грубая реализация атаки "отказ в предоставлении сервиса". Взломщик посылает компьютеру "поток" ICMP-пакетов. Если атака происходит с компьютера с большей полосой пропускания, нежели имеет ваш компьютер, или с нескольких компьютеров одновременно, то ваша машина будет лишена возможности посылать что-либо в сеть. При вариации этой атаки, называемой "smurfing", на определенный сервер посылается поток ICMP-пакетов с обратным IP-адресом вашей машины;
- атака Ping o Death использует тот факт, что поступающие ICMP-пакеты ECHO REQUEST могут быть больше, нежели может вместить структура данных ядра, которая сохраняет эту информацию. Из-за приема единич-

ного большого (65 510 байтов) ping-пакета многие системы зависали, поэтому эта атака быстро обрела свое название;

- Teardrop/New Tear — атака, основанная на ошибке, присутствующей в коде фрагментации IP в Linux- и Windows-платформах. Она была исправлена еще в ядре версии 2.0.33.

Безопасность NFS

Система NFS позволяет серверам предоставлять целые файловые системы для других машин со встроенной в ядро поддержкой NFS. В экспортируемых файловых системах существуют довольно небольшие возможности реализации безопасности. Если вы вынуждены использовать NFS, прежде всего убедитесь, что предоставляете доступ только тем машинам, которым это действительно нужно. Никогда не экспортируйте полностью ваш корневой каталог.

Firewall

Firewall (брандмауэр, сетевой экран), используя определенный набор правил, ограничивает прохождение информации как внутрь, так и за пределы вашей локальной сети. Обычно компьютер, выполняющий роль брандмауэра, соединен с Интернетом и вашей локальной сетью, и доступ к Интернету из локальной сети выполняется только через него. Брандмауэр является полезным инструментом в обеспечении безопасности вашей сети. Однако не следует забывать о безопасности только из-за того, что у вас установлен брандмауэр. Если нарушитель прорвался через брандмауэр или действует изнутри сети, у него будет огромное поле деятельности.

Существует большое количество типов и методов организации брандмауэра. Более подробную информацию вы получите, прочитав великолепную книгу "Брандмауэры в Linux" *(см. приложение 5)*.

Администрирование системы

После установки и настройки системы вы сделали ее настолько безопасной, насколько это было возможно. Теперь необходимо выполнить несколько действий, чтобы быть подготовленным на случай ее взлома или аварии.

Резервная копия системы

Администрированию и резервированию операционной системы посвящено много интересных публикаций, так что особо останавливаться на этом процессе мы не будем. Однако рассмотрим несколько общих соображений.

В настоящее время существует несколько малобюджетных вариантов резервирования системы:

- использование жестких IDE-дисков. Сегодня приличный жесткий диск емкостью 20 Гбайт стоит порядка восьмидесяти долларов. Как представляется, это небольшая плата за систему, которую отлаживали долго и упорно, а тем более, если в ней хранятся важные данные. Еще одно преимущество резервного копирования на жесткий диск — его можно сконфигурировать так, чтобы было достаточно подключить винчестер в сервер, и система практически сразу становится полностью работоспособна;
- использование привода CD-RW. Очень экономичный вариант. При стоимости одного диска CD-R менее шестидесяти центов, а диска CD-RW около доллара, это, пожалуй, самый экономичный вариант;
- использование привода DVD-RAM. Позволяет резервировать достаточно большие объемы данных;
- привод Zip. Удобен тем, что существует внешний вариант исполнения;
- привод Jazz. Емкость диска порядка одного гигабайта;
- накопители на магнитооптике. Существуют разные модели с разной емкостью дисков. Достаточно дороги;
- ленточные накопители. Существуют разные модели с разной емкостью. Достаточно дороги.

Сразу после создания резервных копий на лентах и других перезаписываемых носителях необходимо поставить защиту от записи. Сохраняйте ваши резервные копии в надежных недоступных местах. Периодически проверяйте восстанавливаемость резервной копии. Периодически устраивайте "боевые учения" по восстановлению системы.

Режим резервирования

Существует несколько стратегий резервирования, и только вам решать, какая стратегия резервирования подходит. Достаточно универсальной является следующая стратегия:

- в конце рабочей недели делается полная резервная копия системы;
- в течение недели делается нарастающее резервирование системы (то есть резервируется изменение данных по сравнению с прошлым днем);
- при особо важных изменениях в системе резервная копия делается немедленно.

Однако достаточно часто используется резервирование не всей системы, а только особо важных данных (например, базы данных), но с совершенно другим интервалом (к примеру, каждый час).

Существуют специальные программные пакеты, позволяющие писать сценарии автоматизированного сохранения и восстановления данных. Для простых схем резервирования достаточно часто используются скрипты, написанные администратором системы.

Резервирование RPM-базы

При взломе системы взломщик обычно модифицирует для своих нужд несколько файлов, устанавливаемых при инсталляции из пакетов RPM. Если есть подозрение на взлом системы — одним из первых действий будет проверка целостности этих файлов. Однако, если работал опытный взломщик, велика вероятность того, что он подправит нужным ему образом базу установленных RPM-пакетов или, вообще, уничтожит ее. На функциональности системы отсутствие или повреждение базы установленных RPM-пакетов не влияет, но при этом теряется возможность проверки целостности установленных пакетов. Поэтому крайне желательно периодически копировать базу RPM (/var/lib/rpm/*) на съемный носитель и хранить его отдельно.

Для проверки целостности установленных пакетов можно воспользоваться командой

```
rpm -Va
```

Но не забывайте после установки или удаления пакетов обновлять резервную копию базы RPM.

Файлы регистрации

Первое, что делает опытный взломщик системы после успешного проникновения в нее — заметает следы. А поскольку ведутся специальные файлы регистрации пользователей в системе, подключений к сетевым сервисам и тому подобных событий, вполне очевидным действием взломщика является уничтожение или модификация файлов журналов. Поэтому крайне важно сохранить эти файлы в неприкосновенности. Начинать надо с ограничения списка пользователей, способных читать и писать в каталог /var/log.

Регулярно инспектируйте свои журнальные файлы. Большое количество неудачных попыток регистрации или сканирование портов с одного и того же компьютера может указывать на попытку вторжения. Чтобы узнать, где ваш дистрибутив ведет системные журналы, нужно посмотреть в файл /etc/syslog.conf, который указывает syslog, куда записывать различные сообщения. Если вы заметили, что в журнальных файлах кто-то похозяйничал, необходимо определить, когда это началось и каких процессов касалось. Лучше всего в такой ситуации восстановить журналы с резервных копий и определить момент взлома.

Можно настроить syslog так, чтобы он отсылал копию наиболее важных данных на безопасную систему. Это предотвратит попытки взломщика скрыть свою деятельность путем удаления информации о его действиях в системе. Более подробную информацию по syslog.conf можно найти на соответствующей странице помощи (man-странице).

Обновляйте операционную систему

Большинство систем Linux устанавливаются с компакт-дисков. Но жизнь не стоит на месте, выходят различные обновления и исправления программ. К примеру, к дистрибутиву Red Hat только официальных обновлений за четыре месяца набралось более ста тридцати мегабайт. И это далеко не все обновления! Однако в погоне за номером версии не стоит устанавливать самое свежее программное обеспечение. Есть одно неплохое правило "Работает — не трогай!" Нашли в программе ошибку — обновите. Хотите получить самую свежую версию программы — подождите неделю-другую после ее выхода. Пусть на грабли наступают другие. За это время наверняка в безопасности системы найдутся прорехи или в программе ошибки. Однако после обнаружения ошибки в безопасности программы, не затягивайте с ее обновлением — велик шанс, что вашу систему попытаются взломать, используя именно эту брешь.

Действия во время и после взлома системы

Если вы обнаружили, что ваша система взломана — не паникуйте, расслабьтесь. Поспешные действия еще никого до добра не доводили.

Нарушение безопасности

Проще всего обнаружить физический взлом системы или подключение к вашей сети. В каждой фирме есть служба безопасности, воспользуйтесь ее услугами или вызовите представителей правоохранительных органов.

Если обнаружено вторжение в сеть, первым делом отсоедините вашу сеть. Если это представляется невозможным, запретите доступ из сети взломщика или заблокируйте пользователей в системе. После того как будет сделано что-либо из вышеперечисленного (отсоединена сеть, запрещен доступ из сети взломщика или заблокированы его учетные записи), следует ликвидировать все его пользовательские процессы.

Некоторое время после этого необходимо отслеживать состояние системы, поскольку попытка взлома может повториться, причем необязательно от имени этого же пользователя или с того же сетевого адреса.

Взлом системы произошел

Взлом обнаружен. Что дальше?

Закрытие бреши

Если вы четко знаете, каким образом взломщик проник в систему, постарайтесь сразу же закрыть эту брешь. К примеру, взлом произошел через сервер Samba. Самый простой выход — завершить процесс и отправиться в Интернет искать решение проблемы. Как правило, существует обновленная версия программы или какой-либо список исправлений известных ошибок.

Но это еще не означает, что вы в безопасности. Проверьте все ваши журнальные файлы на предмет сомнительных событий. Проверьте существование более свежих версий ключевого программного обеспечения и обновите его.

Оценка повреждений

Оцените повреждения. Выясните, что было нарушено. Не исключено, что в результате взлома проще переустановить систему, чем пытаться восстановить. Правда, такие тяжелые повреждения встречаются не часто.

Так как Linux достаточно легко инсталлировать, рекомендуется создать специальный конфигурационный файл kickstart, содержащий список установленных в системе пакетов, который затем используется при инсталляции системы. А конфигурационные файлы следует заранее переписать. Рекомендуется также восстановить систему из резервной копии, поскольку наверняка резервная копия содержит важные данные. Однако нужно очень точно определить момент взлома системы, чтобы не случилось так, что восстановленная из резервной копии система уже содержит "закладки" взломщика.

Выслеживание взломщика

Взломщик заблокирован, система восстановлена, откуда пришел взломщик определено. Однако не следует забывать, что раз взломали вас, могут взломать и кого-то другого. Поэтому следует сообщить об атаке администратору системы, с которой была взломана ваша система (этого администратора можно найти с помощью базы internic). Пошлите ему описание процесса взлома, описание нанесенных повреждений и приложите содержимое системных журналов с датой и временем событий. Как правило, система, откуда была произведена атака, оказывается тоже взломанной, но администратор об этом даже не подозревает.

Опытные взломщики используют большое количество промежуточных, посреднических систем. Поэтому не стоит сразу предъявлять претензии администратору системы, откуда произошел взлом. Как говорят китайцы: "Не теряйте лицо". Будьте очень вежливы с администраторами других систем при выслеживании взломщика, и они сделают все возможное для его поимки.

Ссылки

В Интернете существует очень много узлов, посвященных безопасности систем UNIX и специфике безопасности Linux. Обязательно подпишитесь на списки рассылки по вопросам безопасности и анонсы свежих выпусков программ. В частности, большой список русскоязычных рассылок, в том числе и по Linux, можно найти на сайте **www.citycat.ru**. Кроме того, можно посетить и эти адреса:

- **www.rootshell.com** — сайт, полезный для изучения современных методов взлома, которые сейчас используют взломщики;
- **www.netspace.org/lsv-archive/bugtraq.html/** — содержит советы в области безопасности;
- **www.aoy.com/Linux/Security/** — хороший узел по безопасности в Linux.

Вопросы безопасности затрагивают документы, находящиеся и по нижеприведенным ссылкам.

- **www.linuxdocs.org** — Network Administrators Guide (Руководство сетевого администратора).
- **linux.webclub.ru/books/linuxsos/index.html** — безопасность и оптимизация Linux. Редакция для Red Hat — русский перевод.
- **dc.internic.net/rfc/rfc2196.txt** — документ, посвященный политике безопасности системы.
- **www.consensus.com/security/ssl-talk-faq.html** — часто задаваемые вопросы по протоколу SSL.
- **www.kernel.org/pub/linux/libs/pam/index.html** — PAM-модули.
- **linux.webclub.ru/adm/attr_ext2.html** — Безопасность файловой системы Ext2. Michael Shaffer.
- **pw1.netcom.com/~spoon/lcap/** — Linux Kernel Capabilities Bounding Set Editor.
- **www.linuxdocs.org** — содержит соответствующие HOWTO *(см. гл. 13)*:
  - security-howto — документ, посвященный безопасности операционной системы;
  - hacker-howto — документ, посвященный взлому и защите от оного операционной системы;
  - NFS HOWTO — документ о настройке и использовании NFS — сетевой файловой системы;
  - Firewall-HOWTO — документ, посвященный настройке брандмауэра;
  - IP-Masquerade mini-howto — организация маскарадинга.

Глава 8

RPM

Фирма Microsoft и Windows уже приучили нас, что установка любой программы начинается с запуска программ Setup или Install. Затем, после согласия с лицензионным соглашением (по которому фирма-производитель обязывает вас установить программное обеспечение только на один компьютер, и, в свою очередь, сообщает, что не несет никакой ответственности за функционирование этого программного обеспечения), задается пара вопросов (куда и какие модули программного обеспечения установить) и все: "Программа установлена, перезагрузите, пожалуйста, компьютер". Достаточно быстро и просто.

Остаются, конечно, некоторые вопросы: "Почему я не могу убрать лишнюю функциональность, зачем эта игра перезаписала мой DirectX 8 своим DirectX 5, для чего в системе столько DLL для разных версий Visual Basic." Но, в общем, это намного проще, чем вручную копировать какие-то архивы, распаковывать их, править конфигурационные файлы, искать конфликты библиотек и версий, а в самом неприятном случае — получать ошибки компиляции или линковки и просматривать исходные тексты программ. Это еще один упрек Linux со стороны пользователей Windows. И как часто бывает, они не совсем правы. Да, в большинстве своем программы Linux поставляются в виде исходных кодов, упакованных в архив. Но так наиболее просто удовлетворить требование лицензии GNU, которая обязывает дистрибьютора программы в обязательном порядке предоставить потребителю ее исходный код.

Не следует также забывать, что программы разрабатываются не только для Linux, обычно их можно откомпилировать на многих UNIX-платформах, а в UNIX-мире стандартом de-facto для пакетов является так называемый "tarballs". Tarballs — это архивы, которые распаковываются утилитой tar (файлы с расширением tar) или gzip (файлы с расширением tar.gz). Поскольку Linux-программы по большей части распространяются через Интернет, проще выложить на FTP-сервер и скачать оттуда архив только с исходным кодом программы, чем выкачивать архив и с исходными кодами, и с откомпилированной программой. Кроме того, энтузиасты Linux, как правило, имеют привычку смотреть исходные коды программ, изменять их и

компилировать так, как им нравится (включать поддержку команд определенного процессора, добиваться максимального уровня оптимизации, различной степени выдачи отладочной информации и т. д.).

Однако с приходом в мир Linux пользователей, которые не желают учить опции компилятора, помнить, какие библиотеки установлены в системе, ждать по полчаса, пока откомпилируется программа и т. п., остро возник вопрос о стандартизации процесса установки программ в Linux. На сегодняшний день есть, по меньшей мере, три, или, если быть совсем точным, три с половиной способа установки программ. *Способ первый*, "старейшина" — программы распространяются в виде архивов исходных кодов *.tar.gz, которые необходимо распаковать и, в простейшем случае, откомпилировать командами `make`, `make install`. *Способ второй* — воспользоваться программой RPM (Red Hat Linux package management, Red Hat Linux менеджер пакетов) и, соответственно, пакетами RPM, содержащими уже откомпилированный код программ. *Способ "два с половиной"* — воспользоваться программой RPM и пакетами RPM с исходным кодом. Здесь два варианта: или получать исходный код пакетов, или делать из пакета с исходным кодом пакет с исполняемым кодом и устанавливать. *Способ третий* — разновидность второго — менеджер пакетов, входящий в дистрибутив Linux Debian. Возможно, в этой главе не будет упомянут какой-то другой способ инсталляции или менеджер пакетов. Мир Linux и Интернет настолько велики, что узнать или охватить все невозможно. Как уже упоминалось, значительная часть современных дистрибутивов тем или иным образом основаны на дистрибутиве Red Hat Linux, или, по крайней мере, имеют утилиты, способные работать с пакетами формата RPM. Поэтому эта глава полностью посвящена RPM-пакетам и RPM-менеджерам.

Система поддержки пакетов RPM

Во многом благодаря RPM, а так же удобной программе инсталляции Linux, дистрибутив Red Hat Linux завоевал огромнейшую популярность. Рассмотрим вкратце основные особенности RPM.

Для системного администратора RPM предоставляет следующие возможности:

- модернизировать отдельные компоненты системы или набора пакетов, сохраняя их конфигурацию;
- получать информацию об используемых пакетом файлах;
- получать информацию о зависимостях пакетов (необходимых библиотеках и т. д.);
- производить проверку пакетов;
- выдавать раздельно пакеты в авторском виде и сделанные к ним добавки;

- производить автоматизированное обновление пакетов (например, получение обновлений с FTP-сервера).

Благодаря этому с помощью RPM можно устанавливать, обновлять, удалять пакеты единственной командой в текстовом режиме или несколькими щелчками мышью в графическом менеджере пакетов. Пакет RPM содержит информацию о себе в заголовке пакета. Эта информация при установке пакета добавляется в базу данных установленных пакетов, где содержится информация о том, где находится пакет, какие дополнительные (supporting) пакеты ему необходимы и установлены ли они. Знатоки Windows могут заметить, что централизованная база данных установленных пакетов очень сильно напоминает часто критикуемый реестр Windows. Сравнение, однако, поверхностно. На самом деле, реестр Windows помимо списка установленных программ содержит в себе многочисленные системные настройки, без которых (повреждение или отсутствие реестра) не будет функционировать система в целом. Для Linux отсутствие или повреждение базы данных установленных пакетов вовсе не фатально. Как будет показано далее, базу данных всегда можно попытаться создать заново. Но не это главное. Отсутствие или повреждение базы данных никоим образом не сказывается на работоспособности системы — она полноценно функционирует. Могут возникнуть проблемы с обновлением или установкой пакетов, но их можно обойти с помощью специальных ключей программы RPM (принудительная инсталляция, отказ проверять зависимости и т. п.).

Принципы наименования пакетов

Имя пакета характеризует сам пакет, его версию, версию сборки исполняемых файлов (релиз) и архитектуру и задается в виде "имя_программы-версия-релиз.платформа" или "src.rpm".

Рассмотрим для примера пакет telnet-server-0.17-18.i386.rpm. По названию файла можно определить, что пакет содержит telnet-сервер версии 0.17, версия сборки файлов (релиз) 18 для данной версии пакета Red Hat Linux, собрана для процессора Intel 80386 и выше, формат файла — RPM. Файл пакета, у которого вместо архитектуры (например, i586) стоит src, содержит в себе исходные тексты программы. Иногда встречается немного другая структура именования пакета, например, apache-1.3.3-1.src.rpm. Здесь версия пакета — 1.3.3 — состоит из трех цифр. На инсталляционных дисках Red Hat и на FTP скомпилированные пакеты хранятся в каталоге RPMS, а пакеты, содержащие исходный код, — в каталоге SRPMS.

Структура пакета RPM в данной главе описана не полностью. Вкратце можно сказать, что в пакете содержатся исполняемые файлы, конфигурационные файлы, документация, все дополнительные файлы, напрямую связанные с пакетом, а также информация о том, куда должны устанавливаться файлы пакета и какие другие пакеты необходимы для его функционирова-

ния. После успешной установки пакета информация о нем заносится в базу данных системы RPM.

Достоинства RPM

К основным достоинствам RPM относятся:

- удобная установка программ;
- возможность инсталляции по FTP;
- проверка системы на наличие компонентов, необходимых устанавливаемому пакету;
- простое удаление пакетов из системы. При этом осуществляется проверка зависимостей пакетов системы от удаляемого пакета;
- обновление (Upgrade) пакетов с контролем версии, запрет установки пакета с более ранней версией, чем установленный в системе (Degrade);
- просмотр информации о пакете: что делает, кто сделал, где взять, файлы, содержащиеся в пакете, и т. д.;
- наличие общей иерархии пакетов, с помощью которой просто определить, к какой категории программ относится пакет;
- обеспечение возможности определения принадлежности файла или каталога к пакету;
- комплексная проверка состояния пакетов в системе: что изменялось, что испортилось, что случайно удалили и т. д.;
- отсутствие необходимости производить перезагрузку системы после инсталляции нового пакета. Пакет готов к эксплуатации сразу после установки.

Недостатки RPM

Пакет RPM имеет и недостатки:

- многие программы пакета обновляются позже, чем официально выходят версии программного обеспечения;
- отсутствие RPM для некоторых программ;
- централизованная база установленных пакетов.

Информация, содержащаяся в пакете

Каждый пакет RPM содержит в себе стандартный набор полей, которые характеризуют содержание пакета. Наиболее интересные для пользователя поля приведены ниже.

- Build Host — имя хоста, на котором производилась сборка пакета;
- Build Date — время сборки пакета;
- Change Log — краткий список изменений в программе, по сравнению с предыдущими версиями;
- Copyright — копирайт владельца;
- Description — описание пакета, обычно 1—2 Кбайт текста;
- Group — группа/подгруппа программного обеспечения, к которому относится пакет. К примеру — Development/Languages;
- License — лицензия, по которой распространяется пакет. Для большинства программ, поставляемых в дистрибутиве, лицензия — GPL. Для большинства библиотек — LGPL;
- Name — имя программы, к примеру apache;
- Version — версия программы;
- Release — релиз (версия сборки);
- RPM version — версия пакета RPM: для Red Hat Linux 7.*x* версия 4, для более ранних — версия 3;
- Size — размер в байтах;
- Source RPM — пакет с исходными кодами, на базе которого собирался бинарный пакет. Например: gcc-2.96-85.src.rpm;
- Summary — краткое, в одно-два предложения описание пакета. Например: The C Preprocessor;
- URL — Web-адрес разработчика программы;
- Vendor — сборщик пакета, например: Red Hat, Inc.

Категории пакетов

Для удобства пользователей пакеты содержат в себе признак, указывающий, к какой категории программного обеспечения относится пакет (поле Group). Стандартная иерархия пакетов приведена на рис. 8.1.

Ниже дана краткая расшифровка категорий пакетов.

- Amusements — развлечения. К этому разделу обычно относятся игры и всякие бесполезные, но веселые программки — глаза, которые следят за курсором, котенок, бегающий по экрану, и т. п.:
  - Games — подраздел предназначен для игр;
  - Graphics — всякие забавные графические программы, в том числе хранители экрана (screensavers).

Рис. 8.1. Стандартная иерархия пакетов

- Applications — приложения. Раздел предназначен для пользовательских (в широком смысле) программ. Как правило, сюда помещаются программы общего назначения: редакторы, инженерные пакеты, средства мультимедиа:
  - Archiving — подраздел, посвященный программам и утилитам архивации;
  - Communications — подраздел, содержащий все, что относится к связи. Здесь собраны разнообразные программы и утилиты для работы с модемами, факсами, ISDN, ATM, радиосвязью и многое другое;
  - Databases — подраздел, посвященный базам данных и разнообразным утилитам для взаимодействия с базами данных;
  - Editors — редакторы. В этом разделе хранятся разнообразные редакторы, от очень простых консольных редакторов до графических монстров;
  - Engineering — подраздел, посвященный инженерным пакетам: редакторы схем, формул, химических соединений, чертежные пакеты и тому подобные приложения;

- File — подраздел, содержащий утилиты для работы с файлами;
- Internet — программы, предназначенные для работы в Интернете: Web-браузеры, почтовые клиенты, клиенты ICQ и новостей, чатов и FTP;
- Multimedia — все для мультимедиа: проигрыватели CD, MP3-файлов, программы для просмотра телепередач и приема радиостанций, микшеры и т. д.;
- Productivity — подраздел для программ, позволяющих увеличить производительность труда: органайзеры, напоминалки, картотеки и т. п.;
- Publishing — подраздел для программ подготовки документов к печати: программы верстки, разметки и т. п.;
- System — подраздел для системных программ. Здесь могут быть программы, предназначенные только для администратора, и программы, интересные только для пользователя;
- Text — подраздел для программ и утилит работы с текстом: поиск слов и фраз, замены и т. п.

❒ Development — раздел, полностью посвященный программированию и программистам: отладчики, компиляторы, библиотеки разработчика, различные утилиты:

- Debuggers — подраздел для программ-отладчиков;
- Languages — подраздел, посвященный языкам программирования, компиляторам, интерпретаторам;
- Libraries — подраздел для библиотек: по большей части библиотеки разработчика, не системные;
- System — подраздел для системных утилит;
- Tools — подраздел для различного инструментария программиста, не попавшего в предыдущие подразделы.

❒ Documentation — раздел для документации, поставляемой отдельно от программ.

❒ System Environment — раздел системного окружения, наиболее ориентированный на ядро системы:

- Base — подраздел для базовых пакетов;
- Daemons — подраздел исключительно для демонов (daemon, демон — программа, выполняющая некоторые системные функции или являющаяся сервером каких-то услуг, сервисов);
- Kernel — подраздел, предназначенный исключительно для ядра Linux как в двоичном виде, так и в исходных кодах;

- Libraries — подраздел для системных библиотек;
- Shells — подраздел для хранения разнообразных командных оболочек.

❐ User Interface — раздел пользовательского интерфейса. Вернее было бы назвать его разделом, посвященным X Window:

- Desktops — подраздел, посвященный различным оконным менеджерам;
- X — пакеты, относящиеся к X Window;
- X Hardware Support — подраздел содержит пакеты, ориентированные на конкретный тип видеокарт.

Команды консольного менеджера RPM

Раздел полностью посвящен консольному менеджеру RPM. Понятно желание пользоваться графическими менеджерами пакетов — красиво, наглядно, удобно, просто, в конце концов. Но не следует забывать, всегда может случится так, что у вас не будет возможности загрузить X Window (например, необходимо установить новую версию X Window), да и возможностей у RPM побольше, а ресурсов он потребляет несравненно меньше. Тем более, что еще никто не отменял дистанционное администрирование, при котором вообще невозможно воспользоваться графическими пакетами. Раздел практически полностью основывается на содержимом man-страницы RPM.

Итак, использование RPM, Менеджера пакетов от Red Hat. Может быть выбран один из следующих основных режимов:

❐ инициализация базы данных;
❐ пересборка базы данных;
❐ сборка пакетов;
❐ рекомпиляция пакетов;
❐ сборка пакетов из tar-архивов;
❐ запрос;
❐ показ полей запроса;
❐ установка;
❐ обновление;
❐ удаление;
❐ верификация;
❐ проверка подписи;
❐ повторная подпись;
❐ добавление подписи;
❐ установка владельцев и групп;
❐ показ конфигурации.

Общие опции

Общие опции могут быть использованы во всех режимах работы:

❐ `-vv` — выводить много отладочной информации;

❐ `-quiet` — выводить как можно меньше сообщений: как правило, выводятся только сообщения об ошибках;

- `-help` — вывести более детальную, чем обычно, справку об использовании RPM;
- `-version` — вывести одну строку, содержащую номер версии используемого RPM;
- `-rcfile <список_файлов>` — каждый из файлов из разделенного двоеточиями `<списка_файлов>` последовательно читается RPM на предмет конфигурационной информации. По умолчанию `<список_файлов>` выглядит как /usr/lib/rpm/rpmrc:/etc/rpmrc:~/.rpmrc. В этом списке обязана существовать только первая строка; все тильды будут заменены значением `$HOME`;
- `-root <каталог>` — использовать для всех операций файловую систему с корнем в `<каталог>`. Обратите внимание, это значит, что база данных также будет читаться и модифицироваться под `<каталог>` и все pre- и post-скрипты будут исполняться после `chroot()` в `<каталог>`;
- `-dbpath <путь>` — использовать базу данных RPM в `<путь>`;
- `-justdb` — обновить только базу данных, не файловую систему;
- `-ftpproxy <host>` — использовать `<host>` как FTP-прокси *(см. разд. "Опции FTP/HTTP")*;
- `-httpproxy <host>` — использовать `<host>` как HTTP-прокси *(см. разд. "Опции FTP/HTTP")*;
- `-ftpport <порт>` — использовать `<порт>` как FTP-порт прокси-сервера *(см. разд. "Опции FTP/HTTP")*;
- `-httpport <порт>` — использовать `<порт>` как HTTP-порт прокси-сервера *(см. разд. "Опции FTP/HTTP")*;
- `-pipe <cmd>` — перенаправляет вывод RPM на вход команды `<cmd>`.

Опции установки и обновления

Общая форма команды установки новых RPM выглядит так:

```
rpm -i [опции-установки] <файл_пакета>
```

Общая форма команды обновления установленных RPM выглядит так:

```
rpm -U [опции-установки] <файл_пакета>
```

Команда обновления установленных пакетов полностью аналогична работе команды установки за исключением того, что если уже был установлен пакет, `rpm` проверяет версию установленного пакета и если она меньше версии нового пакета, происходит удаление установленного пакета и установка нового. Или более просто, если пакет не был установлен, эта команда произ-

водит установку, а если был установлен и имеет более раннюю версию, то происходит замена более ранней версии на новую.

```
rpm -F [опции-установки] <файл_пакета>
```

или

```
rpm -freshen [опции-установки] <файл_пакета>
```

Эта команда производит обновление пакетов, но только если в системе существуют более ранние версии этих пакетов.

Допускается задание `<файл_пакета>` в виде FTP- или HTTP-адресов (например, **http://www.freshmeat.net/Linux/ww-1.11-5.src.rpm**). В этом случае перед установкой пакет будет получен с сервера, указанного в адресе. Подробную информацию о встроенной поддержке FTP/HTTP см. в *разд. "Опции FTP/HTTP" данной главы.*

Опции:

- `-force` — то же, что и комбинация `-replacepkgs`, `-replace-ffiilleess` и `-oldpackage`. Принудительная установка пакета, невзирая на наличие неудовлетворенных зависимостей или уже установленных пакетов, имеющих более позднюю версию;
- `-h`, `-hash` — выводить 50 раз знак `#` по мере распаковки архива с пакетом. Используется с `-v` для придания читабельного вида. Можно использовать при автоматической установке пакетов, когда результат инсталляции выводится в журнальный (лог, log) файл;
- `-oldpackage` — позволяет заменить новый пакет на более старый при обновлении (откатиться назад). Как правило, необходимость отката (rollback) возникает в двух случаях: первый — при смене версий программного обеспечения (например, компилятор gcc поменял версию с 2.9*x* на 3.0), а новая версия имеет недостатки в функционировании (подвисает, исчезли необходимые вам свойства программы и т. д.). Второй — новая версия программного обеспечения конфликтует с уже установленными пакетами (не те версии библиотек, другой формат вызова модулей и т. п.);
- `-percent` — выводить процент готовности по мере распаковки архива с пакетом. Задумано для облегчения использования RPM из других утилит;
- `-replacefiles` — устанавливать пакеты, даже если они перепишут файлы из других, уже установленных пакетов;
- `-replacepkgs` — устанавливать пакеты, даже если некоторые из них уже установлены в системе;
- `-allfiles` — устанавливать или обновлять все файлы, определенные как missingok (согласно базе RPM — отсутствующие файлы в системе для данного пакета), даже если они уже существуют;

- `-nodeps` — не проверять зависимости перед установкой или обновлением пакета;
- `-noscripts` — не исполнять pre- и post-установочных скриптов;
- -notriggers — не исполнять триггер-скриптов, взведенных на установку данного пакета;
- `-ignoresize` — не проверять файловую систему на наличие достаточного свободного места перед установкой этого пакета;
- `-excludepath <путь>` — не устанавливать файлы, чьи имена начинаются с `<путь>`;
- `-excludedocs` — не устанавливать никаких файлов, отмеченных как файлы документации (включает man-документацию и документы texinfo);
- `-includedocs` — устанавливать файлы документации. Это поведение по умолчанию;
- `-test` — не устанавливать пакет, просто проверить возможность установки и сообщить о потенциальных проблемах;
- `-ignorearch` — произвести установку или обновление, даже если архитектуры бинарного RPM и машины не совпадают;
- `-ignoreos` — произвести установку или обновление, даже если операционные системы бинарного RPM и машины не совпадают;
- `-prefix <путь>` — установить префикс установки в `<путь>` для перемещаемых пакетов;
- `-relocate <старый_путь>=<новый_путь>` — для переместимых пакетов: преобразовывает в `<новый_путь>` файлы, которые должны были бы быть установлены в `<старый_путь>`;
- `-badreloc` — для использования вместе с `-relocate`. Производит перемещение, даже если пакет непереместимый;
- `-noorder` — не переупорядочивать список устанавливаемых пакетов. Обычно список переупорядочивается для удовлетворения зависимостей.

Опции удаления (деинсталляции)

Общая форма команды удаления пакета выглядит так:

```
rpm -e <название_пакета>
```

Опции:

- `-allmatches` — удалить все версии пакета, отвечающие `<название_пакета>`. Обычно если `<название_пакета>` отвечает нескольким пакетам, выдается сообщение об ошибке и удаление не производится;

- `-noscripts` — не исполнять pre- и post-установочные скрипты;
- `-notriggers` — не исполнять триггер-скриптов, взведенных на удаление данного пакета;
- `-nodeps` — не проверять зависимостей перед удалением пакетов;
- `-test` — не производить удаления, только протестировать возможность удаления. Полезна в сочетании с опцией `-vv`.

Опции запроса

Общая форма команды запроса RPM выглядит так:

```
rpm -q [опции-запроса]
```

Можно задать формат, в котором будет выводиться информация о пакете. Для этого используется опция `-queryformat` с последующей строкой формата. Форматы запроса представляют собой модифицированную версию стандартного форматирования `printf()`. Формат состоит из статических строк (которые могут включать стандартные escape-последовательности языка программирования C для переводов строки, табуляций и других специальных символов) и форматов по типу используемых в `printf()`.

Есть два набора опций для запроса — выбор пакетов и выбор информации.

Опции выбора пакетов

Запрос установленного пакета, называющегося <название_пакета>:

```
-q <название_пакета>
```

Опции:

- `-a`, `-all` — запрос всех установленных пакетов;
- `-whatrequires <capability>` — запрос всех пакетов, требующих `<capability>` для правильного функционирования;
- `-whatprovides <virtual>` — запрос всех пакетов, предоставляющих `<virtual>` сервис;
- `-f <файл>`, `-file <файл>` — запрос пакета, которому принадлежит файл `<файл>`;
- `-g <группа>`, `-group <группа>` — запрос пакетов из группы `<группа>`;
- `-p <файл_пакета>` — запрос (неустановленного) пакета `<файл_пакета>`. Файл `<файл_пакета>` может быть задан как FTP- или HTTP-адрес;
- `-specfile <spec_file>` — разбор и запрос `<spec_file>` так, как если бы это был пакет. Хотя не вся информация (например, списки файлов) доступна, этот тип запроса позволяет использовать RPM для извлечения информации из spec-файлов;

- `-querybynumber <num>` — запросить непосредственно запись базы данных номер `<num>`. Полезна для отладочных целей;
- `-triggeredby <имя_пакета>` — запрос всех пакетов, содержащих триггер-скрипты, активизируемые пакетом `<имя_пакета>`.

Опции выбора информации

Опции выбора информации выглядят так:

- `-i` — выводит информацию о пакете, включая название, версию и описание. Использует `-queryformat`, если таковой задан;
- `-R`, `-requires` — выводит список пакетов, от которых зависит данный пакет;
- `-provides` — выводит список сервисов и библиотек, предоставляемых данным пакетом;
- `-changelog` — выводит протокол изменений данного пакета;
- `-l`, `-list` — выводит список файлов, входящих в данный пакет;
- `-s`, `-state` — выводит состояние файлов в пакете (подразумевает `-l`). Каждый файл может находиться в одном из следующих состояний: нормальный, не установлен или заменен;
- `-d`, `-docfiles` — выводит список только файлов документации (подразумевает `-l`);
- `-c`, `-configfiles` — выводит список только конфигурационных файлов (подразумевает `-l`);
- `-scripts` — выводит специфические для данного пакета скрипты, используемые как часть процессов инсталляции/деинсталляции, если таковые есть;
- `-triggers`, `-triggerscripts` — показать все триггер-скрипты, если таковые имеются, содержащиеся в пакете;
- `-dump` — выводит информацию о файлах следующим образом: `path size mtime md5sum mode owner group isconfig isdoc rdev symlink`. Эта опция должна использоваться в сочетании по меньшей мере с одной из опций `-l`, `-c`, `-d`;
- `-last` — упорядочивает список пакетов по времени установки таким образом, что наиболее свежие пакеты находятся в верху списка;
- `-filesbypkg` — показывает все файлы в каждом пакете;
- `-triggerscripts` — показывает все триггер-скрипты для выбранных пакетов.

Опции проверки

Общая форма команды проверки RPM выглядит так:

```
rpm -V [опции-верификации]
```

или

```
rpm -y [опции-верификации]
```

или

```
rpm -verify [опции-верификации]
```

В процессе проверки пакета информация об установленных файлах пакета сравнивается с информацией из оригинального пакета и из базы данных RPM. В числе прочих верификация проверяет размер, контрольную сумму MD5, права доступа, тип, хозяина и группу каждого файла. Обо всех несоответствиях сообщается. Опции выбора пакетов такие же, как и для инспекции пакетов.

Файлы, которые не устанавливались из пакета (например, файлы документации, исключенные из процесса инсталляции при помощи опции `-excludedocs`) молча игнорируются.

Крайне полезная опция для администратора. Эта опция позволит при сбое в системе обнаружить поврежденные файлы (конечно, не все — конфигурационные файлы или файлы, созданные пользователем, так проверить не удастся). В случае взлома системы можно вычислить, какие файлы взломщик модифицировал (например, login).

Опции, которые могут быть использованы в процессе верификации:

- `-nofiles` — игнорировать отсутствующие файлы;
- `-nomd5` — игнорировать ошибки контрольной суммы MD5;
- `-nopgp` — игнорировать ошибки подписи PGP.

Форматом вывода является строка из восьми символов. Каждый из них показывает результат сравнения одного из атрибутов файла со значением, записанным в базе данных RPM. Точка обозначает, что тест прошел. Следующие символы говорят об ошибках некоторых тестов:

- 5 — контрольная сумма MD5;
- S — размер файла;
- L — ссылка (Симлинк);
- T — время модификации;
- D — устройство;
- U — владелец;
- G — группа;
- M — права доступа (включает права доступа и тип файла).

Проверка подписи

Общая форма команды проверки подписи RPM выглядит так:

```
rpm -checksig <файл_с_пакетом>
```

Эта команда проверяет встроенную в пакет PGP-подпись для подтверждения целостности и источника происхождения пакета. Информация о конфигурации PGP читается из конфигурационных файлов. Подробную информацию см. в *разд. "Подписи PGP"*.

Опции сборки пакетов

Общая форма команды построения пакета RPM выглядит так:

```
rpm -bO [опции-сборки] <spec_файл>
```

или

```
rpm -tO [опции-сборки] <arc_файл>
```

Аргумент `-b` применяется в том случае, если для сборки пакета используется spec-файл. Если же команда `rpm` должна извлечь этот файл из архива gzip, используется аргумент `-t`. После первого аргумента ставится следующий: `O`, указывающий, какие этапы сборки и упаковки должны быть выполнены. Это один из:

- `-bp` — исполнить стадию `%prep` spec-файла. Обычно это включает в себя распаковку исходного кода и прикладывание к нему патчей (от англ. *patch* — патч, заплатка, исправление);
- `-bl` — произвести проверку списка. В секции `%files` spec-файла производится расширение макросов и проверка перечисленных файлов на существование;
- `-bc` — исполнить стадию `%build` spec-файла (предварительно исполнив стадию `%prep`). Обычно это сводится к исполнению некого эквивалента `make`;
- `-bi` — исполнить стадию `%install` spec-файла (предварительно исполнив стадии `%prep` и `%build`). Обычно это сводится к исполнению некого эквивалента `make install`;
- `-bb` — собрать бинарный пакет (предварительно исполнив стадии `%prep`, `%build` и `%install`);
- `-bs` — собрать только исходный пакет (предварительно исполнив стадии `%prep`, `%build` и `%install`);
- `-ba` — собрать бинарный (RPM) и исходный (SRPM) пакеты (предварительно исполнив стадии `%prep`, `%build` и `%install`).

Также могут быть использованы следующие опции:

- `-short-circuit` — исполнить непосредственно указанную стадию, пропустив предшествующие. Может быть использована только с `-bc` и `-bi`;
- `-timecheck` — установить возраст для `timecheck` (0 — чтобы запретить). Это значение также может быть установлено путем определения макроса `_timecheck`. Значение `timecheck` определяет максимальный возраст (в секундах) пакуемых в пакет файлов. Для всех файлов, которые старше этого возраста, будет выводиться предупреждение;
- `-clean` — удалить дерево, использованное для сборки, после того, как построены пакеты;
- `-rmsource` — удалить исходный код и spec-файл после сборки (может быть использовано отдельно, например, `rpm -rmsource foo.spec`);
- `-test` — не исполнять никаких стадий сборки. Полезно для тестирования spec-файлов;
- `-sign` — встроить в пакет PGP-подпись. Эта подпись может быть использована для проверки целостности и источника происхождения пакета. Подробную информацию см. в *разд. "Подписи PGP"*;
- `-builroot <каталог>` — использовать каталог `<каталог>` как корневой для сборки пакетов;
- `-target <платформа>` — при сборке пакета интерпретировать `<платформа>` как `arch-vendor-os` и соответственно установить макросы `_target`, `_target_arch` и `_target_os`.

Опции пересборки и перекомпиляции

Существуют два способа запуска RPM:

- `rpm -recompile <файл_исходного_пакета>`
- `rpm -rebuild <файл_исходного_пакета>`

Будучи вызванным любым из способов, RPM устанавливает указанный исходный пакет и исполняет стадии `%prep`, `%build` и `%install`. Кроме того, `-rebuild` собирает новый бинарный пакет. После того как сборка закончена, удаляется дерево, использованное для сборки (как с опцией `-clean`), исходный код и spec-файл.

Подпись существующего RPM

Подпись RPM выполняется следующими командами:

- `rpm -resign <файл_бинарного_пакета>`

  Опция `resign` генерирует и вставляет новые подписи в указанные пакеты. Все существующие подписи из пакетов удаляются.

- `rpm -addsign <файл_бинарного_пакета>`

  Опция `addsign` генерирует и добавляет новые подписи в указанные пакеты. Все существующие подписи пакетов при этом сохраняются.

Подписи PGP

Чтобы использовать возможность подписи, RPM должен быть настроен для запуска PGP. Для этого следует создать свою собственную пару из публичного и секретного ключей. Необходимо также настроить следующие макросы:

- `_signature` — тип подписи. В настоящее время поддерживается только pgp;
- `_pgp_name` — имя "пользователя", чьи ключи вы хотите использовать для подписи ваших пакетов.

При сборке пакетов к командной строке добавляется опция `-sign`. У вас спросят пароль, и ваш пакет будет собран и подписан.

Опции пересборки базы данных

Общая форма команды перестроения базы данных RPM выглядит так:

```
rpm -rebuilddb
```

Для построения новой базы данных:

```
rpm -initdb
```

Этот режим поддерживает только две опции: `-dbpath` и `-root`.

Опции FTP/HTTP

RPM содержит простые клиенты FTP и HTTP для упрощения установки и изучения пакетов, доступных через Интернет. Файлы пакетов для установки, обновления и запроса могут быть указаны как FTP- или HTTP-адрес:

```
ftp://<user>:<password>@hostname:<port>/path/to/packagee.rpm
```

Если часть `<password>` опущена, пароль будет запрошен (по одному разу для каждой пары user/hostname). Если ни `<user>`, ни `<password>` не указаны, будет использован anonymous ftp. Во всех случаях осуществляется пассивная (PASV) пересылка по FTP.

RPM позволяет применять с адресом FTP следующие опции:

- `-ftpproxy <hostname>` — система `<hostname>` будет организована как proxy-сервер для всех пересылок, что позволяет производить FTP-соединения через firewall, использующий прокси для выхода во внешний мир. Эта опция может быть задана также настройкой макроса `_ftpproxy`;

- `-ftpport <port>` — задает номер TCP-порта, открываемого для FTP-соединений вместо порта по умолчанию. Эта опция может быть также задана настройкой макроса `_ftpport`.

RPM позволяет применять с адресом HTTP следующие опции:

- `-httpproxy <hostname>` — система `<hostname>` будет организована как proxy-сервер для всех пересылок, что позволяет производить HTTP-соединения через firewall, использующий прокси для выхода во внешний мир. Эта опция может быть задана также настройкой макроса `_httpproxy`;
- `-httpport <port>` — задает номер TCP-порта, открываемого для HTTP-соединений вместо порта по умолчанию. Эта опция может быть также задана настройкой макроса `_httpport`.

Используемые файлы

Следующие файлы необходимы при работе с пакетом RPM:

- /usr/lib/rpm/rpmrc;
- /etc/rpmrc;
- ~/.rpmrc;
- /var/state/rpm/packages;
- /var/state/rpm/pathidx;
- /var/state/rpm/nameidx;
- /tmp/rpm*.

Примеры использования консольного менеджера пакетов RPM

В предыдущем разделе мы познакомились с опциями менеджера RPM. С легкостью установки программ в Windows не сравнить. Впрочем, пользователи вряд ли применяют даже десятую часть имеющихся опций, поэтому и не следует запоминать их все. Рассмотрим, что практически используется в работе с пакетами.

Установка пакетов осуществляется с помощью команды:

```
rpm -i <полное_имя_пакета>
```

или

```
rpm -i <полное_имя_пакета> <полное_имя_пакета> <полное_имя_пакета> …
```

Например: `rpm -i cpp-2.96-85.i386.rpm`

Таким образом можно установить сразу несколько пакетов. Помимо удобства (сразу указывается список пакетов, и они устанавливаются сами) указа-

ние нескольких пакетов необходимо в том случае, если возникают неудовлетворенные зависимости. Попадаются пакеты, зависящие друг от друга. Без второго пакета не установить первый, а второй не устанавливается — требует установки первого. Простейшее решение — поставить пакеты командой:

```
rpm -i <полное_имя_пакета_1> <полное_имя_пакета_2>
```

Команда простая, работает хорошо, но если в системе уже установлен пакет, пусть и более ранней версии, вы получите предупреждение, а сам пакет не установится. Чтобы обновить пакет, используем следующую команду:

```
rpm -U <полное_имя_пакета>
```

Эта команда производит обновление пакета, если он уже установлен, или установку, если не установлен. Однако не всегда при обновлении необходимо устанавливать отсутствующий пакет. В таких случаях можно воспользоваться командой:

```
rpm -F <полное_имя_пакета>
```

Команда проверит, есть ли в системе соответствующий пакет, и если есть — произведет его обновление. При желании установку или обновление пакетов можно производить прямо с сервера FTP. Например, в локальной сети есть FTP-сервер с именем bluewater. Ваш администратор регулярно скачивает с FTP-сервера Red Hat обновления RPM и выкладывает их на FTP-сервер локальной сети. Вот команда, с помощью которой можно обновить свои пакеты (для определенности возьмем компилятор C++):

```
rpm -F ftp://bluewater/pub/linux/updates/redhat-7.1/cpp-2.96-85.i386.rpm
```

Однако у консольного менеджера RPM есть одна неприятная особенность — в случае успешности операции он ничего не сообщает на консоль. В принципе, это нестрашно, можно задать ключ `-h`, который выводит процент выполнения процедуры. Например:

```
rpm -ih cpp-2.96-85.i386.rpm
```

или

```
rpm -ivh cpp-2.96-85.i386.rpm
```

Если при работе с пакетом возникнут проблемы, RPM выдаст причину, по которой невозможно выполнить какое-то действие. При установке пакета — это, как правило, уже упомянутые неудовлетворенные зависимости либо отсутствие необходимых библиотек или установленных пакетов (или они в системе есть, но не той версии). Эти проблемы решаются просто — установите соответствующие пакеты или обновите их до необходимой версии. Впрочем, бывают и здесь свои трудности. Рассмотрим еще один пример. У автора на компьютере стоит Red Hat 7.1, а в нем удобная система GNOME, использующая менеджер окон Sawfish. Все хорошо функционирует, только есть одна проблема — при сборке пакета Sawfish сборщики

(американцы, им простительно) напутали с кириллическими шрифтами — системное меню вместо кириллицы отображает знаки вопроса. Был найден пакет посвежее, в котором эта оплошность убрана, да еще и функциональности добавлено. Пакет содержал исходные коды, поэтому пришлось сначала собрать его в бинарном виде командой:

```
rpm -rebuild Sawfish-1.0-1.src.rpm
```

После приблизительно пяти минут компиляции в каталоге /usr/src/redhat/RPMS/i386/ образовался пакет Sawfish, который был запущен на обновление командой:

```
rpm -F Sawfish
```

А в результате получено сообщение: "Обновление пакета не произведено, поскольку в системе уже установлен пакет версии 0.36, которая больше, чем версия 1.02". По всей видимости, сборщики пакета что-то перепутали в его описании. Пришлось воспользоваться командой:

```
rpm -U -force Sawfish-1.0-1.i386.rpm
```

которая производит принудительное обновление пакета, не проверяя зависимостей. Ключом `-force`, однако, следует пользоваться достаточно осторожно, можно ненароком развалить всю систему.

Встречаются иногда сообщения и другого рода. При обновлении, например, пакета с исходным кодом ядра Linux версии 2.4.2 пакетом, содержащим исходный код ядра Linux версии 2.4.3, было выдано сообщение, которое в переводе на русский язык звучит так: "Не могу удалить каталог такой-то, потому что он не пуст". Однако пакет успешно обновился, а каталог, фигурирующий в сообщении, на самом деле был пуст. Так что не стоит сразу расстраиваться, достаточно часто сообщения, выдаваемые RPM, весьма безобидны.

Удаление пакетов из системы осуществляется элементарно, с помощью команды:

```
rpm -e <имя_пакета>
```

Обратите внимание — указывается только имя пакета. Если написать полное имя пакета, то RPM выдаст сообщение: "Такой пакет в системе не установлен". Немного нелогично, но так уж исторически сложилось: при установке необходимо указывать полное имя пакета, при удалении — только имя пакета без упоминания версии, релиза и т. п.

При удалении сперва проверяются зависимости, и пакет удаляется, если от него не зависит никакой другой установленный в системе пакет. В противном случае на экран выдаются имена пакетов, для функционирования которых нужен удаляемый пакет. Конечно, если вы все-таки решили удалить пакет, можно воспользоваться ключами `-nodeps` или `-force`, однако рекомендуется применять их с большой осторожностью.

Получить информацию о том, какая версия пакета установлена в системе, можно командой:

```
rpm -q <имя_пакета>
```

Например, на запрос `rpm -q cpp` может быть получен такой ответ: `cpp-2.96-85`.

Для получения расширенной информации о пакете необходимо выполнить команду:

```
rpm -qi <полное_имя_пакета>
```

Результат выполнения команды `rpm -qi cpp-2.96-85` на конкретном компьютере можно видеть ниже:

```
Name    : cpp            Relocations: (not relocateable)
Version   : 2.96            Vendor: Red Hat, Inc.
Release   : 85             Build Date: Срд 09 Май 2001 21:04:50
Install date: Птн 31 Авг 2001 07:38:10  Build Host:
porky.devel.redhat.com
Group    : Development/Languages   Source RPM: gcc-2.96-85.src.rpm
Size    : 292618           License: GPL
Packager  : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL     : http://gcc.gnu.org
Summary   : The C Preprocessor.
Description :
Cpp is the GNU C-Compatible Compiler Preprocessor. Cpp is a macro proces-
sor which is used automatically by the C compiler to transform your pro-
gram before actual compilation. It is called a macro processor because it
allows you to define macros, abbreviations for longer constructs.

The C preprocessor provides four separate functionalities: the inclusion
of header files (files of declarations that can be substituted into your
program); macro expansion (you can define macros, and the C preprocessor
will replace the macros with their definitions throughout the program);
conditional compilation (using special preprocessing directives, you can
include or exclude parts of the program according to various conditions);
and line control (if you use a program to combine or rearrange source
files into an intermediate file which is then compiled, you can use line
control to inform the compiler about where each source line originated).

You should install this package if you are a C programmer and you use
macros.
```

Для получения списка файлов пакета (и каталогов, куда они будут установлены) используется команда:

```
rpm -ql <полное_имя_пакета>
```

Например, `rpm -ql cpp-2.96-85` выведет на экран следующий список файлов:

```
/lib/cpp
/usr/bin/cpp
```

```
/usr/lib/gcc-lib
/usr/lib/gcc-lib/i386-redhat-linux
/usr/lib/gcc-lib/i386-redhat-linux/2.96
/usr/lib/gcc-lib/i386-redhat-linux/2.96/cpp0
/usr/lib/gcc-lib/i386-redhat-linux/2.96/tradcpp0
/usr/share/info/cpp.info-1.gz
/usr/share/info/cpp.info-2.gz
/usr/share/info/cpp.info-3.gz
/usr/share/info/cpp.info.gz
/usr/share/man/man1/cpp.1.gz
```

А если надо произвести обратную операцию — по имени файла узнать, к какому пакету он принадлежит? Выполним следующую команду:

```
rpm -qf /usr/bin/mc
```

В результате получим имя пакета: `mc-4.5.51-32`.

Теперь о безопасности. Прежде чем производить установку пакета, полученного через Интернет, крайне желательно произвести его проверку, вдруг он поврежден?

Для этого можно воспользоваться командой

```
rpm -checksig <имя_пакета>
```

Эта команда проверяет PGP-подпись пакета.

Если ваша система — сервер или к компьютеру имеет доступ кто-то, в чьих действиях вы не уверены, необходимо регулярно производить проверку целостности установленных пакетов и зависимостей командой:

```
rpm -V gimp
```

В ответ можно получить, например, следующее:

```
.M......  /usr/lib/gimp/1.2/modules/libcolorsel_gtk.a
.M......  /usr/lib/gimp/1.2/modules/libcolorsel_triangle.a
.M......  /usr/lib/gimp/1.2/modules/libcolorsel_water.a
```

Результат говорит, что права доступа на эти файлы были модифицированы.

Для проверки всех установленных в системе пакетов можно воспользоваться командой:

```
rpm -Va
```

Вот результат:

```
S.5....T c /etc/printcap
.M......  /var/spool/at/.SEQ
```

отсутствует /etc/rpm/macros.db1

```
.......T  /usr/share/pixmaps/netscape.png
SM5....T  /usr/X11R6/lib/X11/fonts/Speedo/encodings.dir
```

отсутствует /var/cache/ssl_gcache_data.dir

```
.M....G.  /dev/jsfd
......G.  /dev/tty0
.....U..  /dev/vcs3
.....U..  /dev/vcsa3
S.5....T c /etc/X11/fs/config
```

отсутствует /usr/share/ssl/certs/stunnel.pem

```
S.5....T c /etc/openldap/ldap.conf
```

Совет

Если вы применяете дистрибутив, использующий пакеты RPM, избегайте установки программ компиляцией из исходного кода (не из пакетов RPM). Поскольку программа компилируется и устанавливается вручную, информация в базу данных установленных RPM не попадает. Следовательно, достаточно велика вероятность, что при установке или обновлении какого-нибудь пакета вы нарушите зависимости для скомпилированной вами программы, и она не будет работать.

Помимо консольного менеджера RPM, существуют еще несколько утилит, предоставляющих текстовый интерфейс и позволяющих работать с пакетами формата RPM. Однако они имеют обычно значительно меньшую функциональность.

Midnight Commander

Midnight Commander — помимо функций файлового менеджера, работы с архивами и большого количества других возможностей, Midnight Commander способен получить информацию из пакетов форматов RPM и DEB, установить или обновить пакет. Конечно, это не заменит полноценного менеджера пакетов, но быстро поставить или обновить несколько пакетов или посмотреть информацию о пакете также иногда бывает полезно. На рис. 8.2 изображено содержимое RPM-пакета, надо только нажать клавишу <Enter> в нужном месте.

Для нас интересны виртуальные файлы и каталоги (они все пишутся большими буквами):

- HEADER — содержит заголовок пакета — то, что можно получить командой `rpm -qi <имя_пакета>`;

- *INSTALL, *UPGRADE — если запустить на выполнение, Midnight Commander проинсталлирует или обновит этот пакет;
- /INFO — каталог с информацией о пакете. Содержит виртуальные файлы с информацией, описывающей пакет.

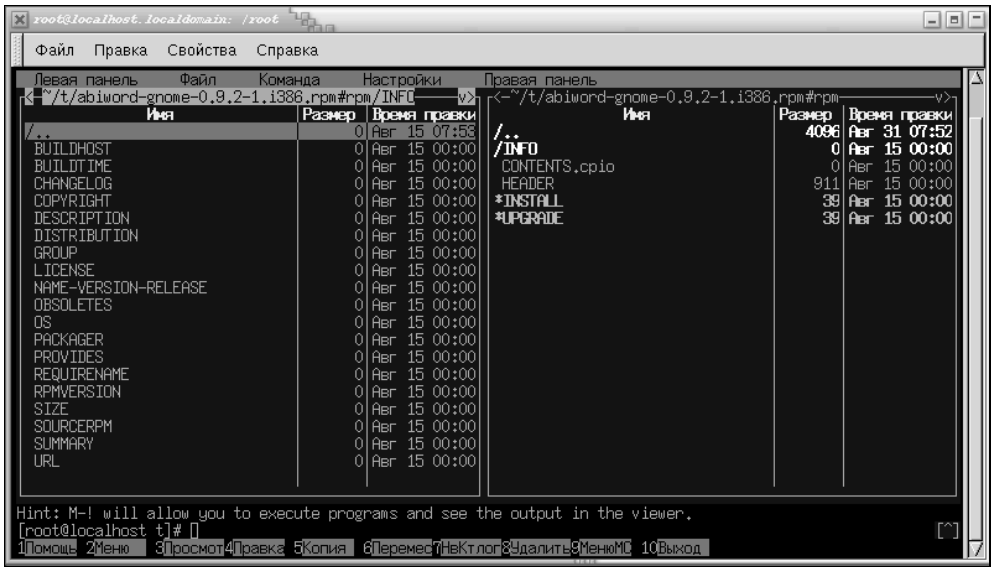

Рис. 8.2. Midnight Commander, работа с пакетами RPM

purp

Программа purp удобна для просмотра установленных пакетов, получения разнообразной информации, установки, удаления пакетов. Весьма полезная программа, по функциональности близка к RPM. На рис. 8.3 показано основное окно программы.

Предназначена для тех, кто не хочет/не может работать в X Window, а пользоваться RPM по каким-то причинам не желает (типичный представитель — бывший пользователь DOS/Windows, для которого привычен и удобен Norton Commander). На рис. 8.4 показан процесс установки пакетов.

Помимо текстовых менеджеров пакетов RPM, существует несколько графических менеджеров.

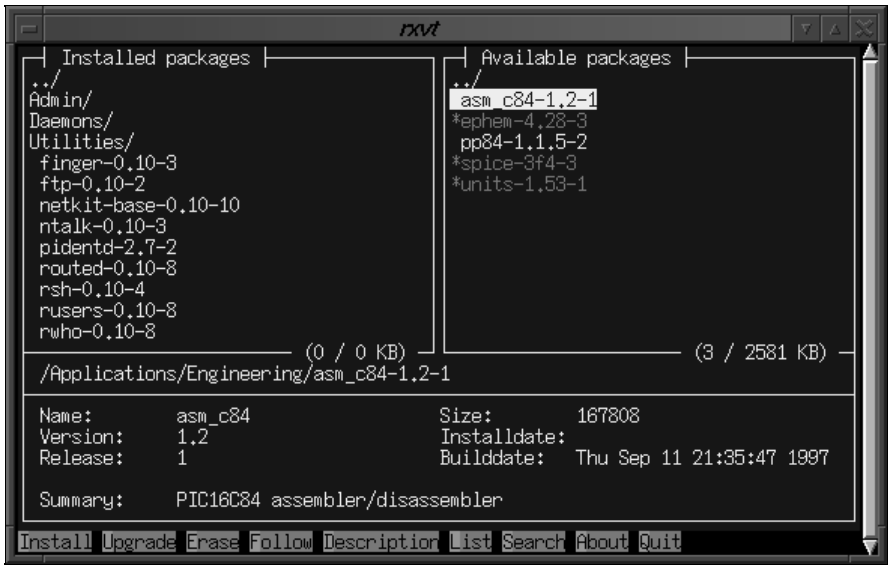

Рис. 8.3. Основное окно purp

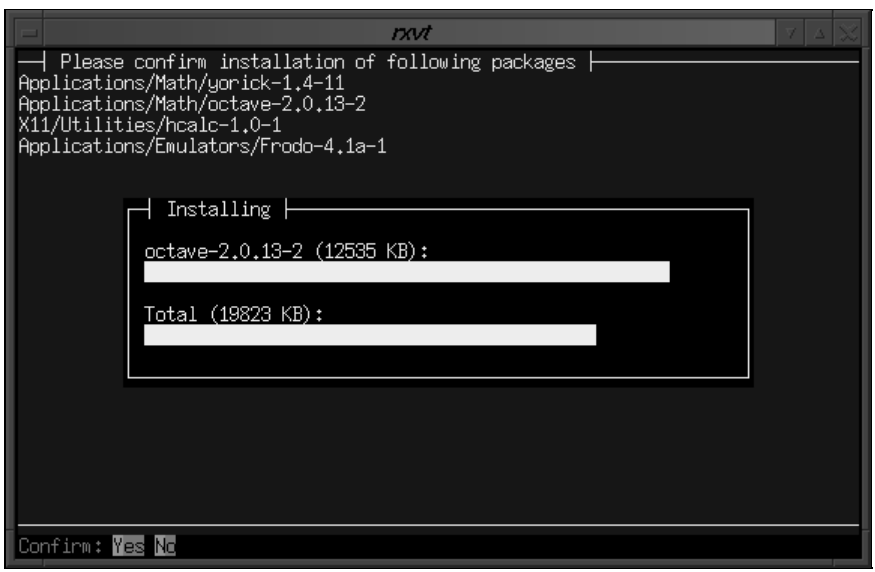

Рис. 8.4. Установка пакетов

Kpackage

Kpackage — это полнофункциональный графический интерфейс для менеджеров пакетов RPM, Debian, Slackware, BSD и KISS. Kpackage является частью рабочей среды K Desktop Environment и тесно интегрирован с файл-менеджером KDE (KFM). Практически все, что можно делать в консольном менеджере RPM, реализовано в Kpackage. Окно менеджера пакетов Kpackage приведено на рис. 8.5.

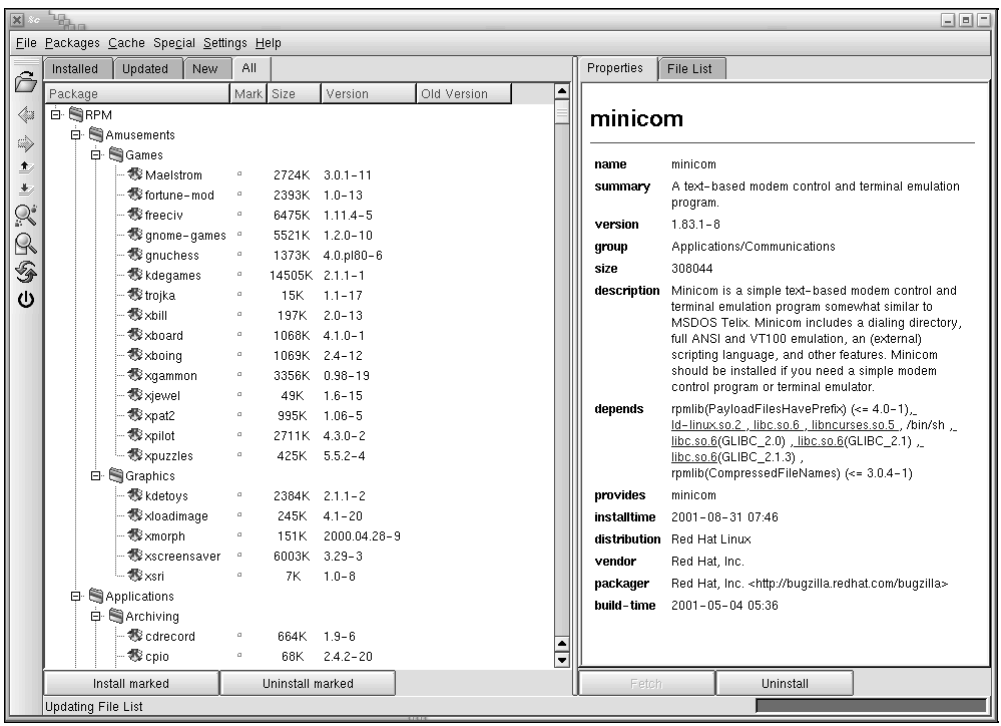

Рис. 8.5. Менеджер пакетов Kpackage

GnoRPM

Менеджер пакетов, входящий в состав GNOME. Полнофункциональный, в целом достаточно удобный. Однако есть несколько неприятных моментов:

- при установке пакетов необходимо отметить соответствующие пакеты. Однако после установки отметки автоматически не убираются;

- если при установке обнаружены неудовлетворенные зависимости, то менеджер не предлагает их автоматического удовлетворения.

Окно менеджера пакетов GnoRPM приведено на рис. 8.6.

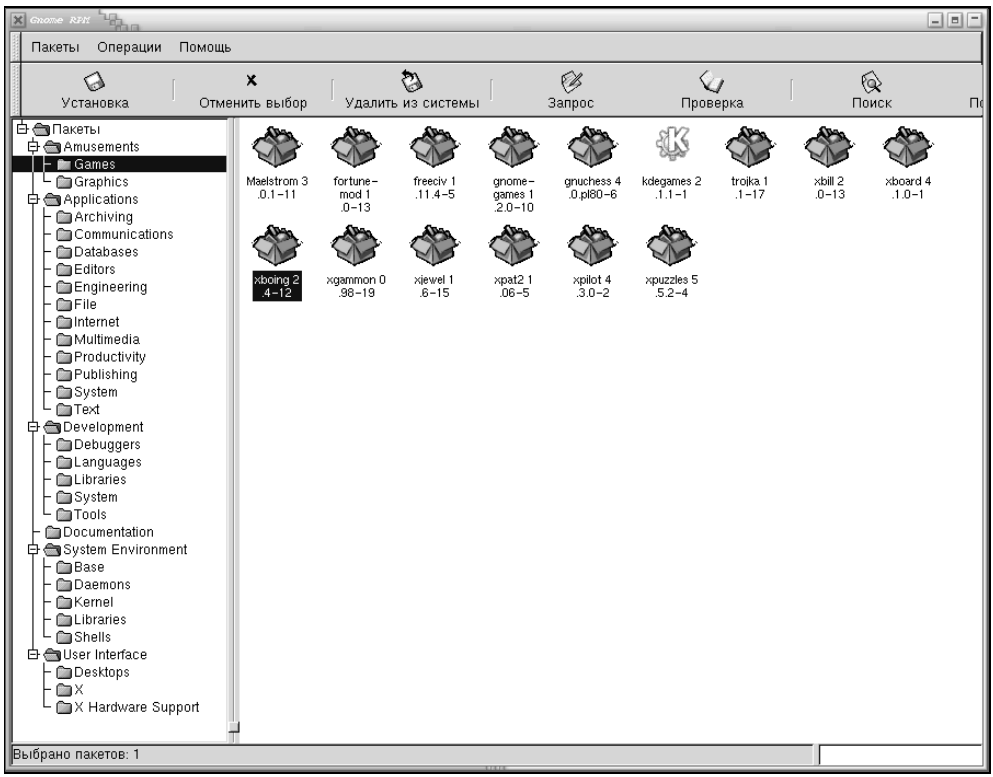

Рис. 8.6. Менеджер пакетов GnoRPM

Существуют также менеджеры glint, grpm, gtkrpm и много других. Однако наиболее распространенные — это Kpackage и GnoRPM. Они являются составной частью KDE и GNOME.

Ссылки

- **www.linuxdocs.org** — одно из собраний документации о Linux.
- **www.rpm.org/maximum-rpm.ps.gz** — источник сведений о RPM: "Maximum RPM" в формате PostScript.

- **www.redhat.com/support/docs/rpm/RPM-HOWTO/RPM-HOWTO.html** — RPM-HOWTO — описание RPM, тонкости работы (на английском языке).
- **www.linux.org.ru** — один из основных русскоязычных сайтов, посвященных Linux, в разделе документации есть RPM-HOWTO на русском языке.
- **www.rpm.org** — сайт, полностью посвященный RPM.
- **rpmfind.net** — репозитарий и поисковая система RPM.
- **rufus.w3.org/linux/RPM** — репозитарий RPM.
- **www.freshmeat.net** — большая коллекция программ, в том числе и в RPM-пакетах.

Часть III

Инсталляция Linux

Глава 9

Подготовка к инсталляции

Рассмотрим процесс подготовки к установке операционной системы Linux. Пользователи Windows, особенно если знакомство с ней произошло во времена третьей версии, знают, что инсталляция операционной системы сопровождается значительными усилиями по установке оборудования и периферии. С Linux (если вы специально не ищете трудностей) такого, скорее всего, не произойдет — вы поставите систему и все. И при выходе следующей версии дистрибутива ничего не придется переустанавливать заново. Даже если кардинально поменять всю аппаратуру (кроме винчестера), в большинстве случаев Linux сама определит новое оборудование и перенастроит систему. Вот конкретный пример. Было: винчестер 10 Гбайт, материнская плата на чипсете BX, видеокарта nVIDIA Vanta, сетевая карта на шине PCI и аудиокарта на шине ISA. После апгрейда системы от старой конфигурации остался только жесткий диск. Новая конфигурация: материнская плата на чипсете i815E — встроенные сетевой адаптер, видео и аудио. Windows 98, находящуюся на том же жестком диске, пришлось перенастраивать около часа. Linux сразу нашла все новое оборудование, оставила корректные сетевые настройки. Единственное, что пришлось сделать — заново настроить X Window: с помощью X-конфигуратора для X Window был получен новый конфигурационный файл. Весь переход на новую платформу занял 3 минуты.

Впрочем, если вы не экспериментатор — кардинально менять систему часто не придется. Существуют серверы, замена операционной системы Linux на которых не производится годами. Администратор к ним подходит раз в два месяца, чтобы сделать профилактику системного блока (пыль и т. п.). На этих машинах лишь периодически обновлялись некоторые прикладные пакеты: одни из-за проблем безопасности (ошибки есть в любой программе), другие — ставились более свежие версии. Опытные пользователи Windows 9*x* (особенно те, кто много и часто ставят разнообразное программное обеспечение) знают — систему надо периодически переустанавливать. С операционной системой Linux все несколько иначе. Во-первых, ее крайне тяжело штатными способами довести до необходимости переуста-

новки. Во-вторых, и это особенность любого программного обеспечения, как правило, новые версии программ весьма сырые. К примеру, очень тяжело дался переход с версии ядра 2.0 на 2.2. Были времена, когда исправления к ядру выпускались буквально ежедневно. Мы уже упоминали хорошее правило: "Работает — не трогай". Поэтому нормальные администраторы и пользователи выдерживают некоторую паузу после выхода очередного обновления, изучают отзывы, и только после этого устанавливают обновление на систему. В-третьих, для перехода с одной версии дистрибутива на другую иногда приходится выводить систему из "общего пользования" на день-два, а то и больше. Вот, собственно, почему, не стоит без особых причин менять одну версию дистрибутива на другую.

Дистрибутивы

Дистрибутивы. Что это такое? Какие они бывают? Чем один дистрибутив лучше другого?

Дистрибутив — это определенный набор программ, утилит и документации, объединенный логичной системой установки и сопровождения программных пакетов, ориентированный на определенную группу пользователей и определенный тип задач. По большому счету, обладая достаточными знаниями, можно накачать из Интернета ядро операционной системы, загрузчик, драйверы, программное обеспечение, и все это установить вручную, а потом долго подгонять и настраивать. Но в следующий раз, когда возникнет необходимость установить систему у другого пользователя, вы дважды подумаете — ставить все это самостоятельно и повторять мучения с настройкой или взять какой-либо дистрибутив и за полчаса установить систему (о настройке мы пока деликатно умолчим, случаи бывают разные).

О пользователях. Условно их можно разделить на начинающих, "продвинутых" и специалистов. Соответственно было бы неплохо иметь для каждой группы свой тип дистрибутива. И дистрибутивы, действительно, в некоторой степени ориентируются на такое разделение пользователей. Есть пакеты, где инсталляция проходит буквально за десять щелчков мышью, а существуют и такие, где очень многое необходимо настраивать вручную. Часть дистрибутивов пытается совместить в себе и легкость в установке, и настраиваемость всего и вся. Кстати, как уже неоднократно замечено, с переходом пользователей из одной группы в другую тяга к тотальной настраиваемости системы возрастает.

Помимо деления по легкости установки и сопровождения, дистрибутивы подразделяются и по назначению. Обычно это: офисный (домашний) дистрибутив, малый сервер, мощный сервер, и, конечно, дистрибутивы специального назначения. Многие дистрибутивы являются многофункциональными.

Попробуем теперь определить дистрибутив, приемлемый для большинства пользователей. Такой подход весьма субъективен, но все же некоторые тезисы для большинства дистрибутивов могут быть общими:

- набор пакетов должен быть логичен и удобен;
- основную часть дистрибутива должны составлять стабильные пакеты, все остальные пакеты должны отражаться в документации как экспериментальные, желательно так же для увеличения стабильности эти пакеты пропатчить (внести в них имеющиеся исправления, патчи, от англ. *patch* — исправление, заплатка);
- компакт-диск должен быть загрузочным и иметь опцию аварийной загрузки;
- дистрибутив должен иметь возможность устанавливаться с компакт-диска, с жесткого диска, по сети;
- набор программного обеспечения в пакете должен соответствовать целям дистрибутива;
- все программное обеспечение должно надежно функционировать на любом оборудовании, выпущенном в пределах 2—3 ближайших лет;
- локализация и интернационализация должны присутствовать во всех программах;
- при установке и настройке пакета администратор должен получить полный контроль над системой;
- все необходимое для инсталляции системы должно находиться на одном компакт-диске, все остальное — на дополнительных (опциональных) дисках.

В свете этих требований более пристально взглянем на проблему локализации и интернационализации. Чисто исторически сложилось, что языком международного общения является английский. Поэтому большинство дистрибутивов "говорят" на хорошем английском языке. Существуют "французский" и "немецкий" дистрибутивы, однако "нормального" русскоязычного дистрибутива до последнего времени не было.

По нашему мнению, в нормально локализованном дистрибутиве должны быть выполнены следующие требования:

- поддержка в интерфейсе всех распространенных языков;
- поддержка ввода и вывода символов национальных алфавитов как в текстовом, так и в графическом режиме во всех официальных кодировках или, по крайней мере, в наиболее распространенных;
- наличие локализованных версий всех сопутствующих программ, документации, проверки орфографии и т. п.;
- толковое, достаточно обширное локализованное руководство пользователя.

На сегодняшний день существуют три базовых дистрибутива и множество их потомков, причем некоторые из них уже имеют крайне мало общего с родителями.

Вот эти три дистрибутива — Debian, Read Hat, Slackware.

Группа Debian

В этой группе представлены дистрибутивы, исторически и идеологически родственные дистрибутиву Debian:

- Debian — фирма-разработчик: Debian. Web-сайт: **www.debian.org**. Весьма не плохой дистрибутив;
- StormLinux — фирма-разработчик: Debian. Web-сайт: **www.stormlinux.com**. Относительно маленький, пригодный для не очень мощных машин;
- Corel Linux — фирма-разработчик: Corel. Web-сайт: **www.corel.com**. Достаточно удачная попытка хорошего программостроителя создать офисный дистрибутив. Существуют некоторые недочеты, имеются проблемы с русским языком.

Группа Red Hat

В этой группе представлены дистрибутивы, исторически и идеологически родственные дистрибутиву Red Hat:

- Red Hat — фирма-разработчик: Red Hat. Web-сайт: **www.redhat.com**. На сегодня — самый популярный дистрибутив. Компания Red Hat предлагает несколько вариантов поставки. Достаточно приемлемая русификация, неплохая поддержка;
- KSI — фирма-разработчик: KSI Linux Company (Сергей Кубушин). Web-сайт: **www.ksi-linux.com**. Базируется на Red Hat. Дистрибутив, сделанный администратором-профессионалом для профессионального же использования — после инсталляции получается очень защищенная система. В настоящее время поддержка прекращена;
- Black Cat — разработчики: Леонид Кантер и Александр Каневский. Web-сайт: **www.blackcatlinux.com**. Переработка дистрибутива Red Hat, выполненная донбасской группой пользователей Linux. Очень качественная локализация. В настоящее время поддержка прекращена;
- ASP Linux (ASP, Advanced Server Platform) — фирма-разработчик: SWsoft. Web-сайт: **www.asplinux.ru**. Практически первый коробочный российский дистрибутив, отличается легкостью установки и настройки;
- Mandrake — фирма-разработчик: MandrakeSoft. Web-сайт: **www.linuxmandrake.com/ru**. Российская фирма IPLabs производит русифицированную версию — Mandrake RE. Простая инсталляция. Неплохая русификация. В принципе — тот же Red Hat;

- Caldera OpenLinux — фирма-разработчик: Caldera. Web-сайт: **www.caldera.com**. Коммерческий дистрибутив. Имеются проблемы с русским языком;
- BestLinux — фирма-разработчик: SOT Finish Software Engineering. Web-сайты: **www.bestlinux.net/ru**, **www.bestlinux.net**. Хорошая поддержка русского языка и удобный графический инсталлятор.
- TurboLinux — фирма-разработчик: TurboLinux Inc. Web-сайт: **www.turbolinux.com**. Средний дистрибутив, ничем особо не отмечен.

Группа Slackware

В этой группе представлены дистрибутивы, исторически и идеологически родственные дистрибутиву Slackware:

- Slackware — производство: Patrick Volkerding, Walnut Creek CDROM. Web-сайт: **www.slackware.com**. Один из старейших дистрибутивов. Сегодня мало распространен в связи с тем, что представляет собой конструктор для опытного пользователя Linux. В результате — настраивается все и вся, ставится только то, что указано (можно получить очень компактную систему);
- SuSE Linux — фирма-разработчик: SuSE. Web-сайт: **www.suse.de**. Дистрибутив чрезвычайно популярен в Германии. Основное его преимущество — огромное количество включенных в дистрибутив программ.

Это далеко не полный список дистрибутивов. Выбор весьма внушителен, и он за вами. Конечно, у автора есть свои предпочтения — Red Hat Linux. Многие с этим не согласятся, и это их право. Как написано в FAQ по Linux — дистрибутив надо выбирать тот, с которым работает ваш знакомый специалист. Намного проще решить любой вопрос с помощью специалиста, чем просматривать горы литературы или рыться в Интернете (хотя использование книг и Интернета тоже никто не отменял). В плане простоты и удобства инсталляции и обновления системы — можно посоветовать дистрибутивы, основанные на Red Hat. Локализация (русификация) достаточно хорошо сделана у российских и украинских дистрибутивов.

Далее мы будем рассматривать дистрибутив Red Hat Linux 7.1, поскольку он очень популярен как у нас, так и за рубежом.

Перед инсталляцией

В первую очередь — необходимо где-то взять сам дистрибутив. Путей несколько:

- скачать с сайта производителя или с одного из зеркал (дублирующих сайтов). К сожалению, не каждый может себе позволить скачать более гигабайта информации;

- ❒ купить на сайте производителя с доставкой по почте;
- ❒ купить в магазине или на рынке;
- ❒ взять у знакомых.

Желательно также посмотреть в Интернете список обновленных пакетов программ и скачать необходимые пакеты.

В начале

Прежде чем начать работу — ознакомьтесь с инструкцией. Если у вас коробочный вариант — в него входит брошюра с инструкцией по установке дистрибутива. Если вы не имеете коробочного варианта дистрибутива — не беда. На сайте Red Hat есть документ в формате PDF — называется "Руководство по инсталляции". Процесс подготовки к инсталляции и сама инсталляция подробнейшим образом описаны (на английском языке).

Список оборудования

При инсталляции система проверяет установленное оборудование и пытается самостоятельно определить его тип. В большинстве случаев это получается неплохо. Однако не всегда определение происходит корректно. Некоторые устройства система вообще может не найти. И тогда вам придется самостоятельно указать их параметры. Что вам необходимо знать об аппаратном обеспечении компьютера:

- ❒ количество установленных жестких дисков;
- ❒ есть ли RAID-контроллер, его чипсет и производитель;
- ❒ объем оперативной памяти;
- ❒ тип мыши;
- ❒ если мышь последовательная, к какому COM-порту подключена;
- ❒ тип видеокарты (объем памяти и марку чипсета);
- ❒ тип SCSI-контроллера (если он есть);
- ❒ тип монитора, его кадровую и строчную частоты, максимальное разрешение;
- ❒ если есть сетевая карта — ее тип и чипсет;
- ❒ тип различных плат расширения (если они присутствуют).

Как правило, достаточно современное распространенное оборудование определяется нормально. Однако могут быть проблемы с мышью, имеющей колесо прокрутки (ее может просто не оказаться в списке) или с видеокартой (неверное определение размера оперативной памяти или типа процессора. Например, видеокарта Geforce 2 MX может быть опознана как Geforce 2 и т. п.).

Дополнительная информация

В том случае, если в компьютере установлена сетевая карта и он подключен к локальной сети, необходимо также знать следующую информацию о сетевых настройках:

- IP-адрес;
- сетевую маску;
- адрес шлюза по умолчанию;
- IP-адрес DNS-сервера;
- доменное имя;
- имя компьютера в сети.

Если в сети используется динамическое распределение адресов, некоторые пункты из этого списка знать необязательно.

Предполагаемый объем инсталляции

Весьма желательно представлять себе, сколько места займет установленный дистрибутив Linux. А для этого необходимо четко знать, что за систему вы устанавливаете. Вряд ли следует весь диск отвести под один раздел Linux. Тем более что на компьютере вполне может быть установлено несколько различных операционных систем.

Немного забегая вперед, посмотрим, что предлагает фирма Red Hat в качестве стандартного решения:

- Workstation-class (рабочая станция). Инсталляция данного типа требует не менее 1,2 Гбайт свободного места на винчестере, если вы устанавливаете GNOME или KDE. При установке одновременно и GNOME, и KDE потребуется не менее 1,5 Гбайт свободного дискового пространства;
- Server-class (сервер). Требуется 650 Мбайт при минимальной инсталляции и не менее 1,2 Гбайт при выборе всех пакетов для установки;
- Laptop-class (ноутбук) — требования аналогичны инсталляции типа "рабочая станция";
- Custom-class (выборочная инсталляция). Требует 300 Мбайт для минимальной инсталляции. Если диск имеет более 1,2 Гбайт свободного места, инсталлятор предлагает автоматически разбить диск на разделы. В противном случае предстоит ручное разбиение диска.

Замечание

На первый взгляд — достаточно внушительные требования. Однако не следует забывать, что в инсталляционный комплект входит большое количество разнообразного программного обеспечения. Конечно, если вам необходимо установить

только роутер (маршрутизатор, от англ. *Route* — маршрут, программное, аппаратное или программно-аппаратное решение, обеспечивающее передачу сетевых пакетов из одной сети в другую) или что-то подобное, нет необходимости занимать на жестком диске так много места. Есть специальные дистрибутивы, предлагающие объем инсталляции в одну-две дискеты. Мы же рассматриваем здесь среднестатистическую инсталляцию, о которой вы и прочтете далее.

Теперь вы представляете, какой объем необходим для инсталляции дистрибутива. Не следует забывать, что помимо самой операционной системы вы будете устанавливать и свое программное обеспечение, записывать свою информацию. Для функционирования операционной системы необходимо будет так же создать так называемый Swap-раздел (своп-раздел, раздел подкачки). Он используется для временного хранения части информации из оперативной памяти. Такая операция необходима, когда какому-то процессу срочно понадобилось большое количество оперативной памяти. Если в системе для выделения этому процессу оперативной памяти не хватает, ядро операционной системы переносит неиспользуемые в данный момент процессы из оперативной памяти на своп-раздел. А когда необходимость в использовании большого количества оперативной памяти отпадет, возвращает эти процессы из своп-раздела в оперативную память. Кроме того, из оперативной памяти на своп-раздел могут быть перенесены и длительно неиспользуемые процессы.

Разбиение жесткого диска

Теперь, когда у нас есть представление об объеме, который будет занимать дистрибутив, рассмотрим вопрос разбиения жесткого диска. В зависимости от того, в каком качестве будет выступать ваша система (сервер, рабочая станция, экспериментальная система и т. п.), изменяются и требования по разбиению жесткого диска на разделы. Для начинающих Red Hat Linux предлагает автоматическое разбиение жесткого диска. Для остальных существуют рекомендации по разбиению. Эти рекомендации — не истина в последней инстанции. Всегда найдется система, которая предъявит специфические требования.

Далее приведен список каталогов и рекомендации по вынесению их на отдельные разделы.

Каталог /

Каталог / является корнем файловой системы. Все остальные каталоги являются подкаталогами каталога /. Поскольку каталог / не может быть смонтирован в другом каталоге, обязательно создается корневой раздел.

Каждый каталог файловой системы Linux, не имеющий своего собственного раздела, является частью корневого раздела.

Обычно каталоги, размещаемые в корневом каталоге, не занимают много места и кардинально не увеличиваются во время эксплуатации системы. Каталоги такого типа (/bin, /dev, /etc, /mnt и т. п.) обычно не помещаются на отдельные разделы, а хранятся в корневой файловой системе.

Каталог /bin

Каталог /bin содержит только исполняемые файлы, используемые в основном администратором. Список файлов, содержащихся в этом каталоге, уже долгое время не претерпевает изменений, поэтому размер каталога /bin увеличивается только тогда, когда системный администратор устанавливает новые административные пакеты. Поскольку это происходит крайне редко, размер каталога /bin можно считать неизменным, а это позволяет поместить его в корневой раздел. Каталог /bin не зависит от других каталогов и не нуждается в свободном дисковом пространстве для выполнения своих задач.

Каталог /boot

Каталог /boot содержит все компоненты, необходимые для загрузки ядра операционной системы. Это могут быть несколько образов ядер операционной системы, карты модулей, конфигурационный файл, содержащий информацию о необходимых компонентах для запуска операционной системы. В процессе эксплуатации эти файлы изменяются только тогда, когда производится компиляция ядра. Перекомпиляция ядра, как правило, не увеличивает занимаемое каталогом /boot дисковое пространство. Если каталог /boot находится в разделе, полностью заполненном информацией, это никоим образом не влияет на нормальную загрузку операционной системы. Это свойство позволяет размещать каталог /boot в корневом разделе. Однако из-за проблемы 1024 цилиндра для каталога /boot зачастую создается небольшой раздел в начале жесткого диска.

Замечание

Не все компьютеры могут производить загрузку с цилиндра жесткого диска, большего, чем 1024. И не все загрузчики решают эту проблему. *(Подробнее об этом см. в разд. "Применение рекомендаций").*

Каталог /dev

В каталоге /dev размещаются специальные файлы устройств, предоставляющие интерфейс для доступа к различному оборудованию компьютера. В этом каталоге находится единственный исполняемый файл — makedev, который используется для создания файлов новых устройств.

Файлы устройств создаются только при установке нового оборудования. Однако если каталог /dev находится на переполненном разделе, то создать

новый файл устройства не удастся, что, как минимум, приведет к невозможности функционирования этого устройства. Каталог /dev не занимает много дискового пространства, однако обычно содержит более тысячи файлов устройств. Поскольку каталог /dev не увеличивается в размерах, он обычно размещается в корневой файловой системе.

Каталог /etc

Вся информация о настройках файловой системы содержится в файлах и подкаталогах, находящихся в каталоге /etc. Каталог /etc обычно не увеличивается в размерах, поскольку конфигурационные файлы программ редко занимают место более чем 15—20 Кбайт. По этой причине каталог /etc обычно размещают в корневой файловой системе. Однако в каталоге /etc находится несколько файлов, изменяемых в процессе эксплуатации операционной системы. В частности, это файлы, содержащие список доступных дисковых разделов и смонтированных разделов. Поэтому, если каталог /etc находится в переполненном дисковом разделе, нормальное функционирование операционной системы нарушается.

Каталог /home

В каталоге /home находятся каталоги пользователей системы. Для систем, в которых существует только несколько пользователей, для каталога /home обычно отдельный раздел не выделяется. Если в системе более десяти пользователей, имеет смысл создать для каталога /home отдельный дисковый раздел. Это позволит избежать проблем с переполнением диска. Для каталога /home рекомендуется также использовать программу quota, что даст возможность ограничить доступное для каждого пользователя место на жестком диске. В больших локальных сетях существует практика размещения каталога /home на сетевой файловой системе (NFS). Это дает пользователю возможность, помимо легкости администрирования и резервного копирования, получать доступ к своему домашнему каталогу с любого компьютера локальной сети.

Каталог /lib

В этом каталоге содержатся основные библиотеки операционной системы. Обновление или установка библиотек обычно производятся только при модернизации системы. Поскольку состав библиотек давно устоялся, резкого изменения занимаемого дискового места в случае их модернизации не происходит. Даже переполнение дискового раздела не оказывает влияния на нормальное функционирование библиотек из каталога /lib. Поэтому каталог /lib размещают в корневой файловой системе. Еще одним аргументом в пользу размещения каталога /lib в корневой файловой системе является то, что практически все исполняемые файлы (в частности, системные утилиты из каталогов /lib и /sbin) используют библиотеки из этого каталога.

Каталог /lost+found

В каждой файловой системе (разделе) автоматически создается каталог /lost+found. В нем утилита fsck размещает записи о файлах этой файловой системы, структура которых оказалась нарушенной. Поскольку каталог создается автоматически, нет необходимости заботиться о его размещении.

Каталог /mnt

Каталог /mnt предназначен для размещения точек монтирования. Обычно в этом каталоге находятся каталоги /floppy и /CDROM, являющиеся точками монтирования дискет и компакт-дисков. Помимо этого, в каталоге /mnt могут монтироваться разнообразные файловые системы, в том числе и NFS. Каталог /mnt никогда не расходует дисковое пространство, поэтому обычно он размещается в корневой файловой системе. Как уже упоминалось ранее, если дисковый раздел, в котором располагается каталог /etc, будет переполнен, автоматическое монтирование файловых систем станет невозможным.

Каталог /opt

Каталог /opt предназначен для установки программного обеспечения, не входящего в стандартный состав операционной системы, к примеру, сервера баз данных Interbase. Размеры каталога /opt сильно зависят от устанавливаемого программного обеспечения. Поэтому для каталога /opt рекомендуется создать отдельный дисковый раздел. Переполнение этого дискового раздела влияет на функционирование программного обеспечения, находящегося в каталоге /opt, и практически не затрагивает нормального функционирования операционной системы.

Каталог /proc

Каталог /proc является точкой монтирования псевдофайловой системы. Файлы и каталоги, находящиеся в каталоге /proc, на самом деле являются интерфейсом к некоторым параметрам операционной системы и не занимают место на жестком диске. Поэтому нет необходимости выделения для каталога /proc отдельного раздела. Каталог /proc создается в корневой файловой системе.

Каталог /root

Этот каталог является личным каталогом пользователя root. В нем находятся конфигурационные файлы этого пользователя. Не рекомендуется использование каталога /root для хранения файлов, поскольку каталог /root находится в корневой файловой системе, переполнение которой приведет к неправильному функционированию операционной системы.

Каталог /sbin

Каталог /sbin по функциональному назначению подобен каталогу /bin. Поскольку изменения в каталоге /sbin маловероятны, каталог этот размещается в корневой файловой системе.

Каталог /tmp

Каталог /tmp используется для хранения временных файлов, создаваемых приложениями операционной системы. Поскольку временные файлы могут иметь размеры, исчисляемые десятками мегабайт, и не все приложения удаляют по окончании работы свои временные файлы, размер каталога /tmp за короткое время может достичь огромных размеров. По этой причине крайне желательно выделять для каталога /tmp отдельный дисковый раздел.

Переполнение каталога /tmp не приводит к краху операционной системы, однако процессы, использующие каталог /tmp, перестают нормально функционировать. По этой причине каталог /tmp при перезагрузке операционной системы очищается от всех файлов.

Каталог /usr

Большинство приложений операционной системы устанавливается в каталог /usr, в связи с чем каталог /usr занимает большое количество дискового пространства и увеличивается после установки нового приложения. По этой причине для каталога /usr фактически всегда выделяется дисковый раздел. Также достаточно часто выделяется дисковый раздел и для каталога /usr/local.

Каталог /var

В каталоге /var приложения размещают свои рабочие файлы. Например, почтовые файлы, файлы групп новостей, буфер печати, файлы для сервера FTP и т. п. В этом же каталоге находятся файлы журналов различных служб операционной системы. Отсюда следует, что в процессе жизнедеятельности операционной системы объем каталога /var не является постоянным. В случае переполнения каталога /var бóльшая часть процессов операционной системы перестает корректно функционировать. Поэтому рекомендуется создать для каталога /var отдельный дисковый раздел. В зависимости от назначения сервера отдельные дисковые разделы могут создаваться и для хранения файлов серверов FTP, HTTP, каталогов /var/log и /var/spool.

Создание разделов на клиентских машинах

Требования, предъявляемые к рабочим станциям, менее жесткие, чем те, которые предъявляются к серверам. Это связано с тем, что на клиентских машинах выполняется значительно меньше процессов, способных вызвать

переполнение раздела. Кроме того, последствия переполнения раздела на клиентской машине не столь опасны, как на сервере.

На клиентских машинах можно создавать два или три раздела.

Создание разделов на сервере

Обычно на серверах требуется создавать больше разделов, чем на клиентских машинах, поскольку очень важно обеспечить максимальную устойчивость работы каждого сервера. Помимо своп-раздела на серверах создаются отдельные разделы для каталогов /, /tmp, /usr, /var. На многих серверах в зависимости от их назначения создаются и другие дополнительные разделы.

Сервер DNS

Сервер DNS предоставляет данные всем остальным компьютерам сети. Для повышения надежности обычно создается несколько дисковых разделов. Для каталога /var обязательно должен быть выделен отдельный раздел не менее 100 Мбайт, поскольку при работе служба DNS генерирует большие файлы журналов и файлы резервных копий.

Сервер NIS

К дисковым разделам ведомого сервера NIS предъявляются те же требования, что и к серверу DNS, за исключением того, что рост журналов не должен вызывать заполнения раздела, содержащего файлы карт YP. (Обычно они размещаются в каталоге /var/yp/maps.) Поэтому рекомендуется выделить отдельные дисковые разделы для каталогов /var и /var/log. Раздел /var/log должен иметь размер не менее 50 Мбайт, а размер раздела /var по крайней мере вдвое превосходить ожидаемый максимальный размер файлов карт YP.

К ведущим серверам NIS предъявляются те же требования, что и к ведомым, причем разделы /var и /var/log должны быть еще больше.

Почтовый сервер

На почтовых серверах для каталогов /var и /var/spool необходимо создать отдельные дисковые разделы достаточно большого объема. Каталог /var/log должен располагаться в разделе размером не менее 100 Мбайт. Требуемый размер этого раздела пропорционален нагрузке на сервер. Каталогу /var/spool должен быть выделен объем, достаточный для размещения всех почтовых сообщений для всех пользователей, обслуживаемых данным почтовым сервером. Если почтовый сервер обслуживает достаточно много пользователей, размер этого раздела может составлять несколько гигабайт.

Серверы FTP и HTTP

Все FTP- и HTTP-серверы в своих разделах /var или /var/log должны иметь не менее 100 Мбайт свободной дисковой памяти, предназначенной для раз-

мещения файлов журналов. Требуемый объем свободного пространства пропорционален нагрузке на сервер.

Кроме того, вся структура каталогов, доступная FTP- или HTTP-демонам, должна располагаться в своем собственном разделе. Это позволит монтировать в системе данный раздел с особыми параметрами (например, с разрешением доступа "только для чтения"). Если службы FTP и HTTP функционируют на одном и том же сервере, для сохранения их данных следует использовать отдельные разделы. Это упростит настройку параметров вычислительной среды.

Сервер NFS

Каталоги, предоставляемые в качестве ресурсов NFS, должны располагаться в отдельных разделах. Это упрощает их резервное копирование, а так же предохраняет сервер от переполнения системных разделов. Величина NFS-раздела зависит только от администратора и политики фирмы. У некоторых это сотня мегабайт, у других — сотни гигабайт.

Сервер Samba

Как и в случае NFS, экспортируемые средствами Samba каталоги должны располагаться в отдельных разделах. Требования по объему раздела абсолютно аналогичны серверу NFS.

Серверы новостей

Серверы групп новостей обрабатывают большое количество временных данных, имеющих сравнительно невысокую ценность. Спулы (под этим термином понимается временное хранилище информации, в частности, очередь печати, почтовый файл пользователя и т. п.) файлов групп новостей могут иметь очень большой размер и должны обеспечивать быстрый доступ к данным.

Для /var/spool/news необходимо выделить отдельный раздел, который, в идеале, должен располагаться на отдельном жестком диске. Это способствует повышению производительности системы. Кроме того, в случае отказа дискового устройства со спулом новостей на новом устройстве потребуется лишь создать пустые разделы. Кроме того, поскольку спулы новостей обычно содержат множество мелких файлов, отношение количества индексных блоков к общему объему дискового пространства должно быть в три или четыре раз больше, чем в случае обычных файловых систем.

Серверы баз данных

Планирование разделов на серверах баз данных должно выполняться при участии администраторов баз данных. Большинство крупных СУБД устанавливается в нескольких файловых системах, размещенных на нескольких дисковых устройствах. Кроме того, часто используются один или более раз-

делов без файловых систем (raw, "сырой раздел"), предназначенных для хранения данных.

Достаточный объем свободного дискового пространства должен быть зарезервирован и для создания файлов журналов, размещаемых в каталоге /var или /var/log.

Серверы приложений

На серверах приложений выполняется программное обеспечение, работа которого обычно имеет жизненно важное значение для организации. Во многих случаях отказ любого из таких серверов вызывает остановку работы части или даже всей компании.

Выполняемые файлы функционирующих на сервере приложений программ обычно размещаются в каталогах /opt или /usr/local. В любом случае, этот каталог должен располагаться в собственном разделе, поскольку объем его будет увеличиваться при каждой модернизации эксплуатируемых программ. Кроме того, следует обеспечить объем дискового пространства, достаточный для размещения нескольких копий приложения. Это упростит модернизацию приложений и обеспечит возможность быстрого отката в случае необходимости. Идеальный вариант — размещение раздела на RAID-массиве.

Сервер общего назначения

Схема создания разделов на сервере общего назначения должна разрабатываться с учетом двух требований. Во-первых, система должна быть способна предоставлять пользователям множество различных служб. Во-вторых, должна быть обеспечена возможность быстрого запуска сервера.

При принятии схемы разделения жесткого диска системные администраторы любого уровня должны учитывать приведенные выше рекомендации, собственный опыт и даже результаты применения метода проб и ошибок.

Применение рекомендаций

На практике применение вышеуказанных рекомендаций может выглядеть следующим образом .

О проблеме 1024 цилиндра. Как мы уже отмечали, не все компьютеры могут производить загрузку с цилиндра жесткого диска, большего, чем 1024. И не все загрузчики с этим справляются. Поэтому, во избежание возникновения проблемы, необходимо создать раздел /boot величиной 16—32 Мбайт, до 1024 цилиндра. Большим его делать смысла не имеет, а вообще размер зависит от того, будете вы держать на этом разделе несколько версий ядра или нет.

Создание своп-раздела. Общее правило для него: RAM × 2. Правило это достаточно корректно для 80% случаев. Но для случаев специфических с размерами своп-раздела необходимо разбираться экспериментально. Впро-

чем, никто не мешает создать несколько разделов свопа и подключить к системе или создать специальные своп-файлы.

Для систем, у которых мало памяти (менее 32 Мбайт), рекомендуется выделять под своп-раздел не менее 64 Мбайт. Сейчас крайне редко можно встретить компьютер с таким объемом оперативной памяти. Поэтому стандартный объем своп-раздела на сегодняшний день — 128 или 256 Мбайт.

Более опытным пользователям можно рекомендовать в процессе работы следить за использованием своп-раздела командой `free` или `top`. Если использование своп-раздела систематически превышает 50% — желательно увеличить его размер или создать своп-файл.

В зависимости от назначения системы можно выделить три их категории:

- домашний (офисный) компьютер, испытательный сервер, сервер небольшой локальной сети;
- удаленный сервер, сервер приложений (обобщенный);
- специальные серверы.

Первый тип систем — простой, мгновенного обслуживания, практически нет угрозы взлома и большой нагрузки, поэтому диск можно разбить всего лишь на 2—3 раздела:

- / — корневой;
- /boot — загрузочный (если надо);
- /swap — раздел подкачки (своп-раздел).

Для второго и третьего типа систем общепринятая практика разбиения диска — создание отдельных разделов для каждого (или для группы) основных каталогов файловой системы. Это увеличивает безопасность и отказоустойчивость системы и, кроме того, удобно для выдачи пользователям дисковых квот. Самый лучший вариант: отдельный раздел — отдельный винчестер.

Достигаемые цели: защита от атак, гибкое управление дисковыми квотами, более быстрая загрузка (впрочем, для серверов это не актуально), легкое резервирование и восстановление системы, лучшая контролируемость файловой системы в целом.

Для систем второго и третьего типа рекомендуется такая разбивка:

- раздел / — 256 Мбайт, здесь находятся каталоги /bin, /sbin и т. п.;
- раздел /boot — 16—32 Мбайт, все образы ядер должны находиться здесь;
- раздел /usr — более 256 Мбайт, поскольку бóльшая часть исполняемых файлов Linux устанавливается в этот раздел;
- раздел /home — N Мбайт пропорционально количеству пользователей + + размерность квоты на каждого пользователя + небольшой запас. Например, 10 Мбайт на пользователя × на количество пользователей;

- раздел /var — более 256 Мбайт, содержит файлы, которые могут изменяться (например, log-файлы, почтовые ящики);
- раздел /tmp — более 256 Мбайт, раздел для временных файлов. Сильно зависит от типа приложений.

Системы третьего типа отличаются особыми требованиями к определенным разделам. К примеру, серверу FTP необходимо выделить отдельный раздел для хранения файлов.

И в заключение. Если у в эксплуатации находятся несколько однотипных систем, старайтесь сделать максимально похожие конфигурации дисковых разделов и операционной системы — будет намного проще сопровождать и администрировать эти компьютеры.

Проблемы с оборудованием

Если у вас нетривиальная конфигурация компьютера, вполне может случиться, что какое-то устройство не установится. В этом случае остается через Интернет обращаться к FAQ, HOWTO, конференциям и службам рассылки. Воспользуйтесь конференцией **ru.linux** — там общаются очень толковые специалисты, наверняка помогут. Так же на сайтах производителей дистрибутивов обычно существуют форумы поддержки и списки аппаратного обеспечения, которое нормально не функционирует под Linux.

Обычно проблемы с оборудованием возникают в следующих случаях:

- очень новая видеокарта. Раньше приходилось ждать по полгода, пока энтузиасты напишут драйвер. Сейчас ситуация с драйверами исправляется. По крайней мере, лидер на рынке видеокарт nVIDIA выпускает драйверы под Linux;
- принтеры. Можно подобрать драйвер похожего принтера или ждать выхода Linux-драйверов;
- модемы. Для нормальных модемов проблем нет. С так называемыми Win-модемами сложнее. На сайте **www.linmodems.org** можно найти драйверы для некоторых типов модемов. В частности, хорошо работают Win-модемы на чипсете Lucent;
- некоторые сетевые карты. По этому поводу существует специальный HOWTO, в котором подробно описывается решение проблем;
- RAID-контроллеры. Поищите драйверы на сайте производителя, почитайте соответствующий HOWTO;
- SCSI-контроллеры. Обратитесь в конференцию **ru.linux**, почитайте FAQ и HOWTO;
- манипулятор "мышь". Не всегда удается задействовать колесо прокрутки или дополнительные кнопки;
- экзотическая периферия. Тут уж как повезет...

Ссылки

- **www.redhat.com/support/manuals** — руководства и документация.
- The Official Red Hat Linux x86 Installation Guide — название говорит само за себя.
- **linuxiso.org** — специальный сайт, содержащий iso-образы.
- **www.linuxlinks.com** — почти полный список существующих дистрибутивов.
- **www.linux-ve.chat.ru** — виртуальная библиотека Linux.
- **www.debian.org** — сайт дистрибутива Debian.
- **www.stormlinux.com** — сайт дистрибутива Storm Linux.
- **www.corel.com** — сайт фирмы Corel, производителя одноименного дистрибутива.
- **www.redhat.com** — сайт дистрибутива Red Hat.
- **www.ksi-linux.com** — сайт дистрибутива KSI.
- **www.blackcatlinux.com** — сайт дистрибутива Black Cat.
- **www.asplinux.ru** — сайт дистрибутива ASP Linux.
- **www.linuxmandrake.com/ru** — русская версия дистрибутива Mandrake.
- **www.caldera.com** — сайт дистрибутива Caldera.
- **www.bestlinux.net** — сайт дистрибутива Best Linux.
- **www.turbolinux.com** — сайт дистрибутива Turbo Linux.
- **www.slackware.com** — сайт дистрибутива Slackware.
- **www.suse.de** — сайт дистрибутива SuSE.

Глава 10

Требования, предъявляемые к инсталлируемой системе

Как можно будет увидеть, дистрибутивы Red Hat Linux допускают следующие варианты инсталляции:

- рабочая станция (Workstation-class);
- сервер (Server-class);
- ноутбук (Laptop-class);
- выборочная инсталляция (Custom-class).

Типовые профили инсталляции — рабочая станция, сервер и ноутбук — представляют собой максимально простые варианты установки операционной системы Linux, ориентированные на начинающих пользователей. Для опытных пользователей существует вариант выборочной инсталляции, позволяющий практически полностью контролировать процесс инсталляции и устанавливаемые пакеты.

По умолчанию Linux устанавливается как полнофункциональная система с большим количеством запущенных сервисов. Большинство из них не нужны всем пользователям и, будучи установлены, могут снизить эффективность системы или нанести вред ее безопасности. Идеально, когда каждый из сервисов работает на отдельной машине.

Правильная инсталляция операционной системы Linux — залог стабильности и безопасности работы, и, кроме того, благодаря ей экономится время, необходимое для последующего удаления лишних пакетов и переконфигурации системы. Выберите в программе инсталляции, какие компоненты системы следует инсталлировать, а затем войдите в каждый из компонентов, чтобы отметить, какие пакеты надо устанавливать, а какие нет (Select individual packages).

Поскольку набор пакетов сильно зависит от дистрибутива, невозможно дать исчерпывающий список, что необходимо инсталлировать, а что нет. Здесь вам помогут книги и информация по безопасности системы, а также вышеуказанные стандартные профили, созданные специалистами Red Hat.

Офисная система

Современное офисное рабочее место — это, прежде всего, локальная сеть во всех ее проявлениях: доступ в Интернет, электронная почта, доступ к Web-серверам, клиент ICQ (возможно, даже использующий корпоративный сервер). Поскольку человек за компьютером проводит бóльшую часть рабочего времени, ему должно быть максимально удобно. Поэтому вполне логичным является предпочтительное использование графического интерфейса X Window System (достаточно трудно представить работу современного офисного служащего с приложениями, имеющими текстовый интерфейс). Тем более, что в последние годы массовое распространение получила идеология "что вижу на экране — то получу на принтере", а без графического интерфейса такое обеспечить невозможно. Таким образом, X Window System обязательно должна входить в инсталлируемый пакет.

Далее, во время инсталляции необходимо выбрать, какой тип графической оболочки будет устанавливаться: KDE, GNOME или оба сразу. Этот вопрос заслуживает более подробного рассмотрения. Идеология KDE и GNOME, по сути, близка к Microsoft Windows: стандартный для всех приложений интерфейс, простота конфигурирования и настройки, упор на использование мыши и, главное, большой набор приложений, идеологически и функционально тесно связанных друг с другом. При этом, в отличие от Microsoft Windows, возможности настройки и изменения внешнего вида системы несравнимы — Microsoft Windows остается здесь далеко позади KDE и GNOME. В комплект обеих оболочек входят наборы программ, используемых в офисной жизни: текстовые редакторы, органайзеры, электронные таблицы, калькуляторы, программы презентаций, создания диаграмм и графиков и т. д. Поэтому, если в процессе обычной деятельности офиса нет необходимости передавать во "внешний мир" документы в электронном виде (читай "в формате Microsoft Office"), вполне достаточно использовать стандартные программы, входящие в KDE и GNOME.

В чем коренное отличие этих оболочек друг от друга? Помимо того, что они базируются на разных библиотеках, идеология их так же достаточно различается. Неофициальный лозунг KDE: "Windows лучше чем Windows". И действительно, при работе в KDE не покидает ощущение, что это Windows с несколько переделанным интерфейсом. Тем не менее, многие считают, что попытка вполне удалась. Благодаря четкому контролю и управлению проектом приложения KDE слаженно взаимодействуют друг с другом и имеют унифицированный интерфейс. Оборотной стороной комфортности является некоторая тяжеловесность и требовательность к ресурсам. Правда, для достаточно современных компьютеров (Celeron 500 МГц, 96 Мбайт оперативной памяти) это уже несущественно. Вывод: лучший вариант для пользователей, привыкших к Windows.

Разработчики GNOME попытались пойти несколько иным путем: совместить эргономику и либеральные требования к аппаратному обеспечению. Получилось удобно и красиво. Внешний вид можно настроить как у Windows, Mac, NextStep, есть и множество оригинальных схем. Система имеет большой набор приложений и потребляет достаточно мало системных ресурсов. К сожалению, в GNOME чувствуется недостаточная координация проекта, в т. ч. не до конца унифицирован интерфейс основных офисных приложений (правда, в последнее время положение исправляется). Вывод: вполне приемлемый вариант для не очень мощных компьютеров.

По большому счету, KDE или GNOME — это дело вкуса. Нравится — используй. И там, и там можно запускать программы от оппонента, функциональность практически одинакова, варианты оформления — вопрос эстетики, а в офисе необходимо работать. В конце концов, можно установить и KDE, и GNOME, и пользоваться обеими системами попеременно.

Если же в офисе стоят маломощные компьютеры (типа Pentium), следует поискать менеджер окон, способный быстро работать на медленных компьютерах. Такими могут быть IceWM, AfterStep или FVWM95.

Ну, а если фирма нуждается в обмене электронными документами формата Microsoft Office, придется установить StarOffice или OpenOffice. Почему именно их? На сегодняшний день — это наиболее мощные и проработанные офисные пакеты, способные достаточно успешно работать с файлами формата Microsoft Office.

Таким образом, для офисной системы достаточно установить X Window, KDE или GNOME и офисные приложения. Потребуется также интернет-браузер, клиент ICQ и, может быть, клиент Samba.

Все остальное: игры, системы верстки, мультимедийные программы — излишество. Вряд ли обычному офисному работнику необходим и компилятор C++ или Java. И совсем нет надобности в установке HTTP-сервера или сервера баз данных.

Рекомендации для администратора

Старайтесь сделать офисную систему одинаковой для всех компьютеров. Некоторая избыточность компенсируется простотой резервного копирования и восстановления поврежденной операционной системы. Когда в офисе три машины, можно все делать не спеша, и каждый компьютер создавать и настраивать индивидуально. Но когда их пара десятков, у администратора только один выход — максимально унифицировать программное обеспечение.

Старайтесь минимизировать набор установленного программного обеспечения. Во-первых, его будет проще обслуживать и модернизировать, а во-вторых, работники не будут постоянно задавать вам вопросы: "А как сделать то-то в том-то?" Когда используются три-четыре программы, их несложно

освоить и запомнить их особенности. Когда же в офисе используются десятки приложений, администратор перестает заниматься администрированием и превращается в "группу поддержки пользователей".

Домашняя система

Домашняя система обычно используется для разнообразных экспериментов, игр, воспроизведения аудио и видео, программирования и т. п. Поэтому здесь трудно посоветовать что-либо конкретное. В любом случае следует установить X Window и KDE или GNOME. Достаточно часто на домашнюю систему устанавливают все находящиеся в дистрибутиве пакеты, а потом постепенно убирают лишнее.

На домашнем компьютере так же обычно находится по крайней мере еще одна операционная система, как правило, Microsoft Windows. Особых проблем при инсталляции это не вызывает. Проблема возникнет тогда, когда после переустановки Windows обнаружится, что вы не можете загрузить Linux. Решение этой проблемы достаточно простое: загружаетесь с загрузочной Linux-дискеты и вызываете команду `lilo` (или `lilo -r`). Можно так же воспользоваться программой LoadLin и потом запустить `lilo`.

Сервер

Само название "сервер" подразумевает, что компьютер выполняет специфические задачи, и обычный пользователь на нем не работает. Серверы бывают разные, поэтому набор инсталлируемых пакетов для различных серверов может сильно отличаться. Однако практически на любом сервере не нужны X Window, а также различные утилиты, применяемые обычными пользователями (MP3-проигрыватели, утилиты для работы с графическими файлами и т. п.). Не инсталлируются также различные компиляторы и исходные коды программ (с целью увеличения безопасности системы).

Опытные администраторы предпочтут произвести выбор пакетов самостоятельно. Для начинающих рекомендуется выбрать тип инсталляции Server-class (сервер) и уже потом осуществить выбор необходимых программных пакетов, устанавливаемых на сервер.

Далее приведен список нежелательного для установки на сервере программного обеспечения.

- arpwatch — содержит две программы: arpwatch и arpsnmp. Эти утилиты предназначены для мониторинга сети. Они следят за сетевым трафиком в Ethernet- или FDDI-сетях, ведут базу данных участников соединений и могут пересылать отчеты об определенных изменениях по почте. Для сервера не нужны, можно установить на администраторской машине.

- chkfontpath — терминальная программа для добавления, удаления и просмотра каталогов, включенных в поисковые пути для шрифтов X-сервера.
- finger — утилита, которая позволяет получить информацию о пользователях системы (регистрационное имя, домашний каталог и т. п.).
- finger-server — содержит демон finger, который позволяет удаленным пользователям получать информацию о пользователях на сервере.
- fwhois — позволяет организовывать запросы к базам данных WHOIS.
- ghostscript — набор программного обеспечения, которое предоставляет интерпретатор PostScript и интерпретатор файлов Portable Document Format (PDF). На сервере — ненужный пакет.
- ghostscript-fonts — шрифты, используемые интерпретатором GhostScript во время визуализации текста.
- mpage — получает на вход текстовый или PostScript-файл, изменяет размер текста и печатает файл на PostScript-принтере, умещая на одном листе несколько страниц текста.
- nfs-utils — утилиты и демон для корневого сервера NFS.
- Network Information Service (NIS) — система, которая предоставляет сетевую информацию (регистрационное имя, пароль, имя домашнего каталога, информацию о группах) всем компьютерам в сети.
- ntalk — клиентская программа и демон для организации чатов между пользователями с использованием протокола Internet Talk.
- portmapper — утилита, управляющая RPC-соединениями, которые используются такими протоколами, как NFS и NIS.
- rsh — включает комплект программ, которые позволяют пользователям выполнять команды на удаленной машине, присоединяться к удаленным машинам и копировать файлы между компьютерами. Используют пересылку пароля по сети в незашифрованном виде. Для замены этого семейства следует использовать комплект ssh.
- rusers — позволяет пользователям узнавать, кто присоединен к удаленным компьютерам в сети. Выдает результаты для определенного списка компьютеров или для всех машин в локальной сети.
- rwho — выводит результаты для всех компьютеров в локальной сети на которых запущен демон rwho.
- rsh-server — сервер, необходимый для работы утилит rsh, rlogin, rcp, которые предоставляют доступ к командам на удаленной машине.
- rusers-server — сервер, который принимает пользовательские запросы и позволяет им узнать, кто подключен к серверу.

- rwall-server — демон, который позволяет принимать сообщения от удаленных пользователей.
- screen — позволяет иметь несколько соединений на одном терминале.
- SNMP (Simple Network Management Protocol) — протокол, используемый для сетевого управления.
- telnet — программа для подключения к удаленным системам через Интернет. Использует пересылку пароля в незашифрованном виде. Для замены telnet используйте ssh.
- Trivial File Transfer Protocol (TFTP) — используется для загрузки бездисковых рабочих станций.
- talk-server — демон, который позволяет беседовать с терминала с пользователями из удаленных UNIX-систем.
- telnet-server — демон, который реализует telnet-протокол на сервере.
- tftp-server — демон, реализующий TFTP-сервер, который позволяет пересылать файлы на или с удаленных машин (используется для удаленной загрузки бездисковых компьютеров).
- ypserv — NIS (Network Information Service) — сервер, который предоставляет сетевую информацию (NIS) всем машинам в сети.

Ссылки

- **http://www.redhat.com/support/manuals** — руководства и документация.
- **http://linux.webclub.ru/books/linuxsos/index.html** — безопасность и оптимизация Linux. Редакция для Red Hat — русский перевод.
- The Official Red Hat Linux x86 Installation Guide.

Глава 11

Инсталляция

Как мы уже отмечали, установка операционной системы сильно зависит от того, что, в конечном итоге, требуется получить: сервер, офисную систему или домашний компьютер. Соответственно, имеется несколько вариантов инсталляции операционной системы.

Во-первых, инсталляцию можно производить в графическом или текстовом режиме. Дистрибутив Red Hat по умолчанию пытается запустить инсталляцию операционной системы в графическом режиме, однако, если на компьютере установлено 16 Мбайт оперативной памяти или менее, инсталляция происходит в текстовом режиме. Во-вторых, есть вариант установки в так называемом режиме kickstart, который позволяет произвести инсталляцию по заранее созданному профилю. Этот режим чаще всего используется для создания большого количества идентичных систем. И в-третьих, есть возможность произвести обновление установленной ранней версии дистрибутива Red Hat до текущей с минимальным вмешательством пользователя. При этом сохраняются все данные пользователей, производится обновление ядра операционной системы, модулей и пакетов программ.

Для начала процесса инсталляции необходимо загрузить компьютер с носителя, содержащего специальную программу инсталляции. Таким носителем может быть компакт-диск дистрибутива или загрузочная дискета (обычная или с поддержкой PCMCIA).

Создание загрузочной дискеты и загрузка

Для создания загрузочной дискеты необходимо иметь дискету, отформатированную в DOS без плохих секторов, и компакт-диск с дистрибутивом Red Hat.

В каталоге /dosutils компакт-диска находится утилита копирования образа дискеты — rawrite. Файлы образов дискет находятся в каталоге /images:

- boot.img — для обычной инсталляции;
- bootnet.img — для инсталляции по сети (FTP, HTTP, NFS);
- pcmcia.img — для ноутбуков.

С помощью утилиты rawrite изготовим загрузочную дискету:

```
rawrite
Enter disk image source file name: E:\images\boot.img
Enter target diskette drive: a:
Please insert a formatted diskette into drive A: and press <ENTER>:
E:\dosutils>
```

После выполнения указанных действий в нашем распоряжении оказывается загрузочная инсталляционная дискета. Точно так же изготавливается загрузочная дискета с поддержкой PCMCIA или дискета для сетевой инсталляции Linux.

После изготовления загрузочной дискеты необходимо перегрузить компьютер, зайти в установки BIOS и установить в качестве первого загрузочного устройства флоппи-дисковод. Далее все просто — вставляем дискету в дисковод и происходит загрузка инсталляционной программы. В зависимости от того, какой образ дискеты был создан, дальнейшая инсталляция может проводиться обычным образом или по сети.

Графическая инсталляция

В процессе работы программа инсталляции использует пять (с 1 по 5 — для текстового режима, со 2 по 5 и 7 — для графического) виртуальных консолей, на которые выводятся различные сообщения, позволяющие решать проблемы, возникающие при установке операционной системы. В табл. 11.1 приводится информация об этих консолях.

Таблица 11.1. *Доступные консоли при инсталляции Red Hat Linux*

№ консоли	Комбинация клавиш	Содержание
1	<Ctrl>+<Alt>+<F1>	Диалог инсталляции
2	<Ctrl>+<Alt>+<F2>	Командная строка
3	<Ctrl>+<Alt>+<F3>	Сообщения от программы инсталляции
4	<Ctrl>+<Alt>+<F4>	Системные сообщения
5	<Ctrl>+<Alt>+<F5>	Другие сообщения
7	<Ctrl>+<Alt>+<F7>	Графический дисплей X Window

Процесс инсталляции в графическом режиме начинается с режима текстового. Таковы особенности инсталляции. В самом начале программа инсталляции спросит у вас, существуют ли дискеты с драйверами для специфического оборудования. Это необходимо для корректной инсталляции оборудования,

не поддерживаемого программой. Такие диски могут быть предоставлены производителем оборудования. Еще одним источником решения проблем с инсталляцией системы на компьютеры со специфическим оборудованием является страница **www.redhat.com/support/errata/**.

После этого программа установки предложит выбрать источник, с которого производится инсталляция дистрибутива. В нашем случае — это CD-ROM.

Выбор языка инсталляции

Затем программа переходит в графический режим и предлагает выбрать язык, который будет использоваться во время инсталляции операционной системы и при локализации. Среди этих языков есть русский и украинский. Выбор языка позволяет программе инсталляции также определить временную зону.

Выбор типа клавиатуры

Следующий этап — определение типа клавиатуры. Помимо выбора клавиатуры, необходимо также выбрать клавиатурную раскладку (язык). Если вы при инсталляции ошиблись, в дальнейшем переконфигурирование клавиатуры можно будет произвести с помощью программы kbdconfig.

Выбор типа мыши

Следующим этапом будет определение типа мыши. Если в предложенном списке вашей мыши нет, выберите тот тип мыши, с которым она совместима. Система пытается автоматически подобрать тип мыши, но у нее это не всегда хорошо получается. Если мышь подсоединяется к COM-порту, следует указать, к какому именно порту она подключена. Если у вашей мыши в наличии только две кнопки, установите флажок **Эмулировать 3 кнопки**. В Linux достаточно активно используется третья кнопка, и нет причин ее лишаться. Эмуляция нажатия третьей кнопки производится одновременным нажатием обеих кнопок мыши.

В дальнейшем переконфигурирование мыши можно будет произвести с помощью программы mouseconfig.

Выбор типа инсталляции

Выполнение подготовительных этапов завершено, переходим к самому процессу инсталляции. Здесь необходимо выбрать, какого типа система будет устанавливаться:

- ❒ рабочая станция;
- ❒ сервер;

- ❒ лэптоп;
- ❒ выборочная инсталляция.

От выбранной системы зависит, какие пакеты программ будут инсталлированы и каким образом будет автоматически разбит жесткий диск (если вы, конечно, позволите программе это сделать).

Автоматическое разбиение жесткого диска на разделы

При автоматическом разбиении жесткого диска (рис. 11.1) происходит следующее:

- ❒ рабочая станция:
  - инсталлятор удаляет все существующие Linux-разделы;

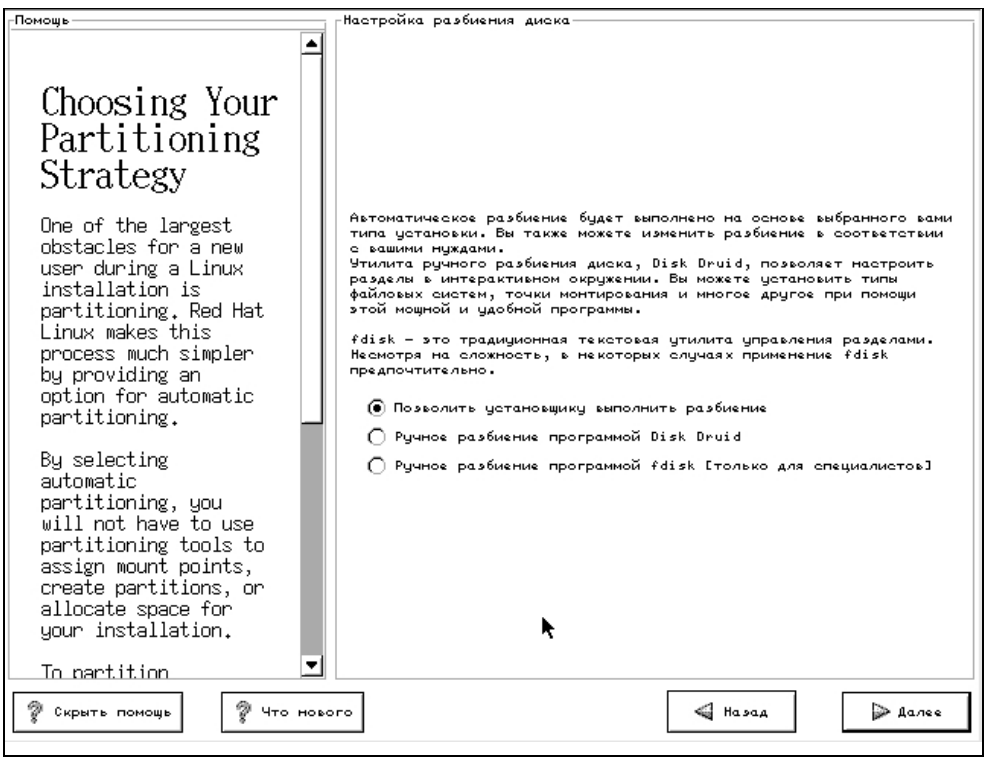

Рис. 11.1. Выбор типа разбиения жесткого диска

- создает:
  - ◊ своп-раздел 64 Мбайт;
  - ◊ раздел 16 Мбайт, монтируемый как /boot;
  - ◊ раздел переменной емкости (в зависимости от оставшегося на диске места);

❒ сервер:

- инсталлятор удаляет все разделы на жестком диске (как Linux, так и других операционных систем);
- создает:
  - ◊ своп-раздел 256 Мбайт;
  - ◊ раздел 256 Мбайт, монтируемый как /;
  - ◊ раздел по крайней мере 512 Мбайт, монтируемый как /usr;
  - ◊ раздел по крайней мере 512 Мбайт, монтируемый как /home;
  - ◊ раздел 256 Мбайт, монтируемый как /var;
  - ◊ раздел 16 Мбайт, монтируемый как /boot;

❒ лэптоп — повторяет рабочую станцию;

❒ выборочная установка — повторяет рабочую станцию.

Ручное разбиение жесткого диска на разделы

Программа Disk Druid

Программа Disk Druid представляет собой достаточно простое средство ручного разбиения жесткого диска (рис. 11.2).

Замечание

При разбиении жесткого диска на разделы желательно указывать величину разделов не в мегабайтах, а в треках. Информация на жестком диске располагается на концентрических дорожках — треках. Эти треки разбиваются на блоки, которые логически объединяются в кластеры. Минимальный элемент, который выделяется под хранение файла, — кластер (по крайней мере, в большинстве операционных систем). Если размер в мегабайтах не совпадет точно с размером в треках, часть последнего трека раздела останется неиспользованной. Конечно, при современных емкостях жестких дисков это капля в море но, все равно, как-то неаккуратно.

Каждая строка в нижней секции **Разделы** представляет собой описание одного дискового раздела. Эта строка содержит следующие поля:

❒ **Точка монтирования** (Mount Point) — указывает точку монтирования в стандартной иерархии файловой системы, например, /boot;

- **Устройство** (Device) — указывает, на каком устройстве размещается дисковый раздел, например, hda6[1];

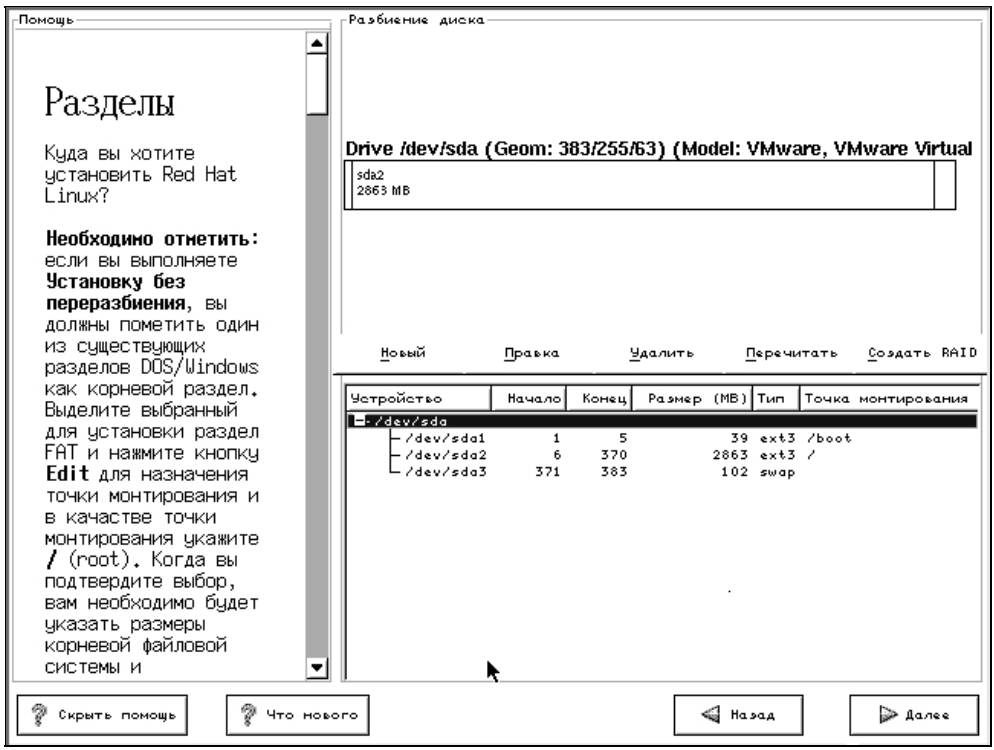

Рис. 11.2. Программа разбиения дисков Disk Druid

- **Размер (MB)** — указывает запрошенный размер дискового раздела;
- **Тип** (Type) — показывает тип дискового раздела, например, Linux Native или DOS.

[1] Жесткие диски бывают двух видов — с интерфейсом IDE и SCSI. Соответственно, и устройства имеют в Linux следующие имена: hd и sd. Третья буква определяет номер жесткого диска в системе: a — 1, b — 2, c — 3, d — 4 (в DOS это физические диски C:, D:, E:, F:). Каждому логическому разделу на жестком диске соответствует своя цифра. Первичных разделов на жестком диске может быть всего 4. Зато никто не мешает определить один расширенный раздел и на нем сделать множество вторичных разделов. Поэтому цифрами от 1 до 4 определяются первичные разделы, а 5 и далее — вторичные. Таким образом, если диск разбит на 2 раздела (пусть это первый IDE-диск), то в системе ему соответствует устройство hda, а первому и второму первичным разделам — устройства hda1 и hda2. Точно таким же образом происходит наименование SCSI-дисков и их разделов.

Каждая строка в верхней секции представляет собой описание жесткого диска и содержит следующие поля:

- **Drive** — содержит имя жесткого диска;
- **Geom:** [C/H/S] — показывает геометрию жесткого диска.

Кнопки программы Disk Druid:

- **Новый** (Add) — используются для добавления нового дискового раздела;
- **Правка** (Edit) — используется для изменения параметров дискового раздела, например, точки монтирования;
- **Удалить** (Delete) — используется для удаления раздела;
- **Перечитать** (Reset) — используется для приведения состояния программы Disk Druid к первоначальному виду. Все изменения, сделанные с дисковыми разделами, теряются;
- **Создать RAID** (Make RAID Device) — используется для создания RAID-массивов. Более подробную информацию см. в Official Red Hat Linux Reference Guide.

Как можно видеть, использовать программу Disk Druid весьма просто.

Fdisk

Кроме Disk Druid существует и программа fdisk. Она является консольным приложением и имеет текстовый интерфейс. Для получения справки по этой программе необходимо нажать клавишу <M>, для выхода без сохранения изменений — нажать клавишу <Q>, для выхода с сохранением изменений — клавишу <W>.

Форматирование разделов

После разбиения жесткого диска программой fdisk программа инсталляции предложит отформатировать созданные разделы. Если установить флажок **Проверять при форматировании на плохие блоки**, то одновременно с форматированием будет произведена проверка поверхности жесткого диска. Если жесткий диск не очень новый, рекомендуется включить его проверку, хотя эта операция занимает достаточно много времени. В случае использования программы Disk Druid необходимость форматирования раздела и проверки его на сбойные секторы определяется при создании раздела диска.

Инсталляция загрузчика операционной системы

Следующим шагом инсталляции будет выбор устанавливаемого загрузчика операционной системы — LILO или GRUB (рис. 11.3).

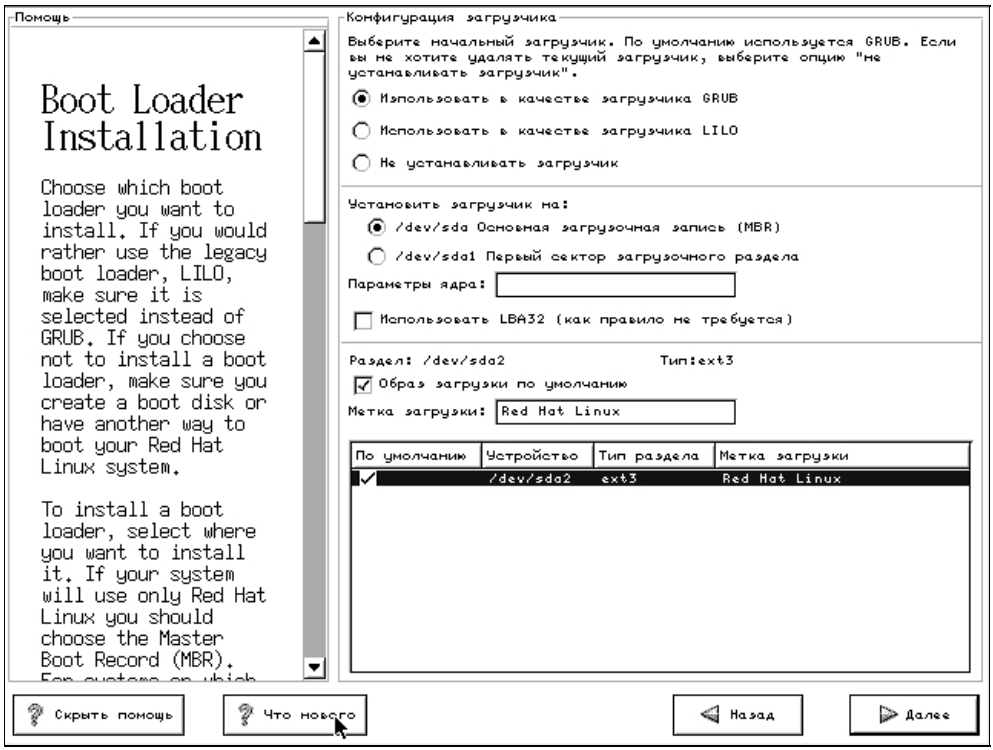

Рис. 11.3. Определение параметров загрузчика операционной системы

Вы можете установить загрузчик операционной системы в одно из двух мест:

- в главную загрузочную запись (MBR) — это рекомендованное место для установки загрузчика в том случае, если Linux является единственной операционной системой, или на компьютере установлена Windows 9*x*;
- в первый сектор вашего корневого раздела — рекомендуется для установки загрузчика операционной системы в том случае, если вы уже используете какой-то загрузчик.

Замечание

Если вы не установите загрузчик операционной системы, вам придется сделать специальную загрузочную дискету или воспользоваться для загрузки операционной системы Linux программой LOADLIN. (**ftp://metalab.unc.edu/pub/Linux/system/boot/dualboot/**). Можно также использовать программу SYSLINUX (**ftp://metalab.unc.edu/pub/Linux/system/boot/loaders/**).

В окне настройки загрузчика операционной системы есть поле ввода **Параметры ядра**. В него можно ввести параметры, которые передаются ядру опе-

рационной системы. В нижней части окна приводится список операционных систем, установленных на компьютере. Здесь можно выбрать загружаемую по умолчанию операционную систему.

После выбора типа загрузчика операционной системы (к примеру, GRUB) программа инсталляции предлагает установить пароль на загрузку операционной системы (рис. 11.4).

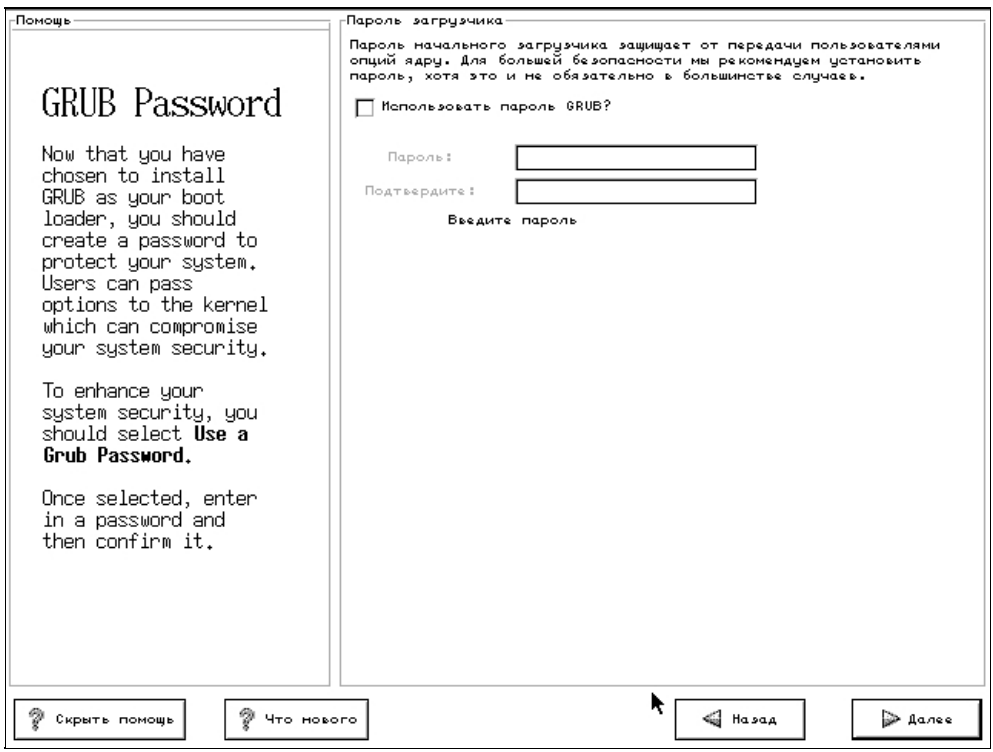

Рис. 11.4. Установка пароля для загрузчика операционной системы

Настройка сетевого интерфейса

Следующее, что вы увидите после установки загрузчика LILO (если в компьютере присутствует сетевая карта), это окно настройки сетевых параметров (рис. 11.5). Если сетевые параметры в локальной сети определяются динамически, используя протокол DHCP, достаточно установить соответствующую отметку. В противном случае придется указать:

- IP-адрес;
- маску сети;

- адрес сети;
- широковещательный адрес.

Помимо этого, необходимо будет указать и следующие данные:

- имя хоста;
- адрес шлюза;
- первичный сервер DNS;
- вторичный сервер DNS;
- третичный сервер DNS.

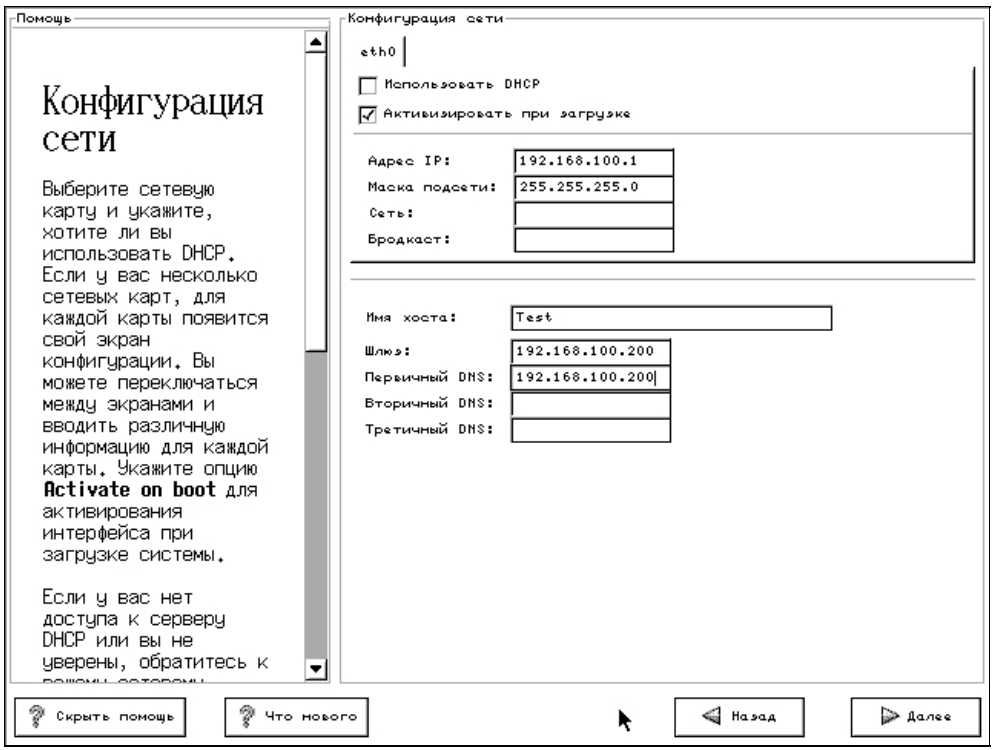

Рис. 11.5. Настройка сетевых параметров

Если вам неизвестны эти сетевые настройки, смело их пропускайте. Сетевой интерфейс всегда можно настроить позже.

Настройка брандмауэра

После настройки сетевого интерфейса программа инсталляции предложит настроить брандмауэр (firewall). Брандмауэр позволяет в определенной мере

защитить компьютер от сетевого вмешательства извне. Как видно из рис. 11.6, существует три уровня защиты: **Высокий**, **Средний** и **Без брандмауэра**.

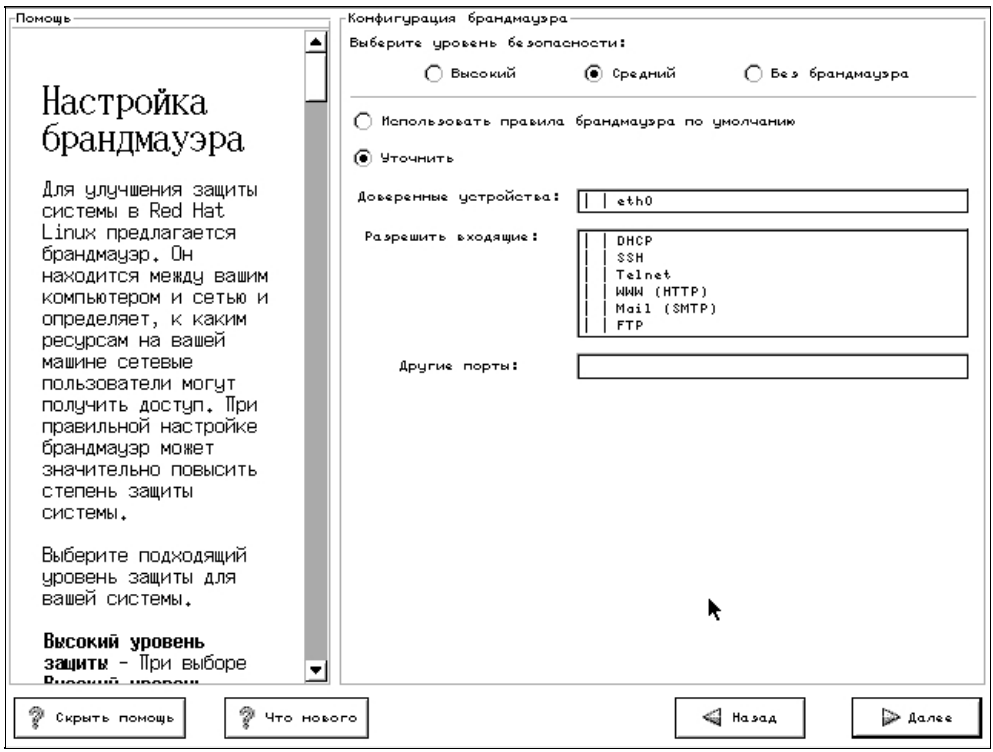

Рис. 11.6. Настройки брандмауэра

Высокий — высокий уровень защиты. Не разрешает сетевых соединений, кроме определенных правилами.

Следующие соединения разрешены:

- DNS-ответы;
- DHCP;
- пассивный режим FTP.

Следующие службы запрещены:

- активный режим FTP;
- IRC;
- RealAudio;
- удаленные клиенты X Window System.

При необходимости можно дополнительно разрешить сетевые службы, используя пункт **Уточнить**.

Средний — средний режим защиты.

Следующие сетевые службы запрещены:

- порты, меньшие чем 1023;
- NFS-порт (2049);
- локальный X Window System дисплей для удаленных X-клиентов;
- X Font сервер-порт.

Как и в предыдущем случае, используя пункт **Уточнить**, можно дополнительно разрешить сетевые службы.

Без брандмауэра — не определены сетевые правила.

Настройка часового пояса

Следующий пункт инсталляции — настройка часового пояса. Для этого нужно указать свое местоположение на карте мира. Можно также настроить временную зону в так называемом Координированном универсальном времени (Coordinated Universal Time, UTC). В дальнейшем вы можете настроить время с помощью утилиты timeconfig.

Настройка языковой поддержки

Операционная система Linux поддерживает одновременно несколько языков (рис. 11.7). При необходимости просто выберите нужные языки.

Пользовательский пароль

На этой странице программа инсталляции предлагает ввести пароль пользователя root и завести обычных пользователей системы. По поводу выбора паролей имеется много рекомендаций, в частности, пароль не должен быть короче восьми символов, должен содержать буквы и цифры в разных регистрах. Неплохой вариант: пароль, обозначающий какое-то русское слово, набранное латиницей. И запомнить легко, и подобрать практически невозможно. Например, ИндустриалиЗация — BylecnhbfkbPfwbz.

Конфигурация аутентификации

Следующий этап — конфигурация аутентификации. Как видно из рис. 11.8, по умолчанию установлены скрытые пароли и использование при создании хэша пароля алгоритма MD5. Это позволяет повысить устойчивость системы ко взлому и создавать пароли длиной до 256 символов. Остальные настройки дают возможность применять альтернативные варианты аутентификации. Более подробную информацию смотрите в специальной литературе.

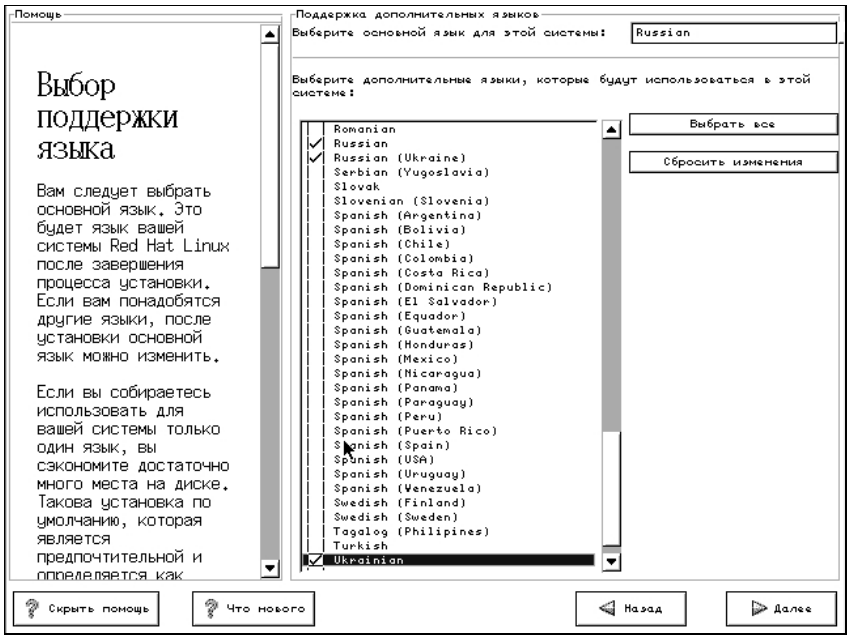

Рис. 11.7. Настройка языковой поддержки

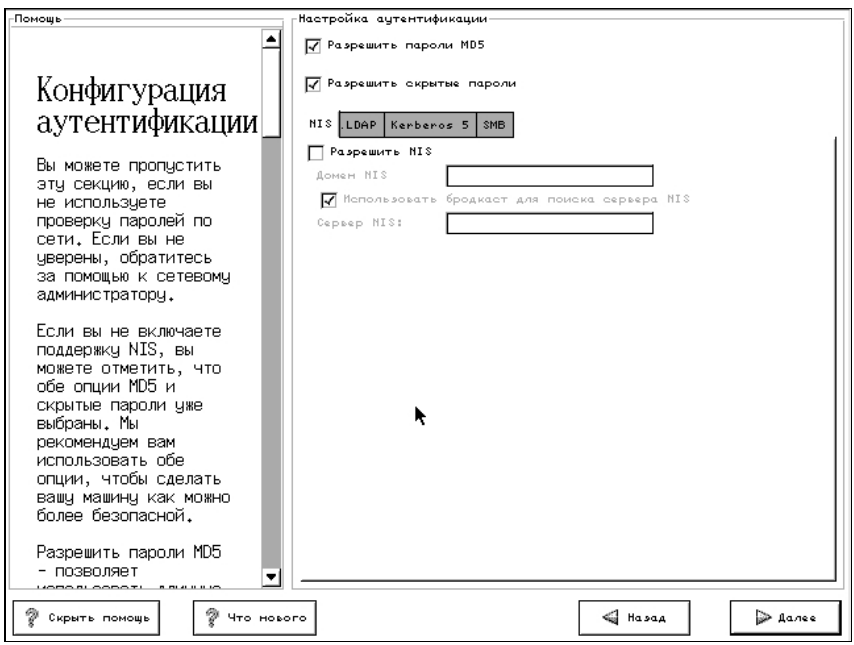

Рис. 11.8. Настройка аутентификации

Выбор инсталлируемых пакетов

Вот мы и подошли к выбору программных пакетов. В дистрибутиве Red Hat пакеты сгруппированы по функциональному признаку. И в зависимости от типа инсталляции программа выбирает те или иные пакеты. Впрочем, возможен и самостоятельный выбор пакетов. В нижней части окна программы инсталляции — краткое описание пакета. Некоторые пакеты программа устанавливает принудительно, поскольку считает, что они необходимы для нормального функционирования системы.

После индивидуального выбора пакетов обычно возникают неудовлетворенные зависимости. Часто тот или иной пакет дополнительно требует установки некоторых пакетов-библиотек или утилит. Программа инсталляции проверяет неудовлетворенные зависимости и выводит список необходимых пакетов (рис. 11.9).

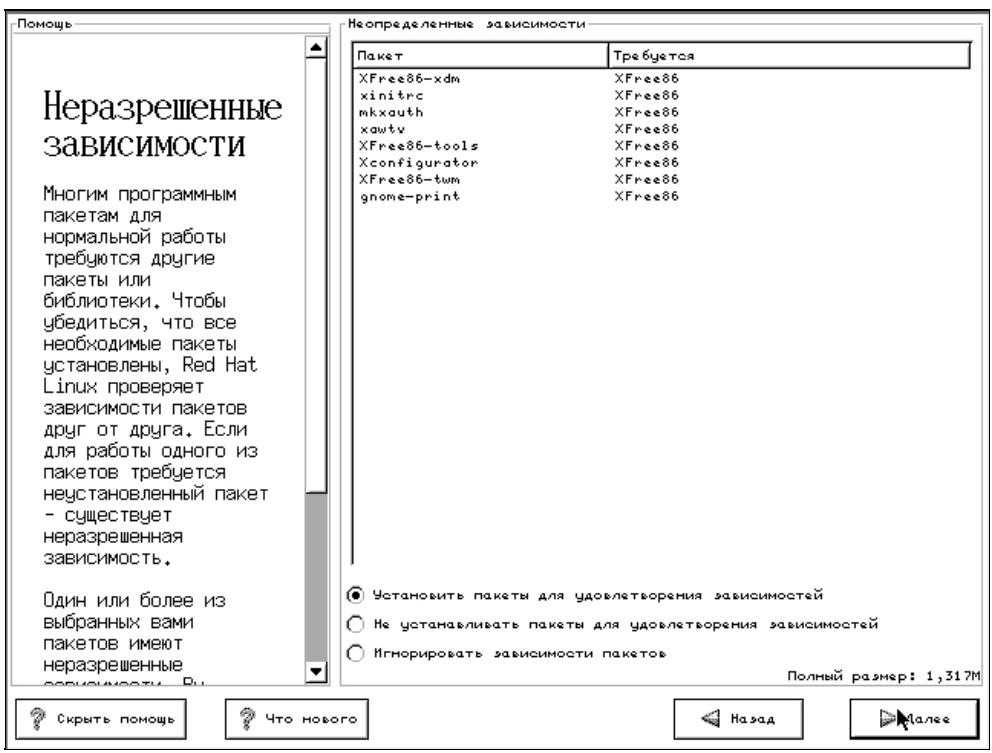

Рис. 11.9. Неудовлетворенные зависимости

Конфигурация X Window

Если вы выбрали среди пакетов установку X Window, самое время настроить используемую X Window аппаратуру. Программа инсталляции пытается самостоятельно определить тип видеокарты и объем видеопамяти, но не всегда это происходит корректно. Поэтому внимательно проверьте тип видеокарты и при необходимости выберите ту карту, которая присутствует в компьютере. Неверное определение видеокарты приводит к тому, что, в лучшем случае, она не будет работать столь производительно, как должна. В худшем — не удастся запустить X Window.

Далее следует указать тип монитора. Если в списке нет вашего монитора, не беда — просто укажите в соответствующем поле кадровую и строчную частоту его развертки. Завершающий штрих — выбор цветовой палитры и устраивающего вас экранного разрешения.

Обязательно протестируйте настройки, нажав на кнопку **Проверить**. В случае неправильной конфигурации настройки системы следует откорректировать. В этом же окне можно указать, какой тип входа (регистрации) в операционную систему вы предпочитаете — текстовый или графический.

Инсталляция

И наконец — инсталляция. По ее окончании вам будет предложено создать загрузочную систему. Перегрузите компьютер и проверьте функционирование операционной системы.

Текстовая инсталляция

Дистрибутив Red Hat Linux старается унифицировать текстовый и графический интерфейсы инсталляции. Конечно, в текстовом режиме все не так красочно, но суть остается той же. Начать инсталляцию в текстовом режиме можно, указав в строке загрузки `boot: text`.

В дальнейшем процесс идентичен инсталляции в графическом режиме. Для примера на рис. 11.10—11.14 показан процесс конфигурации X Window в текстовом режиме.

Инсталляция с жесткого диска

Помимо инсталляции с компакт-диска существует возможность произвести инсталляцию с жесткого диска (рис. 11.15). Для этого необходимо выбрать раздел жесткого диска и каталог, в котором находится ISO-образ инсталляционного диска (рис. 11.16).

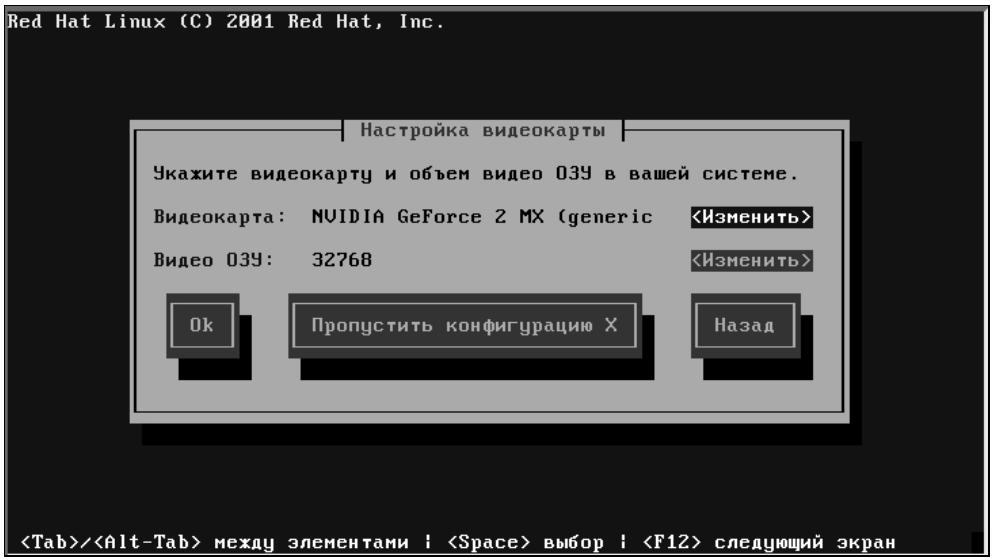

Рис. 11.10. Конфигурация X Window в текстовом режиме (выбор видеокарты)

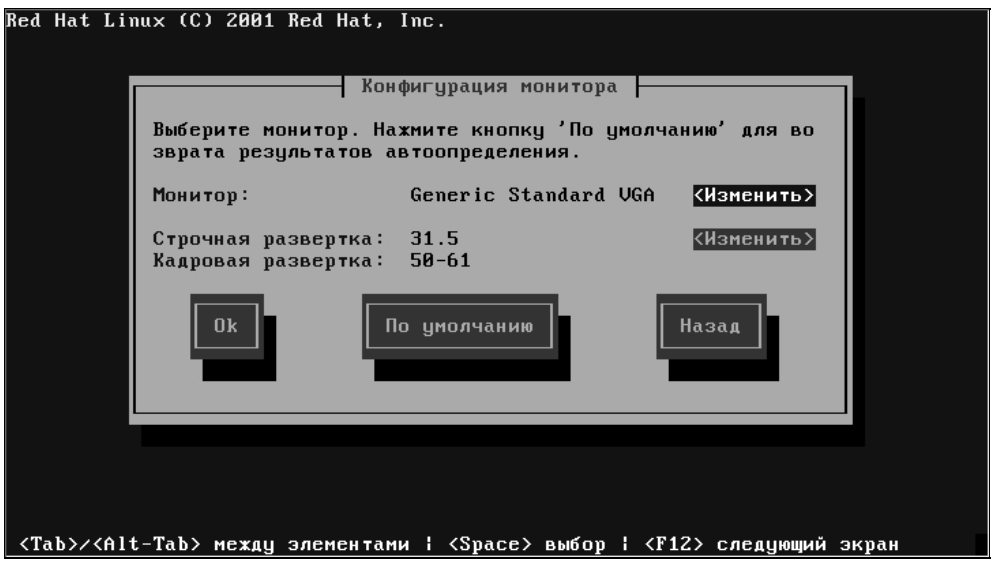

Рис. 11.11. Конфигурация X Window в текстовом режиме (выбор монитора)

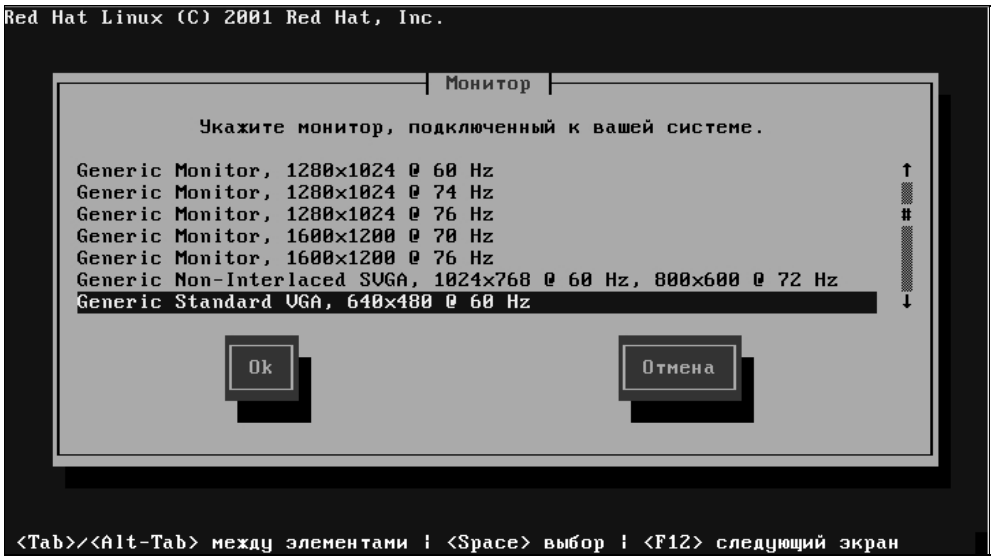

Рис. 11.12. Конфигурация X Window в текстовом режиме (выбор разрешения монитора)

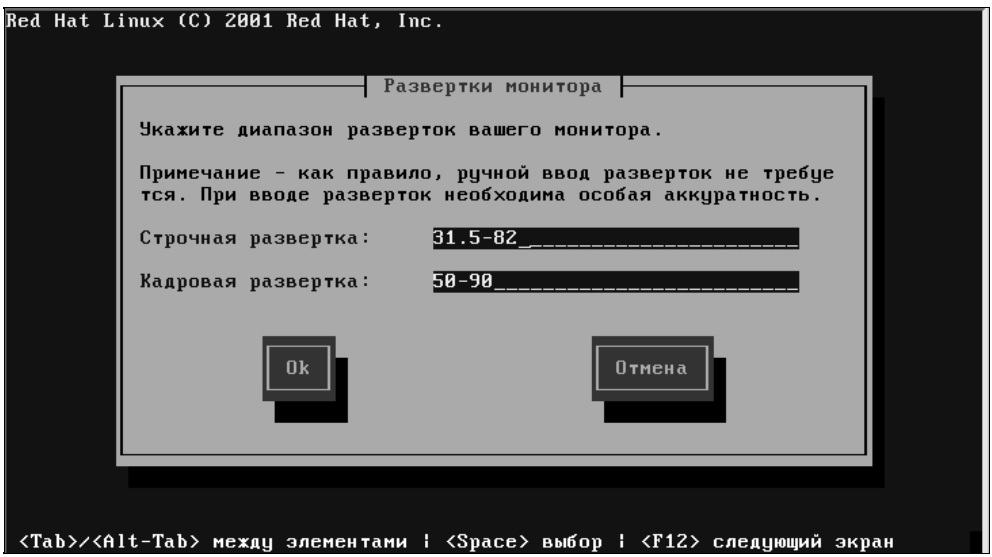

Рис. 11.13. Конфигурация X Window в текстовом режиме (установка частоты строчной и кадровой развертки)

Рис. 11.14. Конфигурация X Window в текстовом режиме (выбор видеорежима)

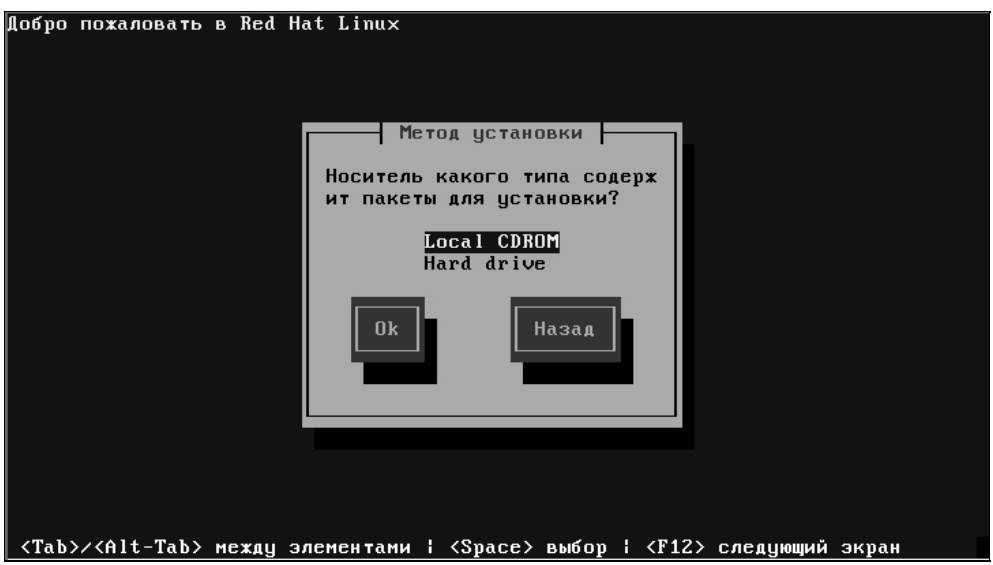

Рис. 11.15. Выбор инсталляции с жесткого диска

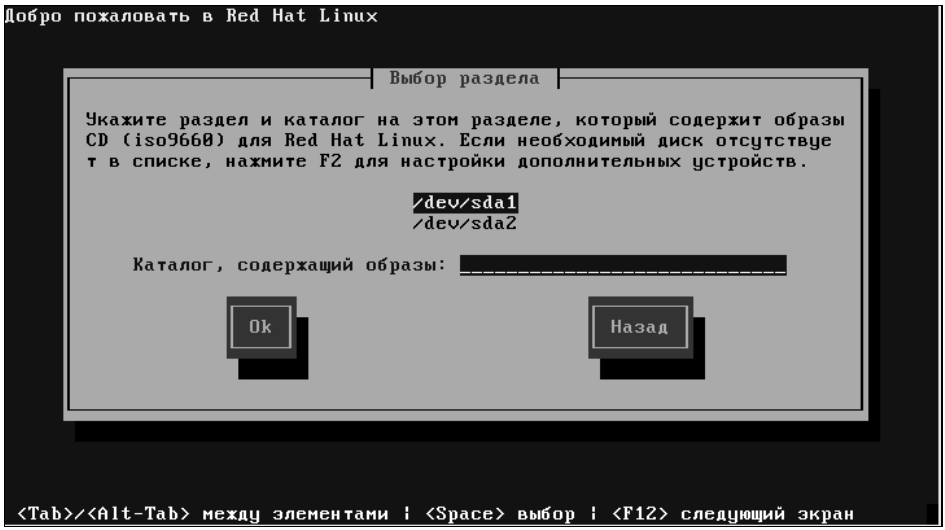

Рис. 11.16. Выбор раздела для инсталляции с жесткого диска

Сетевая инсталляция

Установить операционную систему также можно и по сети (рис. 11.17). Для этого необходимо создать специальную загрузочную дискету *(см. разд. "Создание загрузочной дискеты и загрузка").*

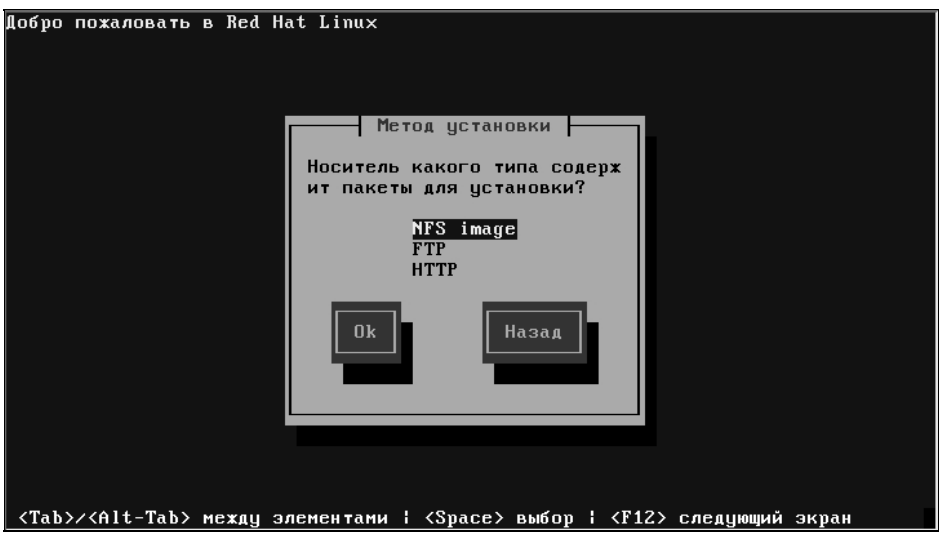

Рис. 11.17. Выбор типа сетевой инсталляции

На рис. 11.18 показан процесс инсталляции с использованием NFS, на рис. 11.19 показан процесс инсталляции с использованием FTP-сервера, на рис. 11.20 показан процесс инсталляции с использованием HTTP-сервера.

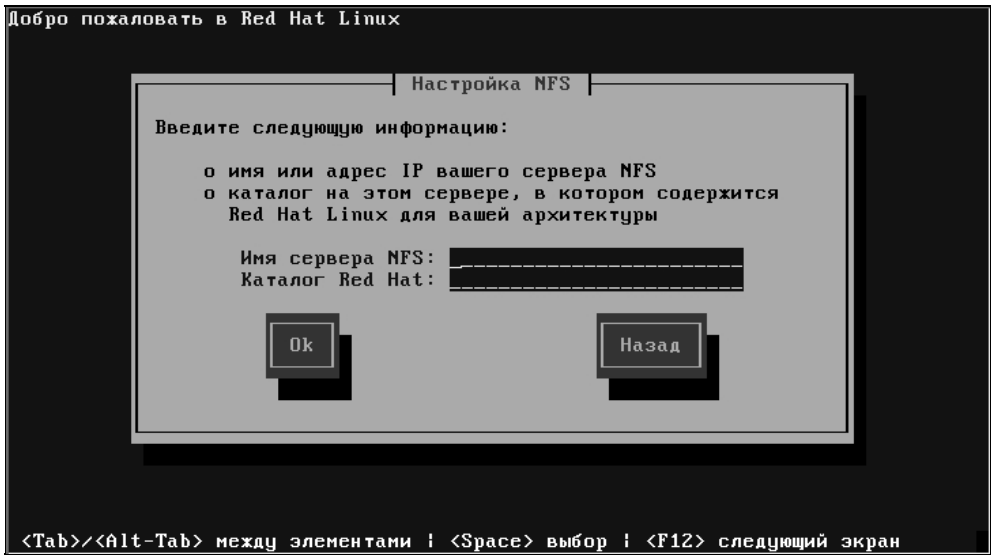

Рис. 11.18. Сетевая NFS-инсталляция

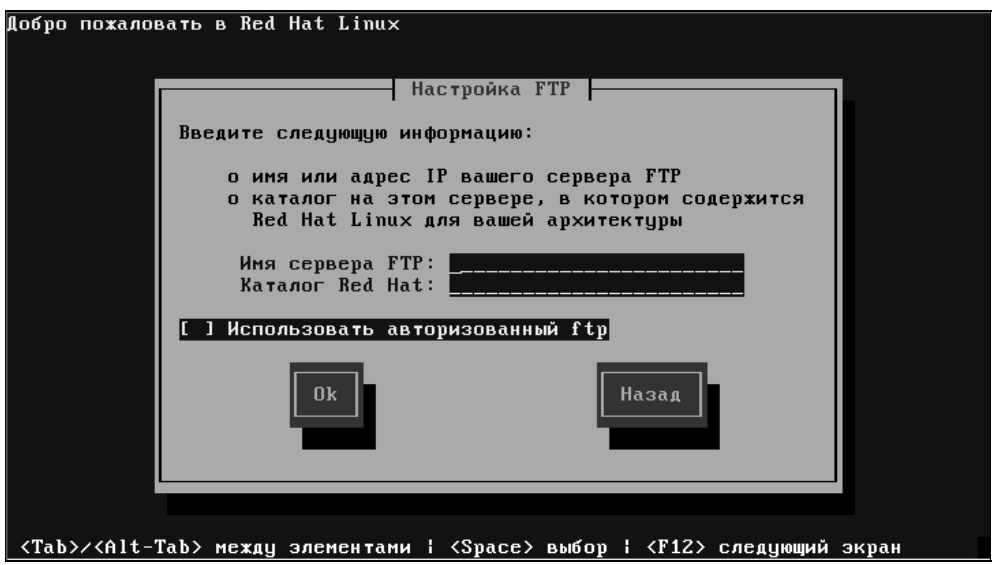

Рис. 11.19. Сетевая FTP-инсталляция

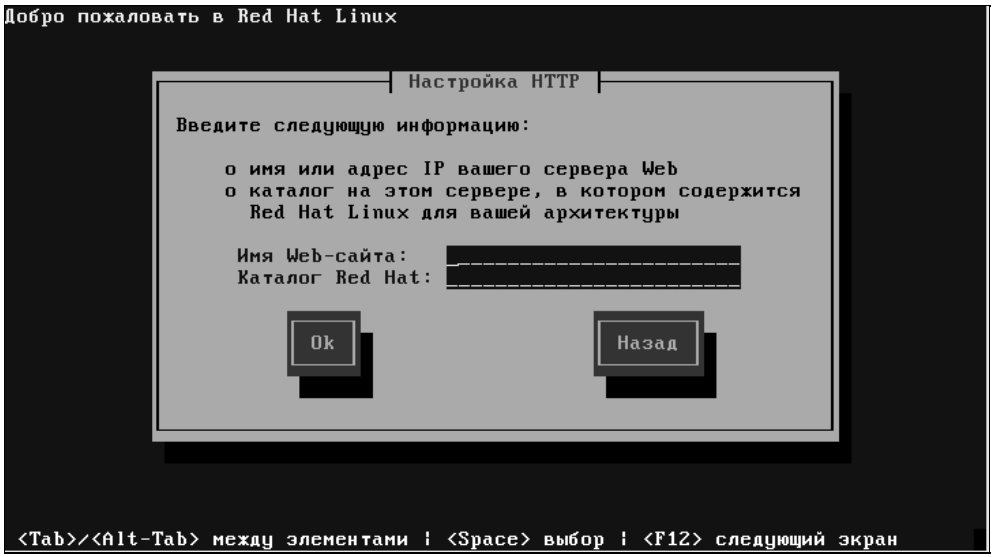

Рис. 11.20. Сетевая HTTP-инсталляция

Ссылки

- **http://www.redhat.com/support/manuals** — руководства и документация.
- The Official Red Hat Linux x86 Installation Guide — официальное руководство по инсталляции Red Hat Linux.

Глава 12

После инсталляции

По окончании процесса инсталляции необходимо заняться конфигурированием операционной системы и удалением лишних пакетов. Но сначала, если вы еще не сделали загрузочную дискету — сделайте, а лучше — две дискеты, т. к. дискета вещь ненадежная. Впрочем, восстановление системы можно произвести и с загрузочного дистрибутивного компакт-диска. В меню, выдаваемом при запуске с дистрибутивного диска, есть пункт **Rescue**, воспользовавшись которым, вы загрузите компьютер с образа восстановительной дискеты.

Второе, что необходимо — создать пользователей системы. Даже если на компьютере работает только один человек, и этот человек — вы, все равно следует создать обычного пользователя и работать только от его имени. Безопасность превыше всего. Работать от имени пользователя root крайне нежелательно. Малейшая ошибка, и операционной системы как не бывало. А восстановить удаленные файлы практически невозможно.

Следующий шаг — установка всех обновлений пакетов, которые следует предварительно получить из Интернета, по крайней мере, для сервера и для рабочих мест, подключенных тем или иным способом к сети.

Дальнейшие действия в целом зависят от варианта инсталляции операционной системы Linux и от назначения компьютера.

Домашний компьютер

Для этого типа компьютеров характерно использование большого количества разнообразного программного обеспечения. Поэтому, если у вас полный дистрибутив (порядка 4—5 дисков) — просмотрите на них все имеющиеся пакеты программ. Это удобно делать из X Window, поскольку в этой среде (в частности, GNOME) производится автомонтирование компакт-дисков, а также автоматический запуск менеджера пакетов RPM (конечно, только в том случае, если эти пакеты на компакт-диске находятся). С менеджерами пакетов мы уже знакомились *(см. гл. 8)*, поэтому здесь на их описании останавливаться не будем. В каждом менеджере пакетов можно посмотреть

краткое описание выбранного пакета и решить — нужен он или нет. Поскольку домашний компьютер обычно предназначен для экспериментов, смело устанавливайте пакеты: не понравится — деинсталлируете. Как правило, в течение одной-двух недель вы определитесь, что понравилось и будет использоваться, а что понадобится вряд ли. После чего следует обновить понравившиеся вам программные пакеты. А для этого необходимо настроить Интернет.

Настраивать модемное соединение намного проще в KDE или GNOME. В пунктах меню **Интернет** или **Администрирование** найдите программу kppp или ppp-соединение. Далее все достаточно прозрачно: логин, пароль, модем, телефонный номер для дозвона, и вы в Интернете.

Как уже говорилось ранее, за решением проблем и советами по настройке различного программного и аппаратного обеспечения лучше всего обращаться в новостную конференцию **ru.linux** — люди там грамотные и достаточно доброжелательные.

Офисный компьютер

Офисный компьютер отличает, в первую очередь, использование офисных приложений и однотипность настроек. Поэтому после инсталляции следует установить (если это не было сделано во время инсталляции) офисные приложения, используемые в вашей фирме. Причем крайне желательно, чтобы это были последние версии, поскольку сейчас Linux-сообщество очень серьезно занялось доработкой офисных приложений, и полезные изменения и дополнения появляются буквально каждую неделю. Офисные приложения вы выбираете сами, но есть и общие рекомендации. Если у вашей фирмы велик объем документации, приходящей или отправляемой во "внешний" мир в электронном виде — вам не обойтись без офисных приложений, способных корректно работать с документами формата Microsoft Office. В настоящее время из бесплатного программного обеспечения наиболее корректно работает с такими документами StarOffice фирмы Sun и FreeOffice, базирующийся на исходном коде StarOffice. Однако за функциональность все же необходимо платить, в данном случае — местом на жестком диске и требованиями к процессору и оперативной памяти.

Далее, если сеть не была настроена при инсталляции (офис подразумевает наличие локальной сети), это надо сделать сейчас. Затем, с помощью менеджера пакетов RPM следует отсеять все лишнее программное обеспечение. Как правило, могут быть установлены службы типа finger, r- (rlogin, rcopy и т. п.), telnet (клиент и сервер) и достаточно много такого, что является потенциальной брешью в защите системы. В идеале, на офисном компьютере не должно быть никаких сетевых служб, кроме SSH. По инерции в дистрибутивы входят (и по умолчанию устанавливаются на компьютер)

много устаревших пакетов, разработанных лет двадцать-тридцать назад. В те времена о мощи современных персональных компьютеров даже и не мечтали, Интернета не было как такового, а локальные сети только стали появляться. Что такое взлом, подбор паролей, трояны и вирусы — никто не знал. Поэтому пароли в сетевых приложениях типа telnet передавались в открытом виде, а порой вообще не использовались. В современном мире такое недопустимо. Поэтому лучше все потенциально опасное убрать от греха подальше.

После установки необходимого программного обеспечения, настройки сети, электронной почты и службы новостей рекомендуется установить и настроить клиент NTP — службу точного времени. Благодаря ей вы навсегда забудете о проблеме синхронизации системных часов компьютера. *(Подробную информацию об NTP см. в гл. 23.)*

И теперь, когда операционная система полностью настроена, крайне желательно сделать ее резервную копию. Вариантов может быть несколько:

- каждый компьютер предприятия имеет свою резервную копию;
- группа компьютеров имеет одну резервную копию, а для каждого компьютера резервируются его конфигурационные файлы;
- все компьютеры имеют одну резервную копию, а для каждого компьютера резервируются его конфигурационные файлы.

Первый вариант представляется несколько излишним. Нет смысла организовывать запись и хранение нескольких десятков компакт-дисков только из-за того, что у компьютеров разные имена и IP-адреса. С другой стороны, в случае проблемы на компьютере просто разворачивается резервная копия.

Вариант номер два. В фирме выделяются группы компьютеров, абсолютно идентичных по установленному программному обеспечению. Делается одна резервная копия. Дополнительно резервируются конфигурационные файлы каждого компьютера. В случае проблемы надо развернуть резервную копию и переписать соответствующие конфигурационные файлы (и информацию пользователя).

Третий вариант. Самый неудачный. В случае проблем надо развернуть резервную копию, переписать соответствующие конфигурационные файлы, информацию пользователя и доустановить отсутствующее программное обеспечение (либо наоборот, убрать ненужное).

И конечно — процедура ежедневного резервного копирования. Для ее облегчения рекомендуется выделить сервер, на котором должны храниться файлы и почтовые сообщения пользователей, и производить резервирование содержимого этого сервера.

Компьютер программиста, администратора

Компьютер программиста или системного администратора несколько отличается по установленному программному обеспечению от обычной офисной машины.

Как правило, на такие компьютеры устанавливается достаточно много специфичного программного обеспечения, которое может быть потенциально опасным для безопасности системы. Поэтому доступ к таким машинам должен быть ограничен как в физическом смысле, так и посредством сетевых соединений.

На компьютер программиста помимо разнообразных сред программирования, отладочных средств, компиляторов и интерпретаторов зачастую устанавливаются сервисы баз данных, FTP-, HTTP-демоны и т. п. Это необходимо для того чтобы в процессе отладки программы (или скрипта) программист (или администратор) ставил эксперименты на тренировочной, тестовой базе данных, а не на рабочем сервере базы данных фирмы.

Поскольку компьютер программиста часто содержит достаточно ценную информацию, а из-за особенности программистской деятельности подвергается повышенному риску, резервирование его данных желательно проводить два-три раза в день.

Приблизительно такой же спецификой обладает компьютер системного администратора. Помимо различных компиляторов и интерпретаторов (C/C++ для компиляции ядра и программ, Perl и Pyton для скриптов и т. п.), на него устанавливается специальное программное обеспечение для мониторинга сети и администрирования. Все вновь разработанные или модифицированные скрипты системный администратор должен попробовать сначала "на себе", и только после этого устанавливать на другие компьютеры.

Сервер

Самым специфичным компьютером, как правило, является сервер. Специфика эта возникает в зависимости от конфигурации, выполняемых задач, количества обслуживаемых пользователей и требований надежности. Поэтому мы дадим здесь лишь весьма общие рекомендации.

Во-первых, сразу по окончании процедуры инсталляции следует проверить, все ли аппаратное обеспечение сконфигурировано и работает правильно. Особое внимание надо обратить на SCSI-устройства и сетевые карты (если их более одной). По умолчанию конфигурируется только одна сетевая карта, все остальные, установленные в системе, приходится конфи-

гурировать самостоятельно. Устройства SCSI так же являются слабым местом. Если на сервере установлены и USB-устройства (правда, для чего они на сервере, представить трудно), надо проверить и их функционирование.

Затем следует установить новое программное обеспечение (и в дальнейшем отслеживать выход новых версий программ). Если необходимо — обновить и/или скомпилировать ядро операционной системы. Здесь надо быть особенно внимательным — для выполнения некоторых функций (например firewall) при компиляции необходимо включить свойства, обычно отключенные по умолчанию. Произвести необходимые настройки сервисов. Удалить все лишние для сервера программы.

Сервер — самая чувствительная к взлому система в локальной сети. От потери функциональности или замедления его работы страдают все работники фирмы, а если сервер почтовый или Web — то проблемы появляются и у людей, желающих отправить вам почту или посмотреть ваш Web-сайт. Именно поэтому, как и с простого офисного компьютера, с сервера необходимо удалить все потенциально опасные службы типа finger, r- (rlogin, rcopy и т. п.), telnet (клиент и сервер), NFS и т. п. На сервере, если это сервер, к примеру, только баз данных, должно стоять *только* программное обеспечение баз данных. И ничего другого. Никаких Web-серверов, игр, X Window и компиляторов.

Ниже приведен небольшой (далеко не полный) список пакетов, которые на серверах общего назначения не нужны:

- BOOTP (Boot Protocol) — используется для загрузки бездисковых рабочих станций. Если сервер не является сервером удаленной загрузки, нет необходимости оставлять этот пакет;
- DHCP (Dynamic Host Configuration Protocol) — протокол, который позволяет отдельным устройствам в IP-сетях получать от сервера конфигурационную информацию (IP-адрес, сетевую маску, широковещательный адрес и т. д.). Если сервер не является сервером DHCP — удалите и этот пакет;
- mt-st — включает программное обеспечение для управления устройствами чтения с магнитных лент: mt (для устройств magnetic tape devices) и st (для SCSI tape devices). Если на сервере не установлен стример — эти пакеты лишние;
- eject — позволяет пользователям извлекать диски (обычно это CD-ROM, Iomega Jazz и Zip) используя программные средства. Эта программа тоже не понадобится;
- apmd — демон расширенного управления питанием и сопутствующие ему утилиты. Такое программное обеспечение должно использоваться на ноутбуках, на сервере ему делать нечего;

- linuxconf — удобная утилита для настройки системы. По умолчанию установка ее не производится. Если она все же установлена, следует знать: помимо того, что эта программа занимает достаточно много места, она так же содержит и ошибки;
- isapnptools — включает утилиты для настройки карт ISA Plug and Play (PnP) и плат, которые совместимы со спецификацией ISA Plug and Play. Поскольку в современном компьютере вот уже на протяжении трех лет не устанавливаются ISA-устройства, наличие этого пакета нецелесообразно;
- setserial — системная утилита для просмотра и установки информации о последовательных портах. Используется на сервере модемного доступа и сервере управления кассовыми аппаратами с последовательным интерфейсом. Может быть необходима для маршрутизаторов, имеющих модемные соединения. Эту утилиту можно использовать при отладке соединения и управления источником бесперебойного питания (UPS). Для всех остальных типов серверов наличие ее нецелесообразно;
- kudzu — утилита для автоматического определения аппаратного обеспечения. Во время загрузки она может определить, какие устройства были добавлены или удалены из системы. Однозначного мнения, нужен или нет данный пакет на сервере, не существует;
- raidtools — включает утилиты, которые нужны для установки и управления программными RAID-массивами. Если программные RAID-массивы не используются — не устанавливайте;
- redhat-logos — файлы логотипов. Трата дискового пространства;
- redhat-release — на сервере не нужен;
- rmt — предоставляет удаленный доступ для резервного копирования. Как и все r-команды — потенциальная брешь в безопасности операционной системы;
- tux — встроенный в ядро HTTP-сервер. Позволяет ускорить обработку HTTP-запросов. Ни для каких серверов, кроме Web-сервера, не нужен. К тому же, на Web-сервере традиционно устанавливается сервер Apache.

После инсталляции и компиляции всего необходимого программного обеспечения рекомендуется удалить с сервера все компиляторы и подобные им программы. Это делается для того, чтобы злоумышленник, проникший на сервер, не смог скомпилировать или модифицировать необходимые ему утилиты. Как известно, знаменитый "червь Морриса" пересылал свой исходный код на компьютер жертвы, там себя компилировал и запускал на выполнение.

Создание разных серверов для разных задач упрощает процесс администрирования и управления ими и увеличивает контроль и настраиваемость для каждого из них.

Дальнейшие действия большей частью административные.

1. Создайте список файлов с их правами и владельцами. Регулярно проверяйте, не изменились ли атрибуты, права и владельцы файлов. Изменение этих параметров — один из признаков взлома или некорректной работы пользователей.
2. Заведите на сервере пользователя, которому делегируйте некоторые права — выключение сервера, монтирование дисковых разделов, административные задачи.
3. Проверьте, чтобы пользователь root не мог зайти через сеть или способом, отличным от входа с консоли.
4. Организуйте перенаправление почты пользователя root на обычного пользователя, отвечающего за администрирование. По умолчанию многие сервисы отсылают электронной почтой сообщения о проблемах, возникающих во время их работы.
5. Не работайте пользователем root — этот пользователь случайно может уничтожить операционную систему.
6. Настройте службу logrotate: на серьезных серверах log-файлы в 100—200 Мбайт — в порядке вещей.
7. Разработайте стратегию резервного копирования сервера.
8. Проводите учебное восстановление сервера из резервных копий раз в квартал. Это позволит и хорошо отработать процедуру восстановления, и проверять целостность резервных копий.
9. Обязательно приобретите источник бесперебойного питания с управлением по последовательному порту и установите соответствующее программное обеспечение.
10. Разработайте и утвердите у руководства правила, в которых четко и кратко описано, что может делать пользователь, а что ему запрещено. Ознакомьте с ними всех сотрудников.
11. Периодически производите инспекцию компьютеров на наличие постороннего программного обеспечения и правильное функционирование штатного программного и аппаратного обеспечения.

Более подробные рекомендации можно прочитать в книгах "UNIX: руководство системного администратора" и "Системное администрирование Linux" *(см. приложение 5)*.

Ссылки

- **www.linuxdocs.org** — Network Administrator's Guide.
- **www.linuxdocs.org** — по этому же адресу расположены и соответствующие HOWTO *(см. гл. 13)*:
  - security-howto;
  - hacker-howto;
  - NFS HOWTO;
  - Firewall-HOWTO.

Часть IV

Основные команды Linux

Глава 13

Помощь

Ни одна мало-мальски приличная программа не обходится без справочного руководства. UNIX-системы обладают, пожалуй, самой обширной и объемной документацией, касающейся функционирования операционной системы и программ.

Традиционно информацию о системе или программе можно получить несколькими путями.

Apropos

Команда `apropos` производит поиск заданного ключевого слова (команды) в базе whatis (см. ниже) и выводит на экран краткое его описание.

Man-справка

В операционной системе Linux справочная система глобальна и использование ее очень просто — в командной строке следует набрать:

```
man имя_программы
```

В результате на текущую консоль будет выведена справочная страница по использованию указанной программы. В принципе, параметром команды `man` необязательно должно быть имя программы, это может быть любой файл операционной системы. В том случае, если для указанного файла не существует справочной страницы, на экран будет выдано соответствующее сообщение.

Если система инсталлирована с языком, отличным от английского, то и справочные страницы (если, конечно, они есть) установятся на языке, который был выбран при инсталляции. К сожалению, для русского языка справочных страниц крайне мало, однако со временем ситуация с локализацией справочных страниц должна быть нормализована.

Программа man сначала пытается найти справочную страницу на языке текущей локали. Если же такой страницы не будет найдено, она выдаст на консоль справочную страницу на английском языке.

Whatis

Команда `whatis` представляет собой мини-справочную систему. В качестве аргумента указывается имя файла, на выходе — строка информации об этом файле.

HOWTO — как сделать

С помощью программы man можно узнать многое, единственное, что она не дает — это алгоритмов решения сложных проблем. Для подобных вопросов существуют так называемые HOWTO — "как сделать что-то". Это достаточно большой набор файлов, предназначенных для решения разнообразных проблем — настройки сети, почтовых программ, Web-серверов, установки различного аппаратного обеспечения и многого другого. Как правило, эти файлы написаны на английском языке, но существуют версии некоторых HOWTO на других языках, в частности, на русском. Список HOWTO приведен *в приложении 2*. Нельзя гарантировать, что данный список полный, но, по крайней мере, это практически все известные HOWTO. Найти их в Интернете не представляет трудности — достаточно ввести наименование искомого HOWTO в строку поиска любой поисковой системы, например **www.rambler.ru**.

Мини-HOWTO

В отличие от HOWTO, решающих достаточно глобальные задачи и имеющих объем порядка пятидесяти-ста страниц, мини-HOWTO решают узкоспециальные задачи и имеют небольшой объем. Список мини-HOWTO приведен *в приложении 3*. Найти их в Интернете также, как и HOWTO, не представляет трудности.

Руководства пользователя Red Hat

На сайте дистрибутива Red Hat существует достаточно большой выбор документации, посвященной инсталляции, работе, настройке операционной системы Linux. Вот только некоторые названия:

- The Official Red Hat Linux Getting Started Guide;
- The Official Red Hat Linux Customization Guide;

- The Official Red Hat Linux x86 Installation Guide;
- The Official Red Hat Linux Reference Guide.

В этих руководствах подробно рассказывается об инсталляции, работе, решениях возникающих проблем в дистрибутиве Red Hat Linux. К сожалению, вся документация на английском языке.

Ссылки

- **www.linuxdocs.org** — одно из полнейших собраний документации о Linux. Ресурс англоязычный, зато почти все, что касается Linux, здесь можно тем или иным образом найти.
- **www.redhat.com** — сайт фирмы Red Hat, производителя одноименного дистрибутива. Одним из достоинств данного дистрибутива является его хорошая поддержка, начиная с версии 4.*x* (создан в 1995 году) и заканчивая текущим. Дистрибутив Red Hat получил очень широкое распространение, его уже начали сравнивать с Windows. В настоящее время Red Hat стал стандартом de-facto для производителей коммерческого программного обеспечения и компьютерного оборудования. Дистрибутив существует в трех вариантах: базовый (доступен для скачивания через Интернет), Professional и Advanced Server. Кроме собственно дистрибутива на сайте присутствует и достаточно большое количество качественно написанной документации (на английском языке).

Глава 14

Справочник наиболее часто употребляемых команд

Эта глава посвящена консольным командам и утилитам. Конечно, после продолжительной работы в графической среде, нет особого желания возвращаться в текстовую консоль. Однако не всегда разумно пользоваться X Window там, где достаточно набрать всего три буквы. Да и не везде будет возможность (желание, необходимость) устанавливать и использовать X Window. Нет никакой необходимости занимать на сервере лишние десятки мегабайт и расходовать драгоценную оперативную память и время процессора (камень в огород Windows NT Server, Windows 2000) на обслуживание графической оболочки. Действительно, зачем серверу (конечно кроме сервера приложений X Window) иметь графическую оболочку, если на нем выполняется сервер баз данных, Web-сервер, почтовый сервер или сервер новостей? Этого не требуется и для администрирования. Практически любое приложение имеет понятный, самодокументированный (или имеющий понятное описание в документации) *текстовый* файл конфигурации. Отредактировать конфигурационные файлы можно либо с текстовой консоли, либо даже удаленно (подключиться к компьютеру, находясь хоть на другом континенте). Для многих приложений существуют также утилиты конфигурирования, имеющие текстовый (а иногда и графический) интерфейс. Многие приложения имеют инструменты удаленного администрирования, использующие Web-интерфейс. Так что наличие консольных команд и утилит для сервера вполне обосновано.

То же и для обычных клиентских машин. Казалось бы, поскольку большинство пользователей безвылазно сидят в X Window, им вообще не надо знать о консоли. Однако это совершенно не так. Как мы уже знаем из *главы 6*, загрузка операционной системы Linux происходит в текстовом (консольном) режиме и лишь на последнем этапе (и то, если это было определено при конфигурации системы) происходит переключение в X Window. Даже из-за этого уже стоит ознакомиться с консольными утилитами.

И еще. Идеология утилит Linux подразумевает модульность, доведенную до совершенства. В распоряжении пользователя огромное количество утилит,

идеально выполняющих какую-то одну конкретную операцию. Из-за узкой специализации утилиты получаются очень маленькими по размеру и, как следствие, — идеально отлаженными.

Замечание

Конечно, пользователю Windows это кажется неудобным. Для утилит Windows принят другой подход — "все в одном" (all in one). В результате — объемный комплекс, в котором достаточно трудно разобраться, который из-за его размеров крайне трудно отладить и который потребляет значительное количество оперативной памяти и процессорного времени.

Кроме того, *все* утилиты Linux (в отличие от Windows-утилит) способны взаимодействовать друг с другом. Это значит, что с помощью утилит можно организовать цепочку взаимосвязанных операций. А с помощью скриптов командной оболочки создать инструментарий для выполнения часто используемых последовательностей операций. Linux требует от пользователя достаточно обширных знаний и некоторого размышления (планирования) перед выполнением нетривиальных действий. С одной стороны, после Windows с ее достаточно бездумным нажиманием кнопок мыши, это несколько напрягает. Но с другой стороны, это позволяет точно, шаг за шагом, понять процесс получения нужного результата, да, заодно, и память потренировать.

Стандартный ввод/вывод, перенаправление

Во многих операционных системах существует понятие стандартного устройства ввода, стандартного устройства вывода и стандартного устройства отображения ошибок. Эти устройства можно задать самостоятельно. По умолчанию используется клавиатура и терминал. Концепция стандартного ввода/вывода очень удобна для автоматизации ввода/вывода.

Для перенаправления стандартного ввода используется символ перенаправления <. Пример использования перенаправления ввода:

```
mysql <2.sql
```

Передает программе mysql данные, содержащиеся в файле 2.sql.

Для перенаправления стандартного вывода используются символы перенаправления > и >>. В чем их отличие? Символ > не проверяет наличие файла, в который сохраняется стандартный вывод программы. Если такой файл существует, то его содержимое полностью заменяется выводом программы. Символ >> проверяет существование файла, и если он существует, то вывод программы дописывается в конец существующего файла.

Примеры:

```
df > 1.txt
ls -A >>1.txt
```

Операции перенаправления ввода/вывода можно использовать одновременно.

Конвейер (поток)

Конвейер (поток, pipe) используется для объединения нескольких команд в одной операции. Обозначается символом `|`. Применяется для передачи стандартного вывода одной программы на стандартный ввод другой программы. В одной командной строке можно использовать несколько операций конвейера.

Пример:

```
ls | grep
```

Команды

Операционная система Linux очень многое наследует от UNIX, в том числе и большую часть команд и утилит. Конечно, эти команды адаптированы и усовершенствованы, но, тем не менее, в целом они сохранили синтаксис соответствующих команд UNIX. Поэтому, по большому счету, не важно, в чем вы работаете: в UNIX или в Linux — система команд на 98% совпадает. Далее мы рассмотрим наиболее часто используемые команды и утилиты. Обратите внимание — наиболее часто используемые. Это означает, что в данной главе упоминаются далеко не все команды и утилиты. Никакая самая объемистая книга не отменяет команду `man`, документацию и файлы HOWTO. За время написания любой книги выходят новые версии программ, и зачастую их возможности кардинально изменяются. Эта глава является обзорной, поэтому в описании команд не всегда (или не в полном объеме) приводятся ключи и параметры вызова.

Дата, время

cal

Команда `cal` выводит на консоль календарь. Если не указаны параметры — выводится календарь на текущий месяц. Если указывается месяц и год — выводится календарь на соответствующий месяц, а если указывается только год — выводится календарь на соответствующий год.

Пример:

```
cal
    Октябрь 2001
```

```
Вс Пн Вт Ср Чт Пт Сб
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

date

Команда `date` выводит текущие дату и время в указанном формате. Так же эта команда позволяет изменять системные дату и время.

Параметры:

- `+` — формат отображения времени и даты в указанном формате;
- `-s` — установка времени и даты;
- `-u` — вывод времени и даты по Гринвичу.

При установке даты и времени можно указывать их как в числовом, так и в нечисловом формате. При использовании числового формата строка данных представляется в следующем виде:

```
MMddhhmmyy
```

где:

- `MM` — месяц;
- `dd` — день;
- `hh` — часы;
- `mm` — минуты;
- `yy` — две последние цифры года.

Пример:

```
date
Сбт Окт  6 19:57:30 EEST 2001
```

Более подробную информацию можно получить по команде `man date`.

Файлы и каталоги

В этом разделе представлены команды и утилиты, которые напрямую взаимодействуют с файлами и каталогами.

Административные команды

Здесь собраны команды, которые отвечают за "административную работу" с файлами и каталогами.

chgrp

Команда `chgrp` изменяет группу каждого заданного файла на группу, которая может быть представлена как именем группы, так и ее числовым идентификатором (GID).

Более подробную информацию можно получить по команде `man chgrp`.

chmod

Команда `chmod` изменяет права доступа файла в соответствии с правами доступа, указанными в параметре, который может быть представлен как в символьном виде, так и в виде восьмеричного числа.

Формат символьного режима:

```
[ugoa...][[+-=][rwxXstugo...]...][,...]
```

Здесь каждый аргумент — это список символьных команд изменения прав доступа, разделенных запятыми. Каждая такая команда начинается с какой-нибудь из букв `ugoa` (впрочем, букв может вообще не быть) или их комбинации, которая указывает, чьи права доступа к файлу будут изменены:

- `u` — владельца;
- `g` — группы;
- `o` — других пользователей, не входящих в данную группу;
- `a` — всех пользователей. Буква `a` эквивалентна `ugo`. В том случае, если не задана ни одна буква, то будет использоваться буква `a`;
- `+` — добавляет выбранные права доступа к уже имеющимся;
- `-` — удаляет эти права;
- `=` — присваивает только эти права файлу.

Буквы `rwxXstugo` выбирают новые права доступа для пользователя, заданного одной из букв `ugoa`:

- `r` — чтение;
- `w` — запись;
- `x` — выполнение;
- `X` — выполнение, если файл является каталогом или уже имеет право на выполнение для какого-нибудь пользователя;
- `S` — `setuid`- или `setgid`-бит;
- `t` — `sticky`-бит;
- `u` — установка для остальных таких же прав доступа, какие имеет пользователь, владеющий этим файлом;

- `g` — установка для остальных таких же прав доступа, какие имеет группа файла;
- `o` — установка для остальных таких же прав доступа, какие имеют остальные пользователи.

Установка `sticky`-бита для каталога приводит к тому, что только владелец файла и владелец этого каталога могут удалять файл из каталога.

В операционной среде Linux, если на файле установлен бит `setgid`, но не установлен бит выполнения группой, то блокировки этого файла становятся жесткими (mandatory), в отличие от обычных — информационных (advisory). Подробная информация по этому вопросу находится в файле /usr/src/linux/Documentation/mandatory.txt.

Числовой режим состоит из четырех восьмеричных цифр, которые складываются из битовых масок 4, 2 и 1. Любые пропущенные разряды дополняются лидирующими нулями:

- первая цифра выбирает установку идентификатора пользователя — `setuid` (4), идентификатора группы — `setgid` (2) или `sticky`-бита (1);
- вторая цифра выбирает права доступа для пользователя, владеющего данным файлом: чтение (4), запись (2) и выполнение (1);
- третья цифра выбирает права доступа для пользователей, входящих в группу;
- четвертая цифра выбирает права доступа для остальных пользователей, не входящих в группу.

Эту команду может применять либо владелец файла, либо пользователь root.

Более подробную информацию можно получить по команде `man chmod`.

chown

Команда `chown` изменяет владельца и/или группу для заданного файла.

В качестве имени владельца/группы берется первый аргумент, не являющийся опцией. Если задано только имя пользователя (или его числовой идентификатор), то данный пользователь становится владельцем каждого из указанных файлов, а группа этих файлов не изменяется. Если за именем пользователя через двоеточие следует имя группы (или числовой идентификатор группы) без пробелов между ними, то изменяется также и группа файла. Если двоеточие стоит за именем пользователя, но группа не задана, то данный пользователь становится владельцем указанных файлов, а группа указанных файлов изменяется на основную группу пользователя. Если опущено имя пользователя, а двоеточие или точка вместе с группой заданы, то будет изменена только группа указанных файлов.

Как и предыдущие команды, ее может применять либо владелец файла, либо пользователь root.

chroot

Команда `chroot` используется только пользователем root, который с помощью команды

```
chroot имя_каталога
```

делает каталог корневым каталогом. Эта команда используется администратором для повышения безопасности системы.

Более подробную информацию можно получить по команде `man chroot`.

lockfile

Команда `lockfile` используется для создания специальных семафорных файлов.

mknod

Команда `mknod` создает именованный канал (FIFO), специальный символьный или специальный блочный файл (файл устройства).

Специальный файл именуется с помощью тройки параметров: один логический и два целых. Логический параметр говорит о том, является ли специальный файл символьным или блочным. Два целых параметра задают старший и младший номера устройства.

Специальный файл практически не занимает места на диске и используется только для общения с операционной системой, а не для хранения данных. Часто специальные файлы указывают на аппаратные устройства или на службы операционной системы.

Специальные блочные файлы обычно являются устройствами, подобными диску. Все другие устройства являются специальными символьными файлами.

Аргумент, следующий за именем, задает тип файла, который нужно создать:

- `p` — для FIFO;
- `b` — для блочного специального файла;
- `c` — для символьного специального файла.

В файле /usr/src/linux/Documentation/devices.tex находится список устройств, где есть имена устройства, тип, старший и младший номер.

Более подробную информацию можно получить по команде `man mknod`.

Общие команды

В этом разделе собраны команды, тем или иным способом воздействующие на файлы и каталоги.

cat

Команда выводит на экран содержимое файла, начиная с первой строки.

cd

Команда `cd` является встроенной в bash командой, которая предназначена для смены текущего каталога.

Пример:

```
cd /var/log
```

Делает текущим каталог /var/log.

cp

Команда `cp` копирует файлы или каталоги. Если последний аргумент является существующим каталогом, то команда `cp` копирует каждый файл в этот каталог. В случае, если задано только два имени файла, то команда `cp` копирует первый файл во второй.

Права доступа к файлам и каталогам будут равны тем, что были на оригинальных файлах, но биты `sticky`, `setuid` и `setgid` будут сброшены.

Пример:

```
cp /home/user1/test /home/user2/1.txt
```

Копирует файл /home/user1/test в файл /home/user2/1.txt.

Более подробную информацию можно получить по команде `man cp`.

dir

См. команду `ls`.

file

Команда `file` определяет тип (или принадлежность к определенному процессу) файла. Иногда для этих целей используется файл /usr/share/magic.

Пример:

```
file file.c
       file.c:  C program text

file -s /dev/hda{,1,2,3,4,5,6,7,8,9,10}
       /dev/hda:   x86 boot sector
       /dev/hda1:  Linux/i386 ext2 filesystem
       /dev/hda2:  x86 boot sector
       /dev/hda3:  x86 boot sector, extended partition table
       /dev/hda4:  Linux/i386 ext2 filesystem
       /dev/hda5:  Linux/i386 swap file
       /dev/hda6:  Linux/i386 swap file
       /dev/hda7:  Linux/i386 swap file
```

```
/dev/hda8:  Linux/i386 swap file
/dev/hda9:  empty
/dev/hda10: empty
```

Более подробную информацию можно получить по команде `man file`.

find

Команда `find` осуществляет поиск файлов. Имеет большое количество параметров, позволяющих ей производить как простой поиск, так и поиск со многими условиями.

Более подробную информацию можно получить по команде `man find`.

head

Команда `head` выводит на экран первые 10 строк файла. С помощью параметров можно изменить размер выводимой части.

ln

Команда `ln` создает ссылки на файлы. По умолчанию создаются жесткие ссылки, а при указании опции `-s` делаются символические ссылки.

Если задан только один файл, то для него делается ссылка в текущем каталоге с таким же именем, как у этого файла. В противном случае, если последний аргумент является именем существующего каталога, то команда `ln` создаст ссылки в этом каталоге для каждого из файлов с такими же именами, как и у исходных файлов. В случае, если задано два имени, то создается ссылка (второе имя) на файл (первое имя).

По умолчанию команда `ln` не удаляет существующие файлы или существующие символьные ссылки.

Пример:

```
ln make test
```

Создает жесткую ссылку с именем test на файл make.

Более подробную информацию можно получить по команде `man ln`.

locate

См. команду `slocate`.

ls

Команда `ls` выводит содержимое каталога. Эта команда сначала выводит список всех файлов, перечисленных в командной строке, а затем выводит список всех файлов, находящихся в каталогах, перечисленных в командной строке. Если не указано ни одного файла, то по умолчанию аргументом назначается текущий каталог.

Каждый список файлов сортируется отдельно в алфавитной последовательности текущих региональных настроек (locale).

Результат выдается на стандартный вывод, по одному файлу на строку.

При использовании ключа `-l` информация выдается в следующем виде:

- тип файла;
- права доступа к файлу;
- количество ссылок на файл;
- имя владельца;
- имя группы;
- размер файла;
- временной штамп;
- имя файла.

Типы файлов могут принимать следующие значения:

- `-` — для обычного файла;
- `d` — для каталога;
- `b` — для блочного устройства;
- `c` — для символьного устройства;
- `l` — для символической ссылки;
- `p` — для FIFO;
- `s` — для сокета.

Пример:

```
ls -l
итого 124
-rw-rw-r--    1 alst     alst          665 Окт  6 16:09 cd
-rw-rw-r--    1 alst     alst          665 Окт  6 16:09 cdd
-rw-rw-r--    1 alst     alst         4005 Окт  6 16:08 chgrp
-rw-rw-r--    1 alst     alst         6909 Окт  6 16:08 chmod
-rw-rw-r--    1 alst     alst         3668 Окт  6 16:08 chown
-rw-rw-r--    1 alst     alst         1126 Окт  6 16:08 chroot
-rw-rw-r--    1 alst     alst        12508 Окт  6 16:10 cp
drwxr-xr-x    2 alst     alst         4096 Авг 31 10:29 Desktop
-rw-rw-r--    1 alst     alst        16011 Окт  6 16:10 file
-rw-rw-r--    1 alst     alst        17248 Окт  6 16:10 find
-rw-rw-r--    1 alst     alst         8497 Окт  6 16:10 ln
-rw-rw-r--    1 alst     alst         2550 Окт  6 16:11 locate
```

```
-rw-rw-r--    1 alst     alst         7228 Окт  6 16:09 locfile
-rw-rw-r--    1 alst     alst            0 Окт  6 16:11 lss
-rw-rw-r--    1 alst     alst         3917 Окт  6 16:09 mknod
drwx------    2 alst     alst         4096 Сен  8 16:03 nsmail
-rw-rw-r--    1 alst     alst          978 Окт  6 16:11 uptime
-rw-rw-r--    1 alst     alst           62 Окт  6 16:11 uptm
```

Более подробную информацию можно получить по команде `man ls`.

mc

Команда `mc` запускает на выполнение файловый менеджер Midnight Commander, который позволяет производить множество операций с файлами и каталогами и имеет огромное количество команд и настроек. Исчерпывающую информацию о Midnight Commander можно получить из его справочной системы, вызываемой нажатием клавиши <F1>.

mkdir

Команда создает каталоги с заданными именами. По умолчанию права доступа к каталогам устанавливаются в 0777 за вычетом битов, установленных в umask.

Пример:

```
mkdir test
```

Более подробную информацию можно получить по команде `man mkdir`.

mkfifo

Команда `mkfifo` создает именованные каналы (FIFO) с указанными именами. FIFO — это специальный тип файла, который позволяет общаться независимым процессам. Один процесс открывает FIFO-файл для записи, а второй для чтения, после чего данные могут передаваться как в обычных неименованных каналах в `shell`.

Более подробную информацию можно получить по команде `man mkfifo`.

mv

Команда `mv` перемещает или переименовывает файлы или каталоги.

Если последний аргумент является именем существующего каталога, то команда `mv` перемещает все указанные файлы в этот каталог. Если задано два файла, то имя первого файла будет изменено на имя второго.

Пример:

```
mv /tmp/test /home/user1
```

Перемещает файл test из каталога /tmp в каталог /home/user1.

Более подробную информацию можно получить по команде `man mv`.

pwd

Команда `pwd` выводит имя текущего каталога.

Пример:

```
pwd
/home/alst
```

rm

Команда `rm` удаляет файлы или каталоги. По умолчанию каталоги не удаляются, но если заданы опции `-r` или `-R`, то будет удаляться все дерево вложенных каталогов.

Пример:

```
rm *.tmp
```

Удаляет все TMP-файлы из текущего каталога.

Более подробную информацию можно получить по команде `man rm`.

rmdir

Команда `rmdir` удаляет пустые каталоги. Если каталог не пуст, то будет выдано сообщение об ошибке. Для удаления непустых каталогов используйте команду

```
rmdir -r
```

size

Команда `size` выводит размеры сегментов программы, указанной в командной строке.

Пример:

```
size /sbin/agetty
 text      data      bss      dec      hex   filename
 10819      844    10336    21999      55ef  agetty
```

slocate

Команда `slocate` — это более защищенный вариант команды `locate`. Команда `locate` производит быстрый поиск в базе данных имен файлов системы.

Пример:

```
locate dir

/var/run/runlevel.dir
/var/www/icons/dir.gif
/var/www/icons/small/dir.gif
```

```
/var/www/icons/small/dir2.gif
/etc/X11/applnk/Games/xpuzzles/.directory
/etc/X11/xdm/authdir
...
/usr/src/linux-2.4.3/net/tux/redirect.c
/bin/mkdir
/bin/rmdir
/home/alst/.kde/Autostart/.directory
/home/alst/Desktop/.directory
/lib/security/pam_mkhomedir.so
/root/.kpackage/dir
```

Более подробную информацию можно получить по команде `man slocate`.

split

Команда `split` предназначена для разбиения файла на несколько частей. По умолчанию создаются части размером в 1000 строк.

stat

Команда `stat` показывает информацию о файле или файлах, заданных в командной строке.

Пример:

```
stat /sbin/agetty

  File: "agetty"
  Size: 13148      Blocks: 32        Regular File
Access: (0755/-rwxr-xr-x)     Uid: ( 0/ root)  Gid: ( 0/ root)
Device: 302        Inode: 350883     Links: 1
Access: Sat Oct  6 20:10:19 2001
Modify: Fri Jul 13 01:22:17 2001
Change: Fri Aug 31 07:44:08 2001
```

Более подробную информацию можно получить по команде `man stat`.

tac

Команда `tac` выводит содержимое файла в обратном порядке, от первой строки к последней.

tail

Команда `tail` выводит на экран последние 10 строк файла. С помощью параметров можно изменить размер выводимой части.

vdir

См. описание команды `ls`.

Сеть

dig

Эта команда используется для формирования запросов о доменах DNS-серверам. Имеет большое количество управляющих параметров, информация о которых содержится в соответствующей документации.

elm

Команда `elm` является интерактивной почтовой программой, имеющей больше возможностей, чем mail.

finger

Команда `finger` получает информацию о пользователе, а также содержимое его файлов .plan и .project. Можно указать пользователя, задав его системный идентификатор, имя или фамилию. Обычно с целью увеличения безопасности системы администраторы не устанавливают на своих системах finger-серверы.

ftp

Команда `ftp` позволяет соединиться с удаленной системой с использованием протокола FTP. После установки соединения можно копировать файлы между локальной и удаленной системами, удалять файлы и просматривать каталоги, если имеются соответствующие права доступа к удаленной системе.

Эта команда — самый простой FTP-клиент. Для эффективной работы необходимо знать команды FTP-протокола. В современных системах используют либо графические FTP-клиенты, либо текстовые с удобным интерфейсом (например, встроенный в `mc`).

getty (mgetty)

Команда позволяет модему осуществлять исходящие звонки и принимать входящие. Имеет гибкую конфигурацию. Более подробную информацию можно получить по соответствующей команде `man`.

host

Команда `host` выводит IP-адрес указанной системы, используя службу DNS. Можно указать IP-адрес, и он будет преобразован в имя системы.

hostname

Команда `hostname` выводит имя локальной системы. Привилегированный пользователь может использовать ее для задания системе нового имени.

ipchains

Команда `ipchains` используется для создания, сопровождения и проверки правил IP-брандмауэра (firewall) ядра операционной системы Linux. Эти правила подразделяются на четыре вида:

- входящие правила;
- исходящие правила;
- правила IP-маршрутизации (forwarding);
- пользовательские правила.

Для каждого из этих правил создается отдельная таблица правил. Несколько устарела, сейчас используется `iptables`.

Более подробную информацию можно получить по команде `man ipchains` и из соответствующей литературы.

iptables

Команда `iptables` используется для создания, сопровождения и проверки правил IP-брандмауэра (firewall) ядра операционной системы Linux. Эта команда сегодня заменила команду `ipchains`.

Более подробную информацию можно получить по команде `man iptables` и из соответствующей литературы.

kppp

Программа, входящая в состав KDE. Позволяет максимально легко настроить модемное PPP-соединение с провайдером. Понятный и удобный графический интерфейс.

lynx

Команда `lynx` запускает текстовый браузер Интернета. Он не позволяет выводить графические изображения или различные шрифты. Обычно используется для просмотра Web-документов (в формате HTML), находящихся локально на компьютере.

mail

Команда `mail` используется для чтения и отправки электронной почты. Простой аскетичный интерфейс. Занимает мало места на жестком диске. Обычно не используется.

mimencode

Команда `mimencode` позволяет производить кодировку и раскодировку MIME — формата сообщений электронной почты.

minicom

Программа minicom предназначена для интерактивной работы с модемами — организации соединения, настройки последовательного порта, чата и т. п. Полезна для отладки модемного соединения. Имеет богатые возможности по автоматизации.

netcfg

Утилита netcfg является частью пакета linuxconf. Эта программа с помощью текстового меню позволяет быстро и просто произвести конфигурацию параметров системы, тем или иным образом относящихся к сетевым настройкам.

netstat

Выводит информацию о сетевых соединениях, таблицы маршрутизации, статистику по сетевым интерфейсам и т. п. Используется для отладки и мониторинга сети.

nslookup

Утилита используется для получения различной информации от DNS-серверов. Обычно применяется для получения расширенной информации о хосте либо для обнаружения и устранения неполадок в конфигурации сети.

pine

Текстовая программа pine используется для чтения и отправки электронной почты и новостей Usenet. Она поддерживает MIME-кодирование и позволяет отправлять письма с MIME-содержанием. Имеет много функциональных возможностей, удобный интерфейс.

ping

Команда `ping` используется для отправки ICMP-пакетов ECHO_REQUEST на указанную систему (IP-адрес или символическое имя системы) для определения прохождения пакетов. Простое, но незаменимое средство диагностики сети.

```
ping
PING 127.0.0.1 (127.0.0.1) from 127.0.0.1 : 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=214 usec
```

```
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=69 usec
64 bytes from 127.0.0.1: icmp_seq=2 ttl=255 time=29 usec
64 bytes from 127.0.0.1: icmp_seq=3 ttl=255 time=30 usec

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.029/0.085/0.214/0.076 ms
```

Более подробную информацию можно получить по команде `man ping`.

procmail

Команда `procmail` используется для сортировки приходящей почты. Для ее вызова обычно применяется механизм перенаправления почты при помощи файла .forward. Кроме того, `procmail` можно настроить для работы в сочетании с почтовой программой.

SSH (OpenSSH)

Команда `SSH` (secure shall) призвана заменить устаревшие и небезопасные команды `telnet` и r-команды. При работе создает шифрованный туннель между системой, к которой подключаются, и подключаемой системой.

telnet

Команда `telnet` позволяет установить соединение с удаленной системой с использованием протокола Telnet. Считается небезопасной, поскольку пересылает по сети логин и пароль в незашифрованном виде. Сегодня рекомендуется использование SSH или OpenSSH.

traceroute

Команда `traceroute` используется для определения маршрута следования пакетов от вашего хоста до указанного вами хоста. Применяется как в целях отладки маршрутизации (если в обслуживании большая группа сетей), так и в познавательных целях, например, чтобы определить, почему внутри одного города между провайдерами так долго проходят пакеты. Оказалось, пакеты передавались не через внутреннюю точку обмена трафиком, а через город на другом континенте.

Более подробную информацию можно получить по команде `man traceroute`.

uudecode

Команда `uudecode` выполняет UU-декодирование файла из формата, подходящего для пересылки по электронной почте (для кодирования используются только цифры и латинские символы).

uuencode

Команда `uuencode` выполняет UU-кодирование файла в формат, подходящий для пересылки по электронной почте (для кодирования используются только цифры и латинские символы).

wget

Программа wget используется для загрузки файлов по протоколу HTTP и включена во все основные дистрибутивы. Также может обрабатывать FTP, временные метки (date stamps), рекурсивно отражать полное дерево каталогов Web-сайта и др.

Кроме того, wget позволяет возобновлять прерванное задание, если задан незавершенный файл, к которому добавляются оставшиеся данные.

Очень мощная программа, полную информацию по которой можно получить из соответствующей документации.

Администрирование

at

Команда `at` позволяет однократно запустить на выполнение команду или группу команд в назначенное время. Эти команды не должны требовать ввода информации с консоли. Как правило, такие команды используются для архивации данных, создания резервной копии данных и т. п.

Более подробную информацию можно получить по команде `man at`. Рекомендуется так же `man crontab`.

atq

Команда `atq` выводит список всех заданий, поставленных на выполнение командой `at`.

atrm

Команда `atrm` позволяет удалить задания из очереди команды `at`.

batch

Команда `batch`, подобно команде `at`, также позволяет выполнять задания, но, в отличие от команды `at`, не существует четкого времени выполнения заданий. Вместо этого критерием запуска команд является минимальная загрузка операционной системы.

cksum

Команда `cksum` вычисляет контрольную сумму (CRC) указанных файлов.

crond

Используется для автоматического запуска программ в указанное время. Использует файлы crontab.

crontab

Программа, позволяющая просматривать и редактировать файлы crontab.

getkeycodes

Команда `getkeycodes` выводит таблицу соответствия скан-кодов кодам клавиш.

ifconfig

Используется для конфигурации сетевых интерфейсов. Обычно необходима для отладки сетевых соединений. При вызове без параметров команда отображает статус активных интерфейсов.

Пример:

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0


ppp0      Link encap:Point-to-Point Protocol
          inet addr:195.114.131.239  P-t-P:195.114.128.4
Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:174301 errors:31 dropped:0 overruns:0 frame:0
          TX packets:98860 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
```

insmod

Утилита insmod используется для добавления модулей в ядро без его перекомпиляции. Рекомендуется при подключении нового модуля предварительно воспользоваться программой modprobe.

isapnp

Утилита считывает конфигурационный файл, в котором определены настройки устройств ISA PnP и конфигурирует их в ядре операционной системы Linux.

kill

Команда `kill` отправляет процессу с указанным идентификатором сигнал. Часто используется для завершения работы процессов. Только владелец процесса или привилегированный пользователь могут использовать эту команду.

killall

Команда `killall` завершает работу всех процессов с указанным именем.

lilo

`lilo` — системный загрузчик, который производит загрузку операционных систем, в частности Linux. Так же применяется для собственного администрирования.

Использует следующие параметры:

- `-v` — повысить уровень информативности;
- `-q` — показать текущую карту загрузки. `lilo` сохраняет в файле /boot/map имена и расположение на диске ядер, предназначенных для загрузки;
- `-m файл-карты` — использовать карту загрузки, находящуюся в указанном файле;
- `-C файл-настроек` — `lilo` читает настройки из файла настроек /etc/lilo.conf. Этот параметр позволяет указать другой файл в качестве файла настроек;
- `-d` — задержка. Если в настройках `lilo` указано несколько ядер, и во время загрузки нажата клавиша <Shift>, загрузчик спросит, какое из ядер нужно загрузить. После указанной задержки будет загружено первое ядро из списка. Этот параметр указывает время задержки в десятых долях секунды;
- `-D` — метка. Использовать по умолчанию ядро с указанной меткой, а не первое ядро, указанное в файле настроек;
- `-r каталог` — сделать указанный каталог корневым. Используется для восстановления записи в MBR;
- `-t` — тестирование конфигурации. Не записывает на диск новый загрузочный сектор и файл карты. Использовать вместе с параметром `-v`, чтобы узнать, что собирается сделать `lilo`;
- `-c` — разрешает уплотнение карты. Запросы на чтение к смежным секторам будут объединяться. Ускоряет загрузку;
- `-f disk-tab` — определяет, в каком файле будет храниться геометрия диска (по умолчанию /etc/disktab);

- `-i загрузочный_сектор` — определяет файл, который будет использован как новый загрузочный сектор. (По умолчанию /boot/boot.b);
- `-l` — вместо адресов типа сектор/головка/цилиндр `lilo` будет генерировать линейные адреса секторов;
- `-P {fix|ignore}` — исправлять `fix` или игнорировать `ignore` поврежденные таблицы разделов;
- `-s резервный_файл` — когда `lilo` переписывает загрузочный сектор, его старое содержимое помещается в специальный резервный файл — /boot/boot.NNNN, где NNNN зависит от используемого устройства. Этот параметр позволяет задать другой резервный файл для загрузочного сектора. Или при использовании вместе с параметром `-u`, определяет, откуда восстанавливать загрузочный сектор;
- `-S резервный_файл` — обычно `lilo` не переписывает существующий резервный файл. Этот параметр разрешает перезапись;
- `-u имя_устройства` — деинсталлировать `lilo` путем копирования резервного файла назад в загрузочный сектор;
- `-U имя_устройства` — то же самое, но без проверки времени сохранения;
- `-R командная_строка` — этот параметр устанавливает загрузчику команду, которая выполнится при следующей загрузке. Сразу после загрузки операционной системы загрузчик стирает эту строку. Обычно она используется в сценариях при перезагрузке;
- `-I метка` — имя используемого ядра после загрузки помещается в переменную окружения `BOOT_IMAGE`. Эта команда выведет путь к файлу, соответствующему указанной метке, на стандартное устройство;
- `-V` — напечатать номер версии.

Более подробную информацию можно получить по команде `man lilo.cohf`.

linuxconf

Эта программа используется для конфигурирования и администрирования системы. В ней собраны практически все системные настройки. Обладает удобным текстовым (и графическим) интерфейсом. Однако использование ее нежелательно — она не всегда корректно выставляет права доступа на важные файлы.

md5sum

Команда `md5sum` рассчитывает контрольную сумму файла или строки по алгоритму md5. Рекомендуется для проверки целостности файлов, полученных через Интернет, и для организации безопасности файловой системы (как один из элементов контроля целостности системы).

Более подробную информацию можно получить по команде `man md5sum`.

modprobe

Команда `modprobe` предназначена для загрузки и тестирования загружаемых модулей ядра. Обычно используется при установке и конфигурировании нестандартного оборудования.

Более подробную информацию можно получить по команде `man modprobe`.

mount

Команда `mount` предназначена для монтирования файловых систем. Используется пользователем root. В качестве аргументов принимает файл устройства и точку монтирования. Использует большое количество дополнительных параметров.

Пример:

```
mount /dev/fd0 /mnt/floppy
```

Более подробную информацию можно получить по команде `man mount`.

nice

Команда `nice` позволяет изменить приоритет запускаемой команды. Значение приоритета может быть выбрано от 15 (низший приоритет) до −20 (высший приоритет). Значения приоритета меньше нуля могут устанавливаться только пользователем root.

При запуске без параметров команда `nice` выводит значение приоритета по умолчанию.

passwd

Эта команда позволяет изменить пароль, который используется для входа в систему. Если запускается `passwd` без параметров, то система запросит старый пароль, а затем новый пароль.

Пользователь root может изменить пароль любого пользователя системы, указав его системный идентификатор (логин) и новый пароль в командной строке.

pnpdump

Команда `pnpdump` используется совместно с `isapnp` для конфигурирования устройств ISA PnP. Она сканирует все ISA-платы и выводит используемые ими ресурсы на стандартный вывод. На основании полученных данных можно правильно сконфигурировать устройства ISA PnP.

renice

Эта команда изменяет приоритет одного или нескольких запущенных процессов.

rpm

`rpm` — менеджер пакетов, который используется для сборки, установки, инспекции, проверки, обновления и удаления программных пакетов.

Может быть выбран один из следующих основных режимов:

- инициализация базы данных;
- перестроение базы данных;
- сборка пакетов;
- рекомпиляция пакетов;
- сборка пакетов из TAR-архивов;
- запрос;
- показ полей запроса;
- установка;
- обновление;
- удаление;
- верификация;
- проверка подписи;
- повторная подпись;
- добавление подписи;
- установка владельцев и групп;
- показ конфигурации.

Подробную информацию по `rpm` см. в *гл. 8*.

rmmod

Команда `rmmod` удаляет загружаемые модули из ядра, если они не используются ядром или другими модулями.

Более подробную информацию можно получить по команде `man rmmod`.

setserial

Команда `setserial` позволяет получить или установить настройки последовательного порта.

Пример:

```
setserial ttyS0
```

Более подробную информацию можно получить по команде `man setserial`.

setterm

Команда `setterm` позволяет установить атрибуты терминала. Как правило, этой командой пользоваться не придется.

Более подробную информацию можно получить по команде `man setterm`.

skill

Команда `skill` отправляет сигналы или изменяет приоритет указанного процесса. По умолчанию отправляется сигнал `TERM`.

snice

Команда `snice` позволяет изменить приоритет запущенного процесса. По умолчанию новый приоритет равен +4. Приоритет может быть задан явно в виде параметров `+приоритет` или `-приоритет`.

strace

Команда `strace` используется для трассировки системных вызовов и сигналов. Используется для запуска определенной программы. После этого `strace` будет производить трассировку системных вызовов и сигналов соответствующих программе процессов. Информация выводится на экран или сохраняется в файле. Команда `strace` обычно используется для отладки программ.

stty

Команда `stty` позволяет выводить и изменять настройки терминала. Без параметров выводит установки терминала, на котором она запущена.

Пример:

```
stty

speed 0 baud; line = 0;
-brkint -imaxbel
```

umount

Команда `umount` используется для размонтирования файловых систем. В качестве аргумента использует точку монтирования файловой системы или файл устройства.

Пример:

```
umount /mnt/floppy
```

Более подробную информацию можно получить по команде `man umount`.

useradd

Утилита позволяет создавать в операционной системе нового пользователя. При вводе пользователя ему можно задать группу, пароль и некоторые другие параметры.

xf86config

Команда `xf86config` создает файл конфигурации Xf86config, используемый X-сервером.

xvidtune

Команда `xvidtune` позволяет произвести точную настройку видеорежимов X-сервера. При задании параметров в командной строке команда `xvidtune` может использоваться для переключения видеорежимов, а также установки времени отключения мониторов, имеющих расширенное управление питанием. Команда `xvidtune` без параметров открывает диалоговое окно, дающее возможность выполнять настройку видеорежима. Полученные режимы могут быть описаны в виде, позволяющем вставить их в файл `Xf86Config`.

zic

Утилита, позволяющая компилировать бинарный файл временной зоны. Использует в качестве источника информации текстовый файл описания временной зоны. Практически никогда не используется администратором системы, поскольку в каталоге /usr/share/zoneinfo/ находятся скомпилированные файлы временных зон на все случаи жизни.

Состояние системы

df

Команда `df` выдает информацию о доступном и используемом дисковом пространстве на файловых системах.

Вызов команды без аргументов выдает отчет для всех файловых систем, которые смонтированы в данный момент. По умолчанию все размеры выдаются в блоках по 1024 байт, за исключением случая, когда установлена переменная `POSIXLY_CORRECT`. В этом случае размер блока соответствует POSIX-версии.

Пример:

```
df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda2              4134932   1607188   2317696  41% /
/dev/hda1              4008372   1085892   2922480  28% /mnt/floppy
```

du

Команда `du` выдает отчет об использовании дискового пространства указанными файлами или каталогами. Под 'использованным дисковым пространством' понимается пространство, занятое под всю иерархию каталогов указанного каталога.

Запущенная без аргументов, команда `du` выдает отчет о дисковом пространстве для текущего каталога.

Размеры используемого дискового пространства указываются в блоках по 1024 байт (если размер не задан посредством опций), за исключением случая, когда задана переменная окружения `POSIXLY_CORRECT`.

Пример выполнения команды в каталоге /root:

```
du
16  ./.gnome/accels
4  ./.gnome/apps
20  ./.gnome/panel.d/default/launchers
52  ./.gnome/panel.d/default
56  ./.gnome/panel.d
4  ./.gnome/nautilus-scripts
8  ./.gnome/gnome-vfs
4  ./.gnome/application-info
168  ./.gnome
.........
16  ./.ee/minis/root
20  ./.ee/minis
32  ./.ee
612  .
```

dumpkey

Команда `dumpkey` выводит информацию о драйвере клавиатуры.

free

Команда `free` выдает информацию об использовании оперативной памяти.

Пример:

```
free
             total       used       free     shared    buffers     cached
Mem:        255532     227600      27932          0      66140      74568
-/+ buffers/cache:      86892     168640
Swap:       257000          0     257000
```

ftpcount

Команда `ftpcount` выдает текущее количество пользователей в каждом классе, подключенных к FTP-серверу, используя определение классов из файла ftpaccess.

ftpwho

Команда `ftpwho` выводит информацию о подключенных к FTP-серверу в данный момент пользователях.

kdb_mode

Эта команда выводит текущий режим драйвера клавиатуры и позволяет изменить его.

last

Команда `last` выводит список последних зарегистрировавшихся в системе пользователей.

Пример:

```
last
alst     tty4                         Sun Nov  4 12:55   still logged in
alst     tty3                         Sun Nov  4 12:55 - 12:56  (00:00)
alst     tty2                         Sun Nov  4 12:54   still logged in

wtmp begins Sun Nov  4 12:54:36 2001
```

ps

Команда `ps` выводит разнообразную информацию о процессах операционной системы.

Пример:

```
ps -A
  PID TTY          TIME CMD
    1 ?        00:00:04 init
    2 ?        00:00:00 keventd
    3 ?        00:00:00 kapm-idled
    4 ?        00:00:00 kswapd
    5 ?        00:00:00 kreclaimd
    6 ?        00:00:00 bdflush
    7 ?        00:00:00 kupdated
    8 ?        00:00:00 mdrecoveryd
    . . . . . . . . . . .
```

```
  741 tty1     00:00:00 login
  742 tty1     00:00:00 bash
  781 tty1     00:00:00 mc
  782 ?        00:00:00 cons.saver
  783 pts/0    00:00:00 bash
  802 tty2     00:00:00 bash
  837 tty2     00:00:00 mc
  838 ?        00:00:00 cons.saver
  839 pts/1    00:00:00 bash
 1292 pts/1    00:00:00 ps
```

Более подробную информацию можно получить по команде `man ps`.

quota

Команда `quota` отображает ограничения на использование дискового пространства пользователями.

Более подробную информацию можно получить по команде `man quota`.

tload

Команда выводит график загрузки системы.

top

Команда `top` выводит список процессов в системе, отсортированных в порядке убывания используемого процессорного времени. Неплохой инструмент для определения подозрительных процессов.

Пример:

```
top

4:19pm up 13 min, 2 users, load average: 0,01, 0,02, 0,00
37 processes: 36 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 1,0% user, 1,0% system, 0,0% nice, 97,8% idle
Mem:319968K av, 50468K used, 269500K free, 0K shrd, 4164K buff
Swap: 216868K av, 0K used, 216868K free, 29524K cached

PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM  TIME COMMAND
 1  root      8   0   544  544   472 S     0,0  0,1   0:04 init
 2  root      8   0     0    0     0 SW    0,0  0,0   0:00 keventd
 3  root      9   0     0    0     0 SW    0,0  0,0   0:00 kapm-idled
 4  root      9   0     0    0     0 SW    0,0  0,0   0:00 kswapd
```

```
 5  root     9   0     0    0     0 SW    0,0  0,0   0:00 kreclaimd
 6  root     9   0     0    0     0 SW    0,0  0,0   0:00 bdflush
 7  root     9   0     0    0     0 SW    0,0  0,0   0:00 kupdated
61  root     9   0     0    0     0 SW    0,0  0,0   0:00 khubd
364 root     9   0   600  600   500 S     0,0  0,1   0:00 syslogd
369 root     9   0  1060 1060   460 S     0,0  0,3   0:00 klogd
383 rpc      9   0   596  596   504 S     0,0  0,1   0:00 portmap
398 rpcuser  9   0   772  772   668 S     0,0  0,2   0:00 rpc.statd
470 root     8   0   532  532   464 S     0,0  0,1   0:00 apmd
519 root     9   0   648  648   544 S     0,0  0,2   0:00 automount
531 daemon   9   0   584  584   508 S     0,0  0,1   0:00 atd
546 root     9   0  1136 1136   948 S     0,0  0,3   0:00 sshd
566 root     9   0   992  992   788 S     0,0  0,3   0:00 xinetd
```

uptime

Команда выводит информацию о системе: количество работающих пользователей, среднюю загрузку системы, время, прошедшее с момента запуска операционной системы.

Пример:

```
uptime
  4:11pm  up 5 min,  2 users,  load average: 0.04, 0.04, 0.01
```

users

Команда `users` выводит информацию о пользователях, подключенных в настоящий момент к системе. Для получения этой информации используется файл /etc/utmp.

who

Команда `who` выводит информацию о системе или о пользователе. Команда без параметров выводит информацию о пользователях, зарегистрированных в системе.

Пример:

```
who
alst     tty1     Oct  6 14:13
root     tty2     Oct  6 14:18
```

Более подробную информацию можно получить по команде `man who`.

w

Команда `w` выводит информацию о системе: список пользователей, подключенных к системе, статистику использования системы, а также выполняемые пользователями задачи. Эта команда является комбинацией команд `who`, `ps`, `-a` и `uptime`.

Создание файловой системы

fdisk

Утилита для создания, изменения и удаления дисковых разделов. Обычно используется во время инсталляции операционной системы или при подключении нового диска.

fdformat

Команда `fdformat` производит низкоуровневое форматирование дискеты.

mkfs

С помощью утилиты mkfs создается файловая система. Обычно используется совместно с утилитой fdisk. При использовании этой утилиты необходимо определить тип файловой системы и количество используемых блоков. Более подробную информацию можно получить по соответствующей команде `man`.

Диагностика файловой системы

fsck

Утилита fsck обычно используется при загрузке операционной системы для проверки и восстановления файловых систем. Более подробную информацию смотрите в документации.

Архивация

gzip

Программа, осуществляющая сжатие файла по алгоритму Лемпела—Зиффа. В отличие от аналогов в MS-DOS или Windows, может сжимать только один файл. Для архивации нескольких файлов в один архив необходимо воспользоваться утилитой tar.

tar

Утилита, предназначенная для изготовления из нескольких файлов/каталогов одного файла архива. При этом компрессия файлов не производится. Первоначально использовалась для записи файлов на ленточный накопитель.

Работа с текстовыми файлами

joe

Команда `joe` запускает текстовый редактор. Простой, гибкий, удобный в использовании.

sort

Команда `sort` сортирует, объединяет или сравнивает строки текстовых файлов. Результат выводится на экран.

uniq

Команда `uniq` удаляет повторяющиеся строки из файла 1 и выводит результат в файл 2.

vi

Команда `vi` запускает текстовый редактор, который является одним из старейших редакторов и установлен практически на всех UNIX-системах. Сегодня обычно не используется оригинальный редактор `vi`. Вместо него вызывается редактор `vim` или редактор `elvis`.

vim

Текстовый редактор, запускаемый по команде `vim`, — это `vi`-совместимый редактор для обработки текстовых файлов. Более "продвинутый", с большей функциональностью и меньшими ограничениями.

Помощь

apropos

Команда `apropos` производит поиск заданного ключевого слова в базе whatis.

man

Команда `man` форматирует и выводит справочные страницы для команд, функций и тому подобных вещей. Справочные страницы man являются официальным руководством и имеют жестко заданный формат.

Более подробную информацию можно получить по команде `man man`.

whatis

Команда `whatis` представляет собой мини-справочную систему. В качестве аргумента указывается имя файла, на выходе — строка информации об этом файле.

Пример:

```
whatis du
du                    (1)  — estimate file space usage
```

Разное

banner

Команда `banner` выводит слева направо строку, рисуя буквы при помощи символа звездочки *.

bash

Команда `bash` запускает интерпретатор командной строки Bourne Again Shell (модификацию интерпретатора командной строки `sh`). Является интерпретатором командной строки по умолчанию.

bc

Команда `bc` представляет собой калькулятор, позволяющий проводить вычисления с произвольной точностью. Также имеется возможность преобразования чисел из одной системы счисления в другую.

chvt

Команда используется для переключения на указанную виртуальную консоль. Имеет смысл использовать, если в системе более двенадцати виртуальных консолей.

clear

Команда `clear` очищает экран в текстовом режиме.

cpp

Команда `cpp` запускает препроцессор, используемый C-компилятором для преобразования программы перед началом компиляции.

csh

Эта команда запускает C shell — один из используемых в Linux интерпретаторов командной строки.

echo

Команда `echo` выводит текст или значения переменных на стандартное устройство (обычно на экран). Существуют три варианта команды `echo`: команда Linux /bin/echo, а также команды echo-интерпретаторов командной строки C shell и Bourne Again Shell. Эти варианты практически одинаковы.

env

Команда `env` устанавливает значения переменных окружения на время выполнения указанной команды или выводит значения переменных окружения на экран.

Операционная система Linux имеет набор переменных окружения, используемых в различных ситуациях. Например, большинство программ, которым для работы нужен текстовый редактор, используют заданный в переменной окружения `EDITOR`. Другие переменные определяют используемый по умолчанию интерпретатор командной строки, тип терминала, путь, домашний каталог пользователя и т. д.

g77

Программа g77 — компилятор программ на языке Fortran. Современные программисты редко используют этот язык, но осталось обширное "наследие" программного обеспечения от прошлых времен (по крайней мере, на Западе), которое необходимо сопровождать. Язык был разработан фирмой IBM специально для математических расчетов.

gawk

Программа gawk представляет собой GNU-версию языка программирования AWK.

gcc

Программа gcc — компилятор языков программирования C и C ++, используемый в Linux. Существует для большинства версий UNIX и для других операционных систем, что облегчает перенос программного обеспечения (и экономит деньги, поскольку бесплатна).

id

Команда `id` выводит информацию об указанном пользователе. Выводятся системный идентификатор пользователя, его номер, идентификаторы и номера групп, к которым принадлежит пользователь.

login

Команда `login` используется для входа в операционную систему, выполняет некоторые административные задачи, такие как установка UID- и GID-терминала, а также уведомляет пользователя о наличии почты. Кроме того, команда `login` позволяет пользователю root вход в систему только с определенных терминалов, список этих терминалов находится в файле /etc/securetty.

logname

Эта команда выводит имя пользователя, которому принадлежит вызывающий ее процесс. Для его определения используется файл /etc/utmp.

make

Команда `make` управляет группой файлов, из которых создается программа.

Для определения зависимостей между файлами и командами команда `make` использует созданный пользователем файл правил. По умолчанию это файл Makefile.

nohup

Программа `nohup` позволяет продолжить выполнение указанной в той же строке команды после выхода пользователя из операционной системы.

Обычно используется для программ, которые качают большие объемы информации из Интернета или производят длительные расчеты.

openvt

Утилита, позволяющая создавать текстовую консоль (до 64). Можно использовать в том случае, если окажется недостаточно стандартных шести виртуальных консолей. Используется совместно с указанием опций и выполняемой команды, для которой создается консоль.

perl

PERL — это сокращение от Practical Extraction and Report Language, интерпретируемого языка программирования, обычно применяемого для написания системными администраторами различных скриптов, призванных автоматизировать и упростить ежедневные операции администратора. Так же очень широко используется при создании CGI-скриптов для Web-сайтов.

printenv

Эта команда выводит значения переменных окружения. Если в командной строке указана переменная, то выводится ее значение, в противном случае выводятся значения всех переменных окружения.

reset

Эта команда выполняет начальную инициализацию терминала.

resizecons

Утилита позволяет изменить разрешение текстовой консоли (стандартное — 80 символов в строке, 25 строк на экране) в достаточно большом диапазоне.

startx

Команда `startx` предназначена для запуска X Window из командной строки.

После запуска `startx` производится поиск файла .xinitrc в домашнем каталоге пользователя. Этот файл содержит информацию о настройках системы X Window, а также о том, какие X-клиенты должны быть запущены. Большинство этих клиентов запускаются как фоновые процессы, за исключением последнего клиента в списке, который обычно является диспетчером окон.

strings

Команда `strings` выполняет поиск текстовых строк в файле. По умолчанию выводятся только строки, длина которых составляет не менее 4 символов.

strip

Команда `strip` удаляет таблицы символов из объектных файлов. Список объектных файлов может включать библиотеки, но должен быть указан по крайней мере один объектный файл. Используется для уменьшения размеров исполняемых файлов и библиотек.

subst

Команда `subst` производит в файлах указанные подстановки. Обычно она используется для настройки программного обеспечения под конкретную систему. Содержимое каждого из указанных файлов изменяется в соответствии с содержимым файла подстановок.

Файл подстановок содержит по одной подстановке на строке. Строка подстановки состоит из двух полей, разделенных одним или несколькими символами табуляции. Первое поле строки представляет подстановку, второе — значение. Ни одно из полей не должно содержать символа. Строки, начинающиеся с `#`, считаются комментариями и игнорируются.

su

Команда `su` запускает интерпретатор командной строки с правами указанного пользователя. Обычно используется в административных целях для

временного входа под именем пользователя root. В качестве запускаемого интерпретатора командной строки используется интерпретатор командной строки, заданный в файле /etc/passwd для указанного пользователя. Если указанный пользователь имеет пароль, то команда `su` запросит его.

true

Эта команда возвращает код возврата, равный 0, что означает успешное выполнение.

yes

Эта команда непрерывно выводит указанную строку, разделяя две выводимые строки символом новой строки.

Если строка не указана, то выводится символ `y`. Эта команда обычно используется для того, чтобы передать ее стандартный вывод программе, на все вопросы которой следует ответить утвердительно.

Ссылки

- Соответствующие страницы руководства man.
- **www.linuxdocs.org** — разнообразная документация, включая HOWTO.
- Соответствующие HOWTO *(см. гл. 13)*:
  - iptables-HOWTO;
  - NAT-HOWTO.

Часть V

Настройка и сервисы Linux

Глава 15

Локализация

Еще лет десять назад нормальным явлением в компьютерном мире было почти полное отсутствие русского, украинского, белорусского и тому подобных языков в большинстве операционных сред и программ. Знание пользователем английского технического (правильнее сказать — "компьютерного") языка считалось само собой разумеющимся. Такой порядок вещей обуславливался множеством факторов, и в первую очередь тем, что популярные операционные системы производились американскими компаниями и были рассчитаны на англоговорящую аудиторию. С той поры компьютер стал массовым явлением, а наш компьютерный пользователь в большинстве своем английский язык знает либо очень плохо, либо совсем не знает.

Замечание

Чтобы постоянно не перечислять здесь множество основанных на кириллице языков, в дальнейшем мы станем упоминать в этом контексте лишь русский язык, но иметь в виду будем, разумеется, и все остальные.

Большинство коммерческих программ и операционных систем в той или иной мере русифицированы. Что же в этом плане может предложить Linux? Как известно, "спасение утопающих — дело рук самих утопающих", и поскольку Linux операционная система некоммерческая — локализация ее выполняется самими пользователями. В последние год-полтора усилиями наших русскоговорящих разработчиков дистрибутивов, а также фирмы Red Hat и многочисленных энтузиастов, большинство коробочных иностранных дистрибутивов (не говоря уже о русских и украинских) непосредственно после инсталляции могут корректно работать с кириллицей, вплоть до того, что на русский язык переведен и интерфейс многих программ.

Тем не менее, для администратора необходимо знать, каким образом можно локализировать операционную систему Linux.

Поскольку Linux-сообщество велико и разнородно, а программы портировались с различных операционных систем, привести их к одному знаменателю для нормальной локализации, к сожалению, весьма затруднительно.

Однако современные тенденции таковы, что в мире Linux назревает осознание необходимости принятия стандартов на ключевые технологии, в частности, написания программ, с минимальными усилиями локализуемых. Но об этом позже.

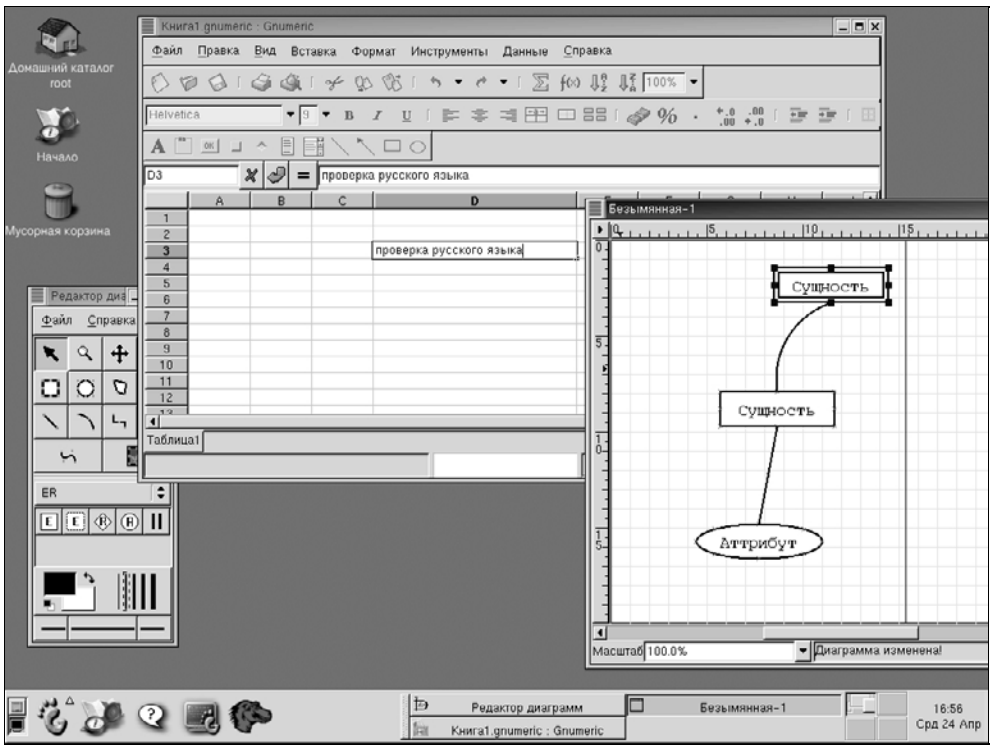

Рис. 15.1. GNOME, говорящий по-русски

А сейчас определим, что нам прежде всего потребуется от хорошо локализованной системы:

- корректно настроенная текстовая консоль (правильные шрифты, ввод и вывод кириллицы);
- корректно настроенная система X Window;
- правильный вывод кириллицы на принтер;
- настроенная на кириллицу система проверки правописания;
- локализованные основные программы — редакторы, офисные пакеты, словари и т. п.

Теоретическая часть

Стандарты кодировки

Как известно, символ как минимальный элемент алфавита, и представление этого символа в компьютере (кодировка) — две разные вещи.

Кодировкой называется совокупность уникальных символов, которые система способна распознать как самостоятельную сущность.

Поскольку первоначально компьютеры были разработаны за рубежом (Великобритания и США) и не предназначались для экспорта, производители не озаботились поддержкой языков, отличных от английского. Со временем это вызвало определенные проблемы, однако решение их осуществлялось хаотично и без учета перспектив дальнейшего развития программ и аппаратуры. В результате русскоязычное компьютерное сообщество получило несколько различных кодировок, в той или иной степени учитывающих национальную специфику.

Стандарт ASCII

Наиболее популярной кодировкой была (и фактически ей и остается) кодировка ASCII (Американский Стандартный Код для Информационного Обмена).

Стандарт ASCII, иногда называемый 7-битный ASCII, включает в себя 128 уникальных символов. Они подразделяются на символы, которые ASCII определяет как печатаемые символы, и на символы управления, бóльшая часть которых использовалась в старых протоколах связи. Каждому элементу набора соответствует целочисленный символьный код от 0 до 127.

Со временем 7-битный стандарт ASCII был расширен до 8-битного ASCII (расширенный ASCII). Этот стандарт подразумевает наличие 256 символов, которые соответствуют кодам от 0 до 255. Первая часть таблицы — от 0 до 127 не претерпела по сравнению с предыдущим стандартом никаких изменений, а во второй половине таблицы пусто. Дело в том, что 8-битный стандарт ASCII не определяет содержание второй половины таблицы кодировки. В этих целях Международная Организация по Стандартизации (ISO) выпустила серию стандартов (известных как семейство ISO 8859-*x*), определяющих кодировку второй половины таблицы для различных языков. Нас как пользователей кириллицы интересуют следующие кодовые страницы:

- 8859-0 — новый европейский стандарт (Latin 0);
- 8859-1 — Европа, Латинская Америка (Latin 1);
- 8859-2 — Восточная Европа.
- 8859-5 — кириллица.

В кодовой странице 8859-1 (Latin 1) старшая половина таблицы определяет различные символы, которые не входят в английский алфавит, но присутствуют в различных европейских языках. Соответственно, в остальных кодировках в старшей половине таблицы находятся специфические национальные символы, входящие в алфавит указанного региона.

Есть еще одна реализация расширенного ASCII — *кодовая страница IBM*. Эта кодировка в старшей половине содержит псевдографические символы.

Казалось бы, вполне достаточный набор стандартов. Однако есть несколько отрицательных моментов:

- ограничение набора модифицируемых символов (128);
- невозможность использования в одном чисто текстовом документе нескольких кодировок.

Альтернативная кодировка (CP866)

Альтернативная кодировка (CP866) — это кодовая страница IBM, где все специфические европейские символы были заменены на буквы из кириллицы, оставив нетронутыми псевдографические символы.

Кодировка Microsoft CP1251

Кодовая страница Microsoft CP1251 — это попытка Microsoft облегчить труд программисту. Используется для кодировки кириллицы в Windows. Устраняет проблему с сортировкой по алфавиту, связанную с тем, что в странице CP866 буквы русского алфавита располагались не подряд.

Стандарт КОИ8

Стандарт был разработан достаточно давно, когда во всю использовалась 7-битная кодировка символов ASCII. Разработчики КОИ8 поместили символы русской кириллицы в верхней части расширенной таблицы ASCII таким образом, чтобы позиции кириллических символов соответствовали их фонетическим аналогам в английском алфавите в нижней части таблицы. Так, если в тексте, хранящемся в кодировке КОИ8, убрать старший (восьмой) бит каждого символа, то получится текст, написанный английскими символами в русской транскрипции. Например, предложение "Мама мыла раму" после удаления старшего бита будет выглядеть так: "Mama myla ramu".

Существует несколько реализаций стандарта КОИ8, в частности, KOI8-R — для русского языка, KOI8-U — для украинского.

Стандарт RFC 1489 Registration of a Cyrillic Character Set, созданный Андреем Черновым, регламентирует использование KOI8-R для представления

русскоязычных документов в Интернете, где KOI8-R давно уже стал фактическим стандартом для русской кириллицы.

Unicode

Unicode — частичная реализация стандарта ISO 10646, в котором первые 256 символов соответствуют кодировке Latin-1 (ISO 8859-1). Основная идея этого стандарта — кодирование символа с использованием переменного количества байтов (до 8). На данном этапе используется двухбайтное кодирование символа, дающее возможность определить 65 535 символов. В настоящее время позиции зарезервированы за буквами практически всех известных алфавитов, включая древнеегипетские иероглифы, благодаря чему можно, используя всего один шрифт, писать одновременно на русском и греческом, английском и иврите и делать еще вставки на японском. Используется в Windows 98 и более поздних версиях. В UNIX-системах поддержка Unicode реализована частично.

Украинский язык

Специфика локализации для Украины состоит в том, что зачастую нужно использовать и украинский, и русский языки одновременно (бóльшая часть жителей городов Центральной, Южной и Восточной Украины — русскоговорящие, а официальный язык — украинский).

Кириллизация консоли

В большинстве современных дистрибутивов кириллизация консоли происходит по запросу при инсталляции. Однако необходимо рассмотреть способы кириллизации текстового режима как фундамента, на котором держится локализация операционной системы в целом.

Консольный драйвер

Для настройки консоли можно воспользоваться пакетами console-tools, Cyrillic console-tools (модификация console-tools с расширенным набором шрифтов и дополнительными свойствами) или kbd. В дистрибутиве Red Hat Linux в ранних версиях применялся пакет kbd, в более поздних (начиная с версии 6) — console-tools. Чтобы не упустить особенностей, рассмотрим использование обоих пакетов.

Схема функционирования консольного драйвера

Для понимания дальнейших действий необходимо четко представлять, как функционирует консольный драйвер.

В Linux применяются две таблицы символов — таблица символов приложения (Application Charset Map, ACM) и таблица экранных шрифтов (Screen Font Map, SFM).

Когда программа предлагает консольному драйверу вывести на экран символ, имеющий код, например A, то консольный драйвер прежде ищет код A в таблице символов приложения. Из нее он узнает, какой код B согласно кодировке Unicode соответствует коду A. Далее консольный драйвер ищет код B в таблице экранных шрифтов. Из нее он узнает, какой символ активного шрифта имеет код B, и выводит его на экран. A используемую операционной системой кодировку посредством таблицы символов приложения определяет пользователь.

Аппаратные ограничения видеокарт VGA не позволяют использовать в текстовом режиме шрифты, имеющие более 512 символов. Поэтому иногда консольный драйвер не может найти код B в таблице экранных шрифтов. В этом случае используется так называемая fallback-таблица. Она определяет для кода B возможные его аппроксимации B1, B2 и т. д. Например, если B является кодом символа "левая двойная угловая кавычка", то, возможно, B1 будет кодом символа "левая одинарная угловая кавычка", а B2 будет просто кодом символа <.

Настройка поддержки кириллицы с помощью пакетов console-tools и kbd состоит из:

- настройки экранного шрифта и таблицы экранных шрифтов. Это делается с помощью программы consolechars (для console-tools) или setfont и mapscrn (для kbd);
- настройки таблицы символов приложения и fallback-таблицы;
- загрузки соответствующей раскладки клавиатуры с помощью программы loadkeys.

Файлы шрифтов обычно размещаются в каталогах /usr/share/consolefonts или /usr/lib/kbd/consolefonts, символьные таблицы в каталоге /usr/share/consoletrans, клавиатурные раскладки в /usr/share/keymap/i386/qwerty.

console-tools

Если на компьютере установлен пакет console-tools, то необходимо выполнить следующие действия:

```
loadkeys ru.map
consolechars -v -f Cyr_a8x16 -m $foo/koi2alt
```

Переключение раскладки клавиатуры производится нажатием правой клавиши <Ctrl> (иногда это можно сделать нажатием клавиши <Alt> или <Caps Lock>).

Cyrillic console tools

Все шрифты в этом пакете основаны на альтернативной кодировке (CP866). Это сделано потому, что в текстовом режиме VGA использование другой кодировки приводит к разрывам в отображении горизонтальной псевдографики. Все шрифты содержат в себе таблицу отображения в Unicode.

В пакет также включены таблицы перекодировки в Unicode из распространенных кодировок русского, белорусского, болгарского, сербского и украинского языков.

Для настройки консоли следует выполнить следующие команды:

```
consolechars -f UniCyr_8x16.psf -m koi8-r.acm
loadkeys console_russian.map
```

Для украинизации вместо `koi8-r` необходимо подставить `koi8-u`.

kbd

Для настройки кириллицы с помощью kbd обычно используются следующие команды:

```
loadkeys /usr/lib/kbd/keytables/ru.map
setfont /usr/lib/kbd/consolefonts/Cyr_a8x16
mapscrn /usr/lib/kbd/consoletrans/koi2alt
# ниже идет "магическая" последовательность
echo -ne "\033(K"
```

Во время загрузки системы для русификации всех виртуальных текстовых консолей необходимо выполнить команду

```
echo -ne "\033(K"
```

семь раз. Это можно сделать с помощью следующей строки:

```
for i in 1 2 3 4 5 6 7; do echo -ne "\033(K" > /dev/tty$i; done
```

"Магическая" последовательность необходима для перекодировки вывода символов на экран при использовании шрифтов, основанных на кодовой странице CP866.

Настройка консольных приложений

После настройки консоли необходимо также настроить и консольные программы, которые работают с символами. Основная проблема большей части этих программ — они считают, что используется 7-битная кодировка символа.

bash

В файле .inputrc, находящемся в домашнем каталоге пользователя, необходимо установить следующие три переменные:

```
set meta-flag on
set convert-meta off
set output-meta on
```

Эти строки указывают, что для кодирования символа используется 8-битная последовательность.

Поскольку файл .inputrc является конфигурационным файлом библиотеки GNU readline, внесенные исправления кириллизируют не только bash, но и другие программы, использующие GNU readline.

csh/tcsh

Те же действия в отношении программ csh/tcsh будут выглядеть следующим образом:

В файле .cshrc необходимо добавить следующие строки:

```
setenv LC_CTYPE iso_8859_5
stty pass8
```

zsh

В файле .zshrc необходимо добавить следующие строки:

```
setenv LC_CTYPE iso_8859_5
stty pass8
```

less

Для нормального функционирования программы less в файл ~/.lesskey необходимо добавить:

```
LESSCHARSET=
```

Это позволяет программе игнорировать установку переменной `LESSCHARSET=` другими программами. После этого надо запустить lesskey для получения бинарного файла ~/.less.

mc (The Midnight Commander)

Чтобы использовать кириллицу в mc, нажатием клавиши <F9> зайдите в системное меню, выберите пункт меню **Options | Display** и установите опцию **full 8 bits**.

nroff

Для корректной работы nroff с кириллицей необходимо запускать его с ключом Tlatin1.

man

Если программа man не желает корректно отображать кириллицу на экране, необходимо правильно настроить less.

Также измените в файле /usr/lib/man.conf строку:

```
NROFF        /usr/bin/groff -S -Tascii -mandoc
```

на

```
NROFF       /usr/bin/groff -S -Tlatin1 -mandoc
```

ls

При неправильно настроенной локали ls не будет выводить кириллические символы. В этом случае поможет одна из следующих команд:

- `ls -N`
- `ls --show-control-chars`

Samba

Чтобы увидеть кириллические символы в именах файлов, в файл /etc/smb.conf следует добавить следующие строчки:

```
[global]
character set = koi8-r
client code page = 866
preserve case = yes
short preserve case = yes
```

Первые две строки указывают кодировку пользователя (`character set = koi8-r`) и кодировку имен файловой системы (`client code page = 866`).

Третья и четвертая строки определяют, что необходимо сохранять регистр длинных и коротких имен файлов.

telnet

При возникновении проблемы ввода русских символов необходимо создать файл ~/.telnetrc, содержащий следующую строку:

```
DEFAULT set outbinary
```

Локализация и интернационализация

Как можно заметить, каждая из вышеперечисленных программ требует своего особого подхода, выражающегося в том или ином изменении индивидуальных конфигурационных файлов. Это сильно раздражает. А проблема возникла потому, что при проектировании программ не учитывались какие-либо национальные особенности.

Решение таких проблем основывается на двух базисных концепциях: локализации (Localization, l1on) и интернационализации (Internationalization, i18n).

Под *локализацией* подразумевается написание программного кода, способного адекватно воспринимать, использовать и обрабатывать различающиеся стандарты представления данных для различных стран. Например: формат записи даты в США имеет вид ММ/ДД/ГГ, в странах СНГ — ДД.ММ.ГГ, а в Японии — ГГ.ММ.ДД. Помимо даты необходимо разрешить проблемы с представлением форматов времени, чисел, валюты и т. п. Кроме того, базовый аспект локализации — определение соответствующих классов символов.

Интернационализация должна решать проблемы, связанные со способностью программ взаимодействовать с пользователем на его родном языке.

Обе эти концепции должны быть стандартизованы, давая программистам непротиворечивый путь создания программ, работающих в национальной среде.

Локаль

Одно из основных понятий локализации — локаль (locale). Под локалью подразумевается набор соглашений, специфичных для отдельно взятого языка в отдельно взятой стране.

Каждая локаль определяет, по меньшей мере, следующие соглашения:

- классификация символов и преобразований;
- представление валюты;
- представление чисел;
- формат даты/времени.

Настройка локали

Локализация включается путем задания переменной окружения `LANG` строкой:

```
export LANG={язык}
```

В том случае, если такой строки не существует, используется значение локализации по умолчанию: `LANG="C"` или `LANG="POSIX"`.

По стандарту POSIX.2 язык локализации записывается в форме:

```
language_TERRITORY.Codeset
```

где

- `language` — двухсимвольный код, обозначающий язык (ru, fr и т. д.);
- `TERRITORY` — двухсимвольный код, обозначающий страну (RU, UA и т. д.);
- `Codeset` — определяет кодировку символов.

Стандарт ISO 639 определяет коды языков, ISO 3166 — коды стран.

Для русского языка переменная `LANG` устанавливается, как правило, равной `LANG="ru_RU.KOI8-R"` или `LANG="ru_RU.ISO_8859-5"`.

Также по стандарту допустимы короткие именования значений локали, которые часто используются в качестве псевдонимов. Наиболее известная пара псевдоним-наименование: "C" — "POSIX".

Локализацию можно провести частично, либо определить для отдельных категорий локализации значения, отличные от общесистемной локализации.

В табл. 15.1 приведены опции локали.

Таблица 15.1. *Опции локали*

Опция	Описание
LC_ALL	Задает значение для всех опций (не рекомендуется использовать)
LC_CTYPE	Определяет одиночные символы
LC_NUMERIC	Задает формат чисел
LC_TIME	Определяет формат времени
LC_COLLATE	Используется для сравнения строк
LC_MONETARY	Задает формат валюты
LC_MESSAGES	Системные сообщения
LC_PAPER	Задает формат бумаги
LC_NAME	Задает формат имен
LC_ADDRESS	Задает формат адресов
LC_TELEPHONE	Задает формат телефонов

Посмотреть текущие значения категорий локализации можно утилитой locale (без параметров).

Интернационализация

Интернационализация детализирует способы общения программы с неанглоговорящим пользователем. Для этого при создании программы используются функции, специально предназначенные для создания интернационализированных программ. Эти функции и особенности их применения описываются документом LI18NUX 2000 Globalization Specification Version 1.0 with Amendment 2 Linux Internationalization Initiative (Li18nux).

Кириллизация X Window

X Window в современных дистрибутивах кириллизованы "из коробки". Однако зачастую некоторая настройка X Window все же требуется. Рассмотрим кириллизацию X Window 4.*x*.

Установка шрифтов для X Window

Сначала необходимо установить кириллические шрифты. В современных дистрибутивах они включены в поставку, однако возможно найти и установить свои кириллические шрифты.

Предварительно следует проверить, установлены ли вообще кириллические шрифты в вашей операционной системе. Для этого выполните команду: `xlsfonts | grep koi8`. В результате ее выполнения будет выдан список кириллических шрифтов. Еще один вариант — поискать местоположение шрифта, в котором заведомо присутствует кириллица: `find / -name crox\*.pcf\*`.

В том случае, если кириллические шрифты не обнаружены, установите их самостоятельно:

1. Создайте каталог /usr/lib/X11/fonts/cyrillic и скопируйте туда кириллические шрифты.
2. Если шрифты получены в формате BDF (файлы *.bdf), то перед использованием их необходимо скомпилировать. Для каждого шрифта выполните: `bdftopcf -o <font>.pcf <font>.bdf`.
3. В каталоге шрифтов для X Window должен присутствовать файл fonts.dir, хранящий список шрифтов, находящихся в каталоге. Этот список создается с помощью команды: `cd <каталог где лежат шрифты>; mkfontdir`.
4. Далее о появлении нового каталога шрифтов необходимо поставить в известность X-сервер. Этого можно достичь несколькими способами:
   - добавьте новый каталог к списку каталогов в файле XF86Config;
   - добавьте новый каталог к файлу запуска xinit;

- персональная настройка. У пользователя есть специальный файл для X Window — .xinitrc.

5. Здесь мы добавляем каталог с новыми шрифтами в список каталогов, просматриваемых сервером в поисках шрифтов: `xset +fp <новый каталог шрифтов>`.
6. А теперь мы указываем серверу перечитать свои конфигурационные файлы: `xset fp rehash`.

Шрифты TrueType

Шрифты TrueType, в отличие от Type1, не являются "родными" для X Window. Первоначально разработанные компанией Apple, эти шрифты используются операционными системами серии Windows. Помимо большого разнообразия шрифтов формата TrueType, у них отсутствуют многие недостатки стандартных шрифтов X Window. Поддержка шрифтов TrueType встроена во все современные дистрибутивы XFree86.

Для того чтобы шрифты отображались в нужной кодировке, в каталоге, где находятся шрифты TrueType, необходимо создать два одинаковых файла, fonts.dir и fonts.scale, следующего вида:

```
12
timesi.ttf -monotype-Times New Roman-medium-i-normal--0-0-0-0-p-0-
microsoft-cp1251
timesbi.ttf -monotype-Times New Roman-bold-i-normal--0-0-0-0-p-0-
microsoft-cp1251
timesbd.ttf -monotype-Times New Roman-bold-r-normal--0-0-0-0-p-0-
microsoft-cp1251
times.ttf -monotype-Times New Roman-medium-r-normal--0-0-0-0-p-0-
microsoft-cp1251
couri.ttf -monotype-Courier New-medium-i-normal--0-0-0-0-m-0-microsoft-
cp1251
courbi.ttf -monotype-Courier New-bold-i-normal--0-0-0-0-m-0-microsoft-
cp1251
courbd.ttf -monotype-Courier New-bold-r-normal--0-0-0-0-m-0-microsoft-
cp1251
cour.ttf -monotype-Courier New-medium-r-normal--0-0-0-0-m-0-microsoft-
cp1251
ariali.ttf -monotype-Arial-medium-i-normal--0-0-0-0-p-0-microsoft-cp1251
arialbi.ttf -monotype-Arial-bold-i-normal--0-0-0-0-p-0-microsoft-cp1251
arialbd.ttf -monotype-Arial-bold-r-normal--0-0-0-0-p-0-microsoft-cp1251
arial.ttf -monotype-Arial-medium-r-normal--0-0-0-0-p-0-microsoft-cp1251
```

Кроме этого, здесь же необходимо создать файл encodings.dir, имеющий всего две строки:

```
1
```

```
microsoft-cp1251 /usr/X11R6/lib/X11/fonts/encodings/microsoft-
cp1251.enc.gz
```

Если шрифты нужны в кодировке KOI8-R, то вместо `microsoft-cp1251` следует прописать `koi8-r`.

Если используется сервер шрифтов xfstt вместо xfsft, то необходимо применять опцию перекодировки:

```
xfstt ... --encoding koi8-r,windows-1251,iso8859-1
```

Ввод с клавиатуры

В X Window имеются два способа ввода символов с клавиатуры: устаревший — с помощью утилиты xmodmap и новый — с помощью модуля XKB (X KeyBoard). Мы рассмотрим только использование XKB.

При старте X-сервера модуль XKB считывает текстовые конфигурационные файлы. Данные, получаемые модулем XKB, состоят из 5 компонентов:

- keycodes — таблицы, которые задают символические имена для скан-кодов клавиатуры;
- types — описывает типы клавиш;
- compat — описывает модификаторы. В XKB имеется несколько внутренних переменных, которые определяют, какой символ будет генерироваться при нажатии клавиши в конкретной ситуации. В compat описывается, как изменяются переменные при нажатии различных клавиш-модификаторов;
- symbols — таблицы, в которых для каждого скан-кода перечисляются все значения, которые должна выдавать клавиша;
- geometry — описывает расположение клавиш на клавиатуре.

Эти компоненты находятся в одноименных каталогах в дереве каталога библиотек X Window.

Набор компонентов, необходимых для настройки XKB, описывается в файле конфигурации X-сервера в секции Keyboard.

Настройка XKB

Для настройки XKB в файле конфигурации X Window необходимо определить параметры `XkbRules`, `XkbModel`, `XkbLayout`, `XkbVariant` и `XkbOptions`.

Например,

```
XkbRules        "xfree86"
XkbModel        "pc104"
XkbLayout       "ru"
```

```
XkbVariant      "winkeys"
XkbOptions      "grp:shift_toggle"
```

В вышеприведенном примере определяется, что XKB должен воспользоваться правилами, описанными в файле xfree86, использовать настройки для клавиатуры типа `pc104` (104 клавиши), русский алфавит (в дополнение к английскому алфавиту). `XkbVariant` определяет, что используется Windows-клавиатура. `XkbOptions` определяют дополнительные настройки клавиатуры, в частности, комбинацию клавиш для переключения раскладки клавиатуры "русская-английская".

Варианты клавиш для переключения раскладки клавиатуры:

- grp:toggle — переключение нажатием правой клавиши <Alt>;
- grp:shift_toggle — переключение нажатием комбинации клавиш <Shift>+<Shift>;
- grp:ctrl_shift_toggle — переключение нажатием комбинации клавиш <Ctrl>+<Shift>;
- grp:ctrl_alt_toggle — переключение нажатием комбинации клавиш <Ctrl>+<Alt>;
- grp:switch — переключение нажатием правой клавиши <Alt> (только на момент нажатия);
- ctrl:ctrl_ac — переключение нажатием клавиши <CapsLock>.

Полная документация по настройке XKB размещена на Web-странице Ивана Паскаля по адресу **www.tsu.ru/~pascal/other/xkb/**.

Работа с текстом

Этот раздел посвящен программам, тем или иным способом обрабатывающим текст.

Проверка правописания

Одна из лучших программ проверки правописания для операционных систем UNIX — программа ispell. Ее путем добавления новых словарей можно использовать при проверке правописания текстов, написанных на языках, отличных от английского.

Для правильной работы ispell необходимо скомпилировать с поддержкой 8-битных символов и установить словарь русских слов. О некоторых таких словарях рассказано ниже.

Словарь Александра Лебедева

Словарь постоянно совершенствуется и дополняется и корректируется. Отличительной чертой его является полноценная поддержка буквы ё. Послед-

нюю версию словаря можно найти по адресу **ftp://mch5.chem.msu.su/pub/russian/ispell/rus-ispell.tar.gz**.

Словарь Константина Книжника

В поставку словаря включен скрипт, обеспечивающий инкрементный режим проверки правописания слов для emacs. Найти словарь можно по адресу **www.ispras.ru/~knizhnik**.

Редактор vim

После корректной настройки редактор vim нормально работает с кириллическими символами. Единственное неудобство: редактор понимает управляющие команды, набранные только в английской раскладке. Такое ограничение можно обойти, произведя (для командного режима) отображение кириллических символов на английские с помощью опции `langmap`. Для этого достаточно добавить в файл .vimrc две строки:

```
set langmap=ж;;
set lang-
map=е`,йq,цw,уe,кr,еt,нy,гu,шi,щo,зp,х[,ъ],фa,ыs,вd,аf,пg,рh,оj,лk,дl,э',
яz,чх,сс,мv,иb,тn,ьm,б\,,ю.,Е~,ЙQ,ЦW,УЕ,КR,ЕТ,НY,ГU,ШI,ЩO,ЗР,Х{,Ъ},ФА,ЫS,
ВD,АF,ПG,РН,ОJ,ЛК,ДL,Ж:,Э\",ЯZ,ЧХ,СС,МV,ИВ,ТN,ЬМ,Б<,Ю>
```

Редактор joe

Для того чтобы распознавать 8-битные символы, joe использует опцию `-asis`. Ее можно указать в командной строке или вставить в файл .joerc.

StarOffice

StarOffice 5.2 — может работать с различными кодировками кириллицы. Таких кодировок поддерживается три: KOI8-R, ISO 8859-5, CP1251. Текущая кодировка определяется по значению переменной окружения `LANG`.

StarOffice использует два вида шрифтов: растровые (pcf) для элементов интерфейса и Type1 — для печати и отображения документов на экране.

Самая полная информация по русификации StarOffice содержится на Web-странице Леона Кантера по адресу **www.blackcatlinux.com/StarOffice/**.

Выбор кодировки для работы со StarOffice

Как указывалось выше, StarOffice 5.2 может работать в одной из трех кодировок: KOI8-R, ISO8859-5, CP1251. Каждая из этих кодировок имеет свои достоинства и недостатки.

- KOI8-R. Достоинство — позволяет в этой кодировке отправлять письма и сообщения в группы новостей. Недостатки: при импорте из Microsoft Word не хватает символов (недостающие заменяются вопросительными знаками).
- ISO 8859-5. Единственная кодировка, которая позволяет работать в StarOffice 5.2 со встроенными в эту версию словарями для проверки русской орфографии и переносов. Содержит также украинские и белорусские буквы. Недостатки: письма, отправленные в этой кодировке, не читаются большинством доступных клиентов.
- CP1251. Позволяет корректно импортировать документы из Microsoft Office, содержит полный набор специальных символов, включая знак Евро, дает возможность работать на любом из славянских языков. Недостатки — недоступен русский словарь.

Для запуска StarOffice в кодировке, отличной от системной, можно указать полное имя locale непосредственно в командной строке:

```
LANG=ru_RU.CP1251 ~/office52/program/soffice
```

Подключение для печати шрифтов Type1

Для того чтобы работать в StarOffice с кириллицей, необходимо сначала добавить новые шрифты к драйверу печати StarOffice — библиотеке Xprinter с помощью специальной утилиты SPAdmin (входит в пакет). Ее необходимо запустить на выполнение и в пункте меню **Add Fonts | Browse** указать каталог, где лежат шрифты Type1.

Печать из StarOffice

Для вывода на принтер StarOffice генерирует данные в формате PostScript. Дополнительная русификация интерпретатора Ghostscript не требуется, т. к. все необходимые шрифты встраиваются в документ.

Проблемы при работе со StarOffice 5.2

Основная проблема заключается в том, что в этой версии испорчены все фильтры для работы с форматами, которые не предусматривают указания кодировки: TXT, RTF, MS Word 6.0/95. При сохранении в любом из этих форматов русские буквы заменяются на знак вопроса.

Кириллица в программах электронной почты и чтения новостей

Для настройки программы электронной почты необходимо указать:

- что письма будут содержать 8-битные символы;
- кодировку, в которой вы работаете;
- кодировку, в которой отсылаются письма.

elm

Добавьте следующую запись в файл ~/.elm/elmrc:

```
CHARSET=koi8-r
```

pine

Добавьте следующую запись в файл pine.conf для настройки всей системы:

```
character-set=koi8-r
```

Можно также изменить настройку pine для того, чтобы предотвратить посылку письма в кодировке quoted-printable:

```
enable-8bit-nntp-posting
enable-8bit-esmtp-negotiation
```

Чтобы настроить перекодировку win в koi в программе pine, в файле .pinerc следует прописать:

```
display-filters=_CHARSET(iso8859-5)_ /usr/local/bin/icat,
        _CHARSET(utf-8)_ /usr/local/bin/ucat,
        _CHARSET(windows-1251)_ /usr/local/bin/wcat
```

Вместо программ icat, wcat и ucat можно воспользоваться другими, например iconv.

mutt

Добавьте следующую запись в файл .muttrc:

```
set charset=koi8-r
set allow_8bit
```

Для перекодировки посылаемых писем из одной кодировки в другую (отображаются символы в кодировке КОИ8, посылается сообщение в кодировке CP1251) необходимо добавить следующие строки в файл .muttrc:

```
set charset= koi8-r
set send_charset= windows-1251
set allow_8bit
```

tin

Для включения отображения кириллицы в файл конфигурации .tin/tinrc добавьте следующие строки:

```
post_mime_encoding=8bit
mail_mime_encoding=8bit
```

Кириллические имена файлов

При стандартном монтировании разделов FAT32 созданные в Windows имена файлов с кириллическими символами видны как набор вопросительных знаков. Для решения этой проблемы необходимо при монтировании раздела указать кодировку символов, в которой хранятся имена файлов, и кодировку, в которой необходимо эти имена файлов отображать.

Так, монтируя раздел FAT32, при вызове команды `mount` добавьте следующие опции: `codepage=866`, `iocharset=koi8-r`.

Если компакт-диск содержит файлы с кириллическими шрифтами, используйте следующую команду монтирования:

```
mount -t iso9660 -o iocharset=koi8-r  /dev/cdrom  /mnt/cdrom
```

Поддержка кириллицы в Perl

Для того чтобы можно было правильно применять регулярные выражения в кириллических текстах, а так же использовать стандартные функции преобразования текста, в программу на Perl необходимо добавить следующие строки:

```
use locale;
use POSIX qw (locale_h);
setlocale(LC_CTYPE, 'ru_RU.KOI8-R');
```

Перекодировщики

Наиболее широко распространены перекодировщики iconv и recode. Для использования iconv следует указать в командной строке кодировку файла и кодировку, в которой необходимо сохранить файл. Например при перекодировке из CP866 в КОИ8-R:

```
iconv -f866 -tKOI8-R -o<outfile> infile
```

Похожим образом используется и программа recode:

```
recode CP1251..KOI8-R  winfile.txt
```

Ссылки

- RFC 1489 — стандарт, описывающий кодировку KOI8-R.
- RFC 2319 — стандарт, описывающий кодировку KOI8-U.
- **www.unicode.org** — сайт, посвященный Unicode.
- **charts.unicode.org** — на этом сайте можно посмотреть набор символов Unicode.
- **www.sensi.org/~alec/** — сайт, посвященный локализации.
- **www.tsu.ru/~pascal/x_locale/** — сайт Ивана Паскаля: локаль и X Window.
- **www.inp.nsk.su/~baldin** — Балдин Евгений. The Linux Cyrillic HOWTO (rus). Здесь же расположен Cyrillic HOWTO (old rus), перевод устаревшего англоязычного документа.

Глава 16

Обновление и компиляция ядра

Системный администратор рано или поздно сталкивается с необходимостью обновления ядра операционной системы Linux. И возникает дилемма — искать новое ядро операционной системы в виде инсталляционного пакета или самостоятельно скомпилировать его из исходных текстов.

Рассмотрим более простой вариант — обновление ядра операционной системы Linux из пакета RPM, созданного специалистами фирмы Red Hat.

Обновление ядра операционной системы Linux

Мир не без добрых людей. Как правило, почти все производители дистрибутивов Linux производят выпуск обновленных пакетов программ, в том числе и ядра операционной системы. Это, правда, происходит с некоторой временной задержкой, да и не всегда в инсталляционных пакетах выходят все версии ядра операционной системы.

Дальнейшее описание процесса обновления ядра операционной системы будет основываться на документе "Red Hat Linux 7.2 The Official Red Hat Linux Customization Guide".

Подготовка к обновлению ядра операционной системы

Как обычно, перед любыми действиями, затрагивающими жизнедеятельность системы, необходимо произвести ряд мероприятий, позволяющих восстановить систему в случае краха. Для этого следует создать загрузочную дискету, содержащую образ работоспособного ядра операционной системы Linux.

Прежде чем создать загрузочную дискету, надо выяснить, какая версия ядра установлена в вашей операционной системе. Самый канонический способ — выполнить следующую команду:

```
uname -r
```

После того как версия ядра опознана, можно создавать загрузочную дискету. Для этого необходимо зайти в систему пользователем root и выполнить следующую команду:

```
/sbin/mkbootdisk kernelversion
```

где `kernelversion` — версия ядра, полученная с помощью команды `uname`.

После этого загрузите систему с помощью полученной загрузочной дискеты, чтобы убедиться в ее работоспособности. Небольшой совет — сделайте две загрузочных дискеты, всякое бывает.

Следующим этапом подготовки будет определение всех установленных пакетов, относящихся к ядру операционной системы. Для этого выполним команду:

```
rpm -qa | grep kernel
```

В результате вы получите что-то подобное:

```
kernel-headers-2.4.7-3
kernel-2.4.7-3
kernel-source-2.4.7-3
kernel-doc-2.4.7-3
```

На основании этого списка определим пакеты, которые необходимо загрузить из Интернета. Если у вас хороший канал, желательно загрузить и обновить все относящиеся к ядру установленные пакеты. Если же нет — загружаемые пакеты зависят от ваших намерений:

- для обновления ядра операционной системы — загружаем только `kernel-2.4.xx`.;
- для перекомпилирования ядра операционной системы — необходимо загрузить пакеты `kernel-headers-2.4.xx`, `kernel-source-2.4.xx`.

Загрузку необходимых пакетов можно осуществить напрямую с FTP-сервера. Список доступных зеркал находится по адресу **www.redhat.com/download/mirror.html**.

Обновление ядра операционной системы

Теперь, когда все необходимые пакеты получены, можно приступать к обновлению ядра операционной системы. Выполнить это можно двумя способами:

- командой `rpm -Uvh kernel-2.4.XX.i386.rpm` — обновить ядро операционной системы;
- командой `rpm -ivh kernel-2.4.xx.i386.rpm` — установить новое ядро операционной системы.

Второй способ позволит в случае, если новое ядро вызывает проблемы, безболезненно "откатиться" (roll back, downgrade) на старое ядро операционной системы.

Аналогично обновляются пакеты с исходными текстами ядра операционной системы Linux.

Для проверки обновления ядра выполните следующую команду:

```
ls -l /boot
```

Вы должны увидеть следующий файл: vmlinuz-2.4.*xx*.

После обновления ядра операционной системы необходимо осуществить конфигурирование загрузчика (boot loader).

Конфигурирование загрузчика

После установки нового ядра операционной системы необходимо сконфигурировать загрузчик таким образом, чтобы при последующих стартах операционной системы производилась загрузка ее обновленного ядра.

Предупреждение

Будьте предельно внимательны во время конфигурирования загрузчика — если вы ошибетесь, операционная система Linux не сможет загрузиться. В этом случае придется воспользоваться заблаговременно созданной загрузочной дискетой и внимательно переконфигурировать загрузчик.

В дистрибутиве Red Hat Linux 7.2 существует возможность при инсталляции выбрать устанавливаемый загрузчик — GRUB или LILO. Поэтому рассмотрим конфигурирование обоих загрузчиков.

GRUB

Если у вас установлен загрузчик GRUB, вы должны отредактировать файл /boot/grub/grub.conf.

Типичный конфигурационный файл GRUB приведен ниже:

```
# NOTICE: You have a /boot partition. This means that
# all kernel paths are relative to /boot/
default=0
timeout=30
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Linux (2.4.7-3)
root (hd0,0)
kernel /vmlinuz-2.4.7-3 ro root=/dev/hda3
initrd /initrd-2.4.7-3.img
```

Добавление нового ядра в список загрузчика, рекомендуется производить в два этапа:

1. Сначала добавить новую секцию для нового ядра и убедиться, что загрузка происходит нормально. Добавить новую секцию проще всего, скопировав существующую и подправив ее в нужном месте. В результате получим следующий текст (добавленная секция выделена полужирным шрифтом):

```
# NOTICE: You have a /boot partition. This means that
# all kernel paths are relative to /boot/
default=0
timeout=30
splashimage=(hd0,0)/grub/splash.xpm.gz
title My new kernel (2.4.12)
root (hd0,0)
kernel /vmlinuz-2.4.12 ro root=/dev/hda3
initrd /initrd-2.4.12.img
title Red Hat Linux (2.4.7-3)
root (hd0,0)
kernel /vmlinuz-2.4.7-3 ro root=/dev/hda3
initrd /initrd-2.4.7-3.img
```

   После редактирования конфигурационного файла следует произвести перезагрузку операционной системы и выбрать новое ядро.

2. Убедившись, что загрузка происходит без эксцессов и система функционирует нормально, удалите из конфигурационного файла описание старой версии ядра.

LILO

Конфигурирование LILO в целом похоже на конфигурирование GRUB. Конфигурационный файл LILO находится по адресу /etc/lilo.conf.

Типичный файл /etc/lilo.conf похож на тот, что приведен ниже:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=linux
```

```
image=/boot/vmlinuz-2.4.7-3
label=linux
initrd=initrd-2.4.7-3.img
read-only
root=/dev/hda5
```

Аналогично GRUB, модернизацию конфигурационного файла LILO производим в два этапа.

1. Сначала получим файл, где прописаны оба ядра — старое и новое:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=linux
image=/boot/vmlinuz-2.4.12
label=new Linux kernel
initrd=initrd-2.4.12.img
read-only
root=/dev/hda5
image=/boot/vmlinuz-2.4.7-3
label=Linux
initrd=initrd-2.4.7-3.img
read-only
root=/dev/hda5
```

   Для того чтобы внесенные в конфигурационный файл изменения вступили в силу, необходимо выполнить следующую команду: `/sbin/lilo`. Если все в порядке, на экране появятся следующие строки:

```
Added linux *
Added new Linux kernel
```

   Символ `*` после `linux` обозначает, что секция, помеченная как `linux`, является загружаемой по умолчанию.

2. Проверьте, как функционирует новое ядро. Если проблем нет — уберите из конфигурационного файла LILO секцию, относящуюся к старому ядру.

Компиляция ядра операционной системы Linux

Ядро операционной системы Linux может быть двух типов — монолитное и модульное.

Модульное ядро — это такое ядро, где драйверы почти всех устройств не вкомпилированы в него, а содержатся во внешних модулях, загружаемых по мере необходимости. С одной стороны, это несколько замедляет работу ядра, а с другой — нет необходимости пересобирать ядро при установке какого-нибудь нового аппаратного устройства.

Монолитное ядро — это ядро, в котором все необходимые драйверы в него вкомпилированы. Недостатком такого ядра является необходимость пересобирать ядро операционной системы при появлении новых аппаратных устройств, а достоинством — сравнительно небольшой объем ядра операционной системы и повышенная ее производительность. Мы рассмотрим компиляцию и модульного ядра, и монолитного.

Для чего пересобирают ядро? Каковы аргументы "за" и "против"?

"За" компиляцию ядра операционной системы

Рассмотрим аргументы "за".

- Основная причина для самостоятельной компиляции ядра — *выход новых версий ядра* операционной системы Linux. Как правило, в новых версиях добавляются новые функциональные возможности и исправляются замеченные ошибки. К сожалению, большинство сборщиков дистрибутивов отстают в выпусках "фирменных" ядер операционной системы, а иногда даже делают доступным новую версию ядра операционной системы только с выходом новой версии дистрибутива.
- Пересборка ядра используется *для оптимизации ядра операционной системы* Linux конкретно под имеющийся набор аппаратных средств — процессора, чипсета материнской платы, контроллеров, сетевых плат, видеокарт и т. п.
- Компиляцию ядра осуществляют так же *для включения специфических свойств ядра*, которые появляются в нем после наложения специальных "заплаток" — (патчей), разрабатываемых отдельными программистами или группами. Обычно эти свойства связаны с безопасностью системы или с функционированием экзотических аппаратных средств. Наиболее известен вариант "альтернативного" ядра от Алана Кокса (Alan Kox).
- Самостоятельно компилировать *экспериментальные ядра с новыми возможностями* приходится потому, что разработчики дистрибутивов их не тестируют и не выпускают с ними инсталляционные пакеты. Следует

помнить, что экспериментальные ядра не всегда стабильны, и с новыми возможностями можно получить набор ошибок, правда зачастую не критичных для функционирования системы.

"Против" компиляции ядра операционной системы

Против компиляции ядра операционной системы есть столько же пунктов (если не больше), сколько и "за".

- Для того чтобы скомпилировать ядро операционной системы, необходимо много знать из различных областей администрирования: особенности настройки и функционирования сетей, поддержка аппаратуры, периферии, файловых систем, специфического программного обеспечения и др.
- При неправильно сконфигурированном или неверно установленном новом ядре операционной системы (это вытекает из предыдущего пункта) получаем проблемы вплоть до полной потери работоспособности операционной системы.
- Это противоречит принципу "работает — не трогай". Если ядро операционной системы работает устойчиво и его функционирование всех удовлетворяет — зачем его компилировать?
- Если вы решили уменьшить объем ядра операционной системы, изъяв все лишнее, не забудьте: почти все свойства ядра операционной системы вынесены в загружаемые модули, поэтому сэкономить удастся только пятьдесят-сто килобайт.

Утилиты конфигурирования ядра операционной системы Linux

Если вы все-таки решились скомпилировать ядро операционной системы Linux, перед вами встанет вопрос — как сконфигурировать его параметры? Можно, конечно, изучить исходные тексты ядра операционной системы, но это займет не менее одной-двух недель. Есть и более легкие способы — сделать это при помощи специальных утилит:

- xconfig — утилита по конфигурированию параметров ядра в графической системе X Window (рис. 16.1). Как можно видеть из рисунка — простая, понятная, легкая в использовании;
- menuconfig — простая текстовая утилита с системой меню для конфигурации ядра операционной системы (рис. 16.2). Передвигаясь по пунктам меню, достаточно удобно настраивать ядро операционной системы;

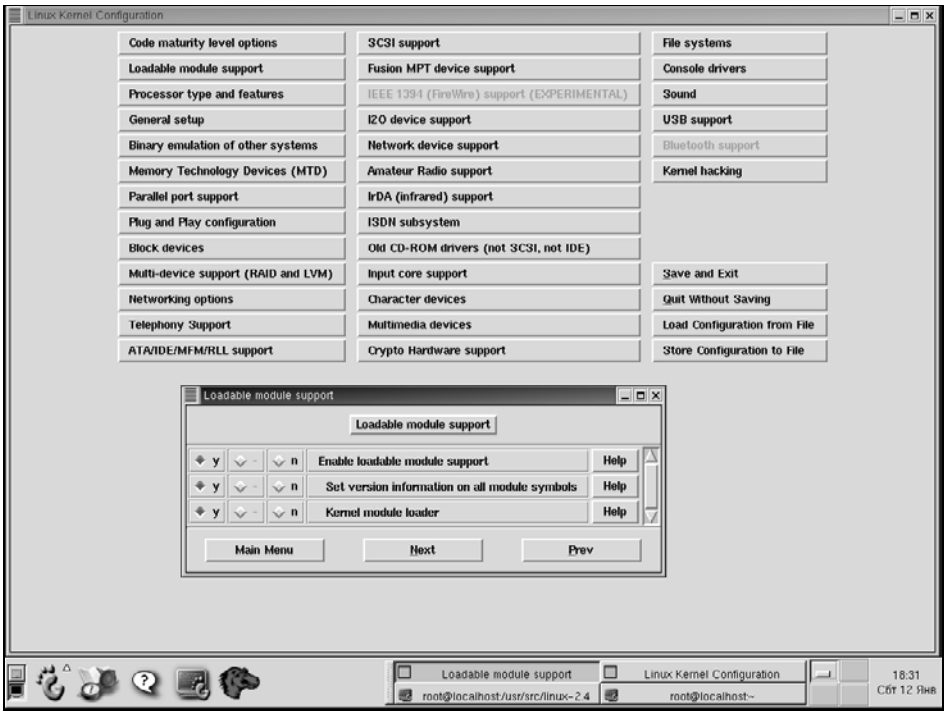

Рис. 16.1. xconfig

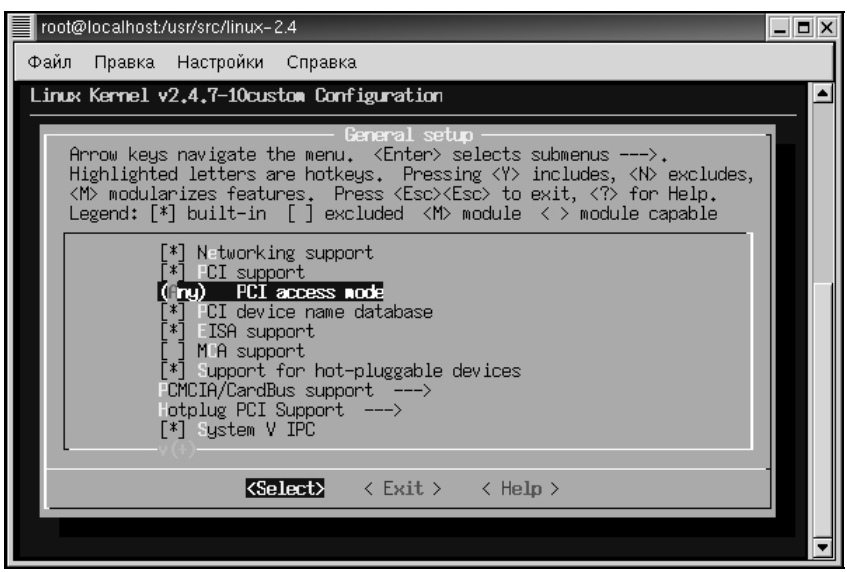

Рис. 16.2. menuconfig

- config — старейшая текстовая утилита для конфигурирования ядра операционной системы (рис. 16.3). Требует минимального количества ресурсов и библиотек, однако неудобна в эксплуатации. Нельзя вернуться назад и отредактировать предыдущие пункты, необходимо закончить конфигурирование (или прервать) и начать его заново. На сегодняшний день практически не применяется.

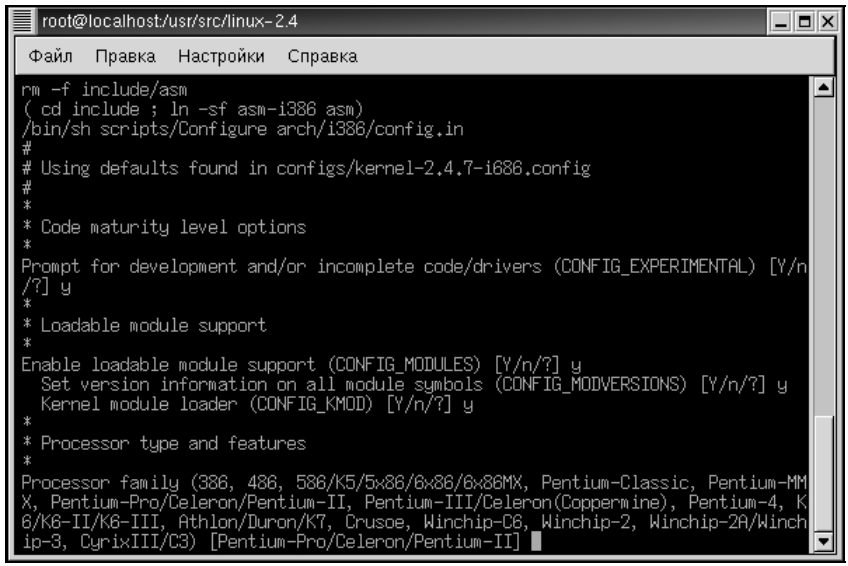

Рис 16.3. config

Процесс компиляции ядра

Как обычно, к компиляции ядра операционной системы следует предварительно подготовиться — создать загрузочную дискету. Для этого зайти в систему пользователем root и выполнить команду:

```
/sbin/mkbootdisk kernelversion
```

где `kernelversion` — версия ядра, полученная с помощью команды `uname`.

Далее необходимо получить из Интернета (или с инсталляционного компакт-диска) и установить пакеты kernel-headers и kernel-source.

После этого перейти в каталог /usr/src/linux-2.4 и все дальнейшие команды выполнять из этого каталога.

Ядро с поддержкой загружаемых модулей (модульное)

Компиляция ядра операционной системы Linux происходит в несколько этапов.

1. Необходимо быть уверенным, что после предыдущих компиляций ядра операционной системы в дереве исходных кодов не осталось каких-либо несуразностей типа исходных текстов ядра младшей версии или неубранных объектных файлов. Поэтому компиляцию ядра операционной системы рекомендуется начать с команды `make mrproper`. Эта команда помимо того, что удаляет "мусор" после предыдущих компиляций, уничтожает также конфигурационный файл ядра, который находится в /usr/src/linux-2.4/.config. Если у вас уже есть рабочий конфигурационный файл (/usr/src/linux-2.4/.config) который вы хотите взять за основу, перед выполнением этой команды скопируйте его в свой домашний каталог, а после выполнения команды `make mrproper` верните на прежнее место.
2. Теперь следует произвести конфигурацию ядра операционной системы. Если у вас уже есть готовый конфигурационный файл — пропустите этот шаг. Для конфигурирования ядра операционной системы, как мы уже указывали выше, можно воспользоваться четырьмя разными утилитами, приводящими к одному результату:
   - `make xconfig`. Утилита используется для конфигурирования ядра операционной системы в среде X Window;
   - `make config`. Простая текстовая утилита конфигурации ядра операционной системы;
   - `make menuconfig`. Текстовая утилита с системой меню для конфигурации ядра операционной системы Linux;
   - `make oldconfig`. Неинтерактивный скрипт, который устанавливает в конфигурационном файле ядра значения по умолчанию.
3. После создания конфигурационного файла /usr/src/linux-2.4/.config для корректной установки всех зависимостей выполняем команду `make dep`.
4. Для подготовки исходных текстов для компиляции выполняем команду `make clean`.
5. Теперь необходимо отредактировать файл /usr/src/linux-2.4/Makefile таким образом, чтобы полученное новое ядро не перекрыло старое ядро операционной системы (более подробную информацию смотрите в Kernel-HOWTO). Редактируем /usr/src/linux-2.4/Makefile и исправляем строку, начинающуюся с `EXTRAVERSION=` таким образом, чтобы создать уникальное имя. Самый простой вариант — добавить дату компиляции ядра. К примеру `EXTRAVERSION= -0.1.21-jul2001`. Это позволит одновременно иметь старую и новую версии ядра операционной системы.

6. Компилируем ядро операционной системы командой `make bzImage`.
7. Компилируем модули ядра операционной системы командой `make modules`.
8. Устанавливаем модули операционной системы командой `make modules_install`. Эта команда должна установить модули ядра в каталог /lib/modules/KERNELVERSION/kernel/drivers, где KERNELVERSION — версия, описанная в файле Makefile. В нашем примере это /lib/modules/2.4.7-3-jul2001/kernel/drivers/.
9. Если в вашей системе установлен SCSI-контроллер, и вы сделали SCSI-драйвер модульным, необходимо создать новый файл initrd (см. далее).
10. Выполняем команду `make install` для того, чтобы скопировать наше новое ядро операционной системы и необходимые файлы в соответствующие каталоги.
11. Ядро успешно скомпилировано и установлено. Далее необходимо произвести конфигурирование загрузчика (*см. разд. "Конфигурирование загрузчика"*).

Создание образа initrd

Файл initrd необходим для загрузки SCSI-модуля во время старта операционной системы. Скрипт /sbin/mkinitrd создает соответствующий образ initrd для вашего компьютера, если выполнены следующие условия:

- ❒ loopback block device доступно;
- ❒ файл /etc/modules.conf содержит описание вашего SCSI-контроллера.

Для построения образа initrd необходимо выполнить команду `/sbin/mkinitrd` со следующими параметрами: `/sbin/mkinitrd /boot/initrd-2.4.7-3-jul2001.img 2.4.7-3-jul2001`

Здесь `/boot/initrd-2.4.7-3-jul2001.img` — имя файла для нового образа initrd, а `2.4.7-3-jul2001` — ядро, чьи модули (из /lib/modules) должны быть использованы при создании образа initrd.

Этапы компиляции

Подведем итог. Для компиляции и инсталляции модульного ядра операционной системы Linux необходимо выполнить следующие команды:

1. `make mrproper`.
2. `make menuconfig`.
3. `make dep`.
4. `make clean`.
5. Редактирование /usr/src/linux-2.4/Makefile.

6. `make bzImage`.
7. `make modules`.
8. `make modules_install`.
9. `/sbin/mkinitrd /boot/initrd-2.4.xx.img 2.4.xx` (если в вашей системе установлен SCSI-контроллер).
10. `make install`.
11. Конфигурирование загрузчика.

Монолитное ядро

Компиляция монолитного ядра операционной системы в основном повторяет компиляцию модульного ядра за некоторыми небольшими исключениями:

- когда конфигурируется ядро, не должны использоваться модули, т. е. на любой вопрос надо отвечать только `Yes` или `No`. Так же необходимо ответить `No` для пунктов kmod support и module version (CONFIG_MODVERSIONS) support;
- необходимо пропустить следующие команды:
  - `make modules`;
  - `make modules_install`;
- для загрузчика LILO в файл lilo.conf необходимо добавить строчку `append=nomodules`.

Этапы компиляции

Подведем итог. Для компиляции и инсталляции монолитного ядра операционной системы Linux необходимо выполнить следующие команды:

1. `make mrproper`.
2. `make menuconfig`.
3. `make dep`.
4. `make clean`.
5. Редактирование /usr/src/linux-2.4/Makefile.
6. `make bzImage`.
7. `/sbin/mkinitrd /boot/initrd-2.4.xx.img 2.4.xx` (если в вашей системе установлен SCSI-контроллер).
8. `make install`.
9. Конфигурирование загрузчика *(см. разд. "Конфигурирование загрузчика")*. Для загрузчика LILO в файл lilo.conf необходимо добавить строчку `append=nomodules`.

Параметры настройки ядра

Этот раздел полностью посвящен параметрам настройки ядра операционной системы Linux. Структура раздела выглядит следующим образом — в первой части на основе утилиты menuconfig показано дерево параметров настройки ядра Linux, во второй части — краткие пояснения параметров.

Дерево параметров настройки ядра

Полное дерево настроек ядра Linux с установками, используемыми в дистрибутивном ядре Red Hat Linux 7.2, приведено в *приложении 4*.

Параметры настройки ядра (комментарии)

Как вы уже заметили, параметров настройки ядра много, и чтобы правильно их сконфигурировать, необходимо иметь обширные знания о функциях операционной системы Linux. Еще одна особенность — почти 90% всех настроек и свойств ядра вынесены в модули. Если вы не уверены в том или ином свойстве — поставьте "использовать модуль".

Далее кратко прокомментируем основные пункты меню конфигурации ядра операционной системы:

- Code maturity level options — для использования альфа- или нестабильных версий драйверов. В основном с целью отладки;
- Loadable module support — отвечает за вид скомпилированного ядра операционной системы: модульное или монолитное;
- Processor type and features — определяет тип процессора, для которого компилируется ядро операционной системы, поддержку набора процессорных команд, поддержку мультипроцессорной системы, объем поддерживаемой оперативной и виртуальной памяти и некоторые другие параметры. Для максимальной производительности системы рекомендуется выбрать именно тот тип процессора, который установлен в вашей системе, однако с целью совместимости в дистрибутиве ядро компилируется таким образом, чтобы оно работало на любом процессоре — от Pentium до Pentium 4, AMD и Cyrix;
- General setup — определяет основные свойства ядра: что оно сможет делать и какие типы устройств будет поддерживать. Здесь выбирается поддержка сети, шин PCI, EISA, MCA, PCMCIA-устройств, различных форматов исполняемых файлов и т. п;
- Binary emulation of other systems — определяет, для каких операционных систем поддерживается выполнение бинарных файлов. В основном, это поддержка бинарных файлов для UnixWare, Solaris, SCO Unix;

- Memory Technology Devices (MTD) — данные устройства (твердотельные накопители) для нас еще достаточно экзотичны;
- Parallel port support — отвечает за поддержку ядром параллельного порта;
- Plug and Play configuration — поддержка Plug and Play, в том числе и ISA-устройств;
- Block devices — определяется поддержка различных блоковых устройств и контроллеров, в том числе дисководов, жестких дисков старого (XT) типа или жестких дисков, подключаемых к параллельному порту. Здесь же включается поддержка Loopback-устройства, сетевого блочного устройства и диска, создаваемого в оперативной памяти (RAM Disk). С последним надо быть очень аккуратным, как вы знаете, он используется при загрузке операционной системы;
- Multi-device support (RAID and LVM) — включается поддержка RAID-устройств и LVM (управление массивом дисков);
- Networking options — поскольку одним из важнейших назначений для операционной среды Linux является работа с сетью, то, наверное, половина параметров ядра тем или иным образом касается сети. Здесь определяются различные сетевые параметры — используемые сетевые протоколы, сетевая маршрутизация, конфигурирование виртуального сервера и многое другое;
- Telephony Support — отвечает за поддержку плат IP-телефонии — специализированных устройств для преобразования телефонного звонка в цифровое представление и обратно. Подавляющее число пользователей вряд ли даже знают названия этих карт, не говоря уже об использовании;
- ATA/IDE/MFM/RLL support — определяются поддерживаемые устройства и контроллеры ATA/IDE, а так же устаревшие устройства MFM/RLL. Здесь же есть параметры для исправления ошибок различных чипов контроллеров ATA/IDE, большей частью времен 386 и 486 процессоров. Поскольку у большинства пользователей в компьютере присутствует хотя бы одно устройство ATA/IDE — нужно быть очень внимательным при конфигурировании этого раздела. В этом же разделе включается поддержка модных ныне аппаратных контроллеров IDE RAID;
- SCSI support — определяет поддержку SCSI-устройств и контроллеров. Поскольку SCSI-устройства сложнее IDE-устройств, и каждый производитель контроллеров старался "отличиться", этот раздел получился большим и несколько запутанным. Здесь определяется поддержка дисков, ленточных накопителей, CD-ROM и RAID-массивов;
- Fusion MPT device support — поддержка устройства Fusion MPT — объединенного высокоскоростного SCSI-адаптера и сетевого интерфейса. Работает по протоколу Ultra 320 или Fibre Channel;

- IEEE 1394 (FireWire) support (EXPERIMENTAL) — поддержка контроллеров FireWire;
- I2O device support — поддержка Intelligent Input/Output. Используется для ввода/вывода без участия процессора;
- Network device support — поддержка различных сетевых устройств, в том числе протокола PPP, SLIP, оптических устройств, беспроводных сетей и т. п.;
- ARCnet devices — протокол и устройства, разработанные достаточно давно. Практически вышли из употребления;
- Appletalk devices — устройства для сетей Appletalk фирмы Apple. В России широкого распространения не имеют;
- Ethernet (10 or 100Mbit) — поддержка сетевой карты Ethernet. Представлены карты и семейства карт различных производителей, работающие на скорости 10 или 10/100 Мбит;
- Ethernet (100 Mbit) — представлены сетевые карты Ethernet, работающие на скорости 100 Мбит;
- Wireless LAN (non-hamradio) — осуществляется поддержка устройств типа Radio Ethernet;
- Token Ring devices — устройство, разработанное фирмой IBM достаточно давно. Большого распространения (по крайней мере у нас) не получило. Сегодня практически не используется;
- Wan interfaces — поддержка интерфейсов глобальных сетей;
- ATM drivers — поддержка для ATM-сетей. Очень распространены в США для организации связи на большом расстоянии;
- Amateur Radio support — поддержка любительского радио. Не представляет интереса для подавляющего числа пользователей (если вы, конечно, не радиолюбитель);
- IrDA (infrared) support — поддержка устройств и инфракрасных портов. Обычно используется для связи с портативными принтерами, органайзерами и мобильными телефонами;
- ISDN subsystem — поддержка ISDN-устройств. В основном используется для подключения к цифровым телефонным станциям;
- Old CD-ROM drivers (not SCSI, not IDE) — поддержка старых CD-ROM, работающих по специфическому протоколу. Подобные устройства не выпускаются с 1996 года (максимальная скорость устройств — 4^{x}—6^{x});
- Input core support — поддержка USB Human Interface Device (в частности USB-клавиатуры и мыши);
- Character devices — поддержка символьных устройств: терминалов, последовательных портов и многопортовых контроллеров, аппаратных датчиков, мышей, джойстиков и манипуляторов, видеокарт и чипсетов;

- Multimedia devices — поддержка мультимедиа-устройств. В данном случае это платы TV- и FM-приемников, платы телетекста;
- Crypto Hardware support — поддержка устройств аппаратного шифрования (для закрытых учреждений);
- File systems — поддержка различных файловых систем (VFAT, Ext3, ISO 9660 и т. п.), сетевых файловых систем, поддержка квотирования дискового пространства для пользователей, типов разделов жесткого диска. Помимо этого, сюда же вынесена поддержка различных языковых кодировок;
- Console drivers — консольные драйверы, поддержка ряда видеокарт;
- Sound — поддержка звуковых карт. Из-за того, что до последнего времени не было единого стандарта на звуковой интерфейс, почти каждая карта была в своем роде уникальна;
- USB support — поддержка USB-контроллеров и устройств, в том числе Flash-накопителей, мышей, видеокамер и т. п.;
- Bluetooth support — поддержка нового Bluetooth-интерфейса (радиосеть);
- Kernel hacking — используется для отладки ядра операционной системы Linux.

Ссылки

- Red Hat Linux 7.2 The Official Red Hat Linux Customization Guide.
- The Linux Kernel on Red Hat Linux Systems — обновление ядра операционной системы.
- **www.gnu.org/software/grub/** — домашняя страница GRUB.
- **www.redhat.com/support/docs/howto/kernel-upgrade/kernel-upgrade.html** — Upgrading.
- /usr/src/linux-2.4/Documentation — большой объем документации, посвященный ядру операционной системы Linux и ее модулям.
- Kernel-HOWTO (The Linux Kernel HOWTO) — описание конфигурирования и компиляции ядра *(см. гл. 13)*.

Глава 17

DNS

DNS — это Доменная Система Имен (Domain Name System). DNS преобразует символические имена машин в IP-адреса и наоборот — из IP-адреса в символическое имя. Для чего это нужно? Во-первых, человеку легче запомнить осмысленное имя — типа vasya.ru чем 195.66.195.42, а для компьютера проще передать четыре байта адреса, чем 50—60 байтов имени. Во-вторых, за одним и тем же IP-адресом могут скрываться сотни различных доменов. Когда-то, на заре эры глобальных сетей, все пары "имя-IP-адрес" хранились в файле /etc/hosts. Со временем, когда компьютеров в сети стало тысячи и десятки тысяч, эти файлы превратились в монстров, на смену которым пришли DNS-серверы.

DNS-сервер представляет собой базу данных, в которой для тысяч компьютеров хранится соответствие символического имени компьютера IP-адресу. В сети существуют десятки тысяч серверов DNS, которые обмениваются информацией с другими серверами DNS.

DNS — это иерархическая система. Вершина записывается как "." (точка) и произносится как root (корень). В корне существует некоторое количество доменов верхнего уровня (Top Level Domains, TLDs), наиболее известными из которых являются ORG, COM, EDU, GOV, MIL, NET, RU, UA и т. п.

При поиске машины запрос обрабатывается рекурсивно, начиная с корня. Если нужно найти адрес машины user.ogpu.odessa.ua, то ваш сервер имен должен найти сервер имен, который обслуживает ua. Он запрашивает корневой сервер (.), который выдает список серверов ua. Из полученного списка выясняется, какие серверы имен обслуживают ua. Затем запрашивается сервер (выбирается по определенному алгоритму или берется первый в полученном списке) чтобы узнать, какие серверы обслуживают odessa.ua. Затем берется сервер из полученного списка и выясняется, кто обслуживает ogpu.odessa.ua, и уже у этого сервера узнается IP-адрес компьютера user.ogpu.odessa.ua. А чтобы в следующий раз не повторять этот поиск, полученную пару "имя—IP-адрес" ваш сервер DNS сохраняет в своей базе данных.

В том случае, если необходимо по IP-адресу узнать имя компьютера, опять используется DNS-сервер. Для этих целей существует псевдодомен in-addr.arpa и в нем точно так же прописываются адреса, только порядок следования цифр обратный. Например, для адреса 195.66.195.22 запрос получится как к 22.195.66.195.in-addr.arpa, а схема поиска ответа остается такая же.

По своим функциональным обязанностям различают два вида DNS-серверов — обычный и кэширующий.

- *Кэширующий сервер* DNS используется для локального хранения запрошенных пользователем пар "имя—IP-адрес", что при интенсивном общении со многими Web-серверами позволяет экономить время на DNS-запросах. Кэширующий сервер не отвечает на внешние DNS-запросы.
- *обычный сервер* DNS — это полнофункциональный сервер, позволяющий получать, передавать и синхронизировать DNS-данные с другими DNS-серверами.

Настройка сетевых параметров

Поскольку настройка (и функционирование) DNS-сервера затрагивает практически все сетевые параметры, работоспособность DNS-сервера зависит от правильной конфигурации сети. В современных дистрибутивах, если вы выбрали "устанавливать DNS-сервер", конфигурирование его производится автоматически. Однако разработчик дистрибутива рассчитывает на абстрактную среднестатистическую систему, которой, как показывает практика, не существует. Поэтому следует убедиться, что с сетевыми настройками у вас все в порядке.

host.conf

Следующая запись в файле host.conf означает, что при поиске хостов система сначала посмотрит в /etc/hosts, а потом только обратится к серверу DNS:

```
order hosts,bind
```

/etc/hosts

В этом файле должны находиться пары "IP-адрес—имя":

```
127.0.0.1     localhost localhost.localdomain
192.168.0.1   user
192.168.0.2   user2
```

Причем обязательно должна присутствовать следующая строка:

```
127.0.0.1     localhost localhost.localdomain
```

/etc/resolf.conf

В этом файле должны находиться строки, подобные приведенным:

```
search ogpu.odessa.ua
nameserver 195.66.195.22
```

В строке, которая начинается со слова `search`, указывается, какое доменное имя будет принято по умолчанию. Так, если вы напишете user, то система сразу попытается обратиться к компьютеру user.ogpu.odessa.ua. После `search` можно указывать несколько имен. В следующей строчке указываются адреса DNS-серверов, к которым будет обращаться ваша машина.

Настройка кэширующего сервера

Кэширующий сервер найдет ответ на запрос об имени машины и запомнит его, чтобы ответить, когда вы запросите эту же информацию в следующий раз. Это значительно уменьшит время ожидания ответа при следующем запросе, особенно если у вас медленное соединение.

/etc/named.conf

Это основной конфигурационный файл для DNS-сервера. Для кэширующего сервера он должен содержать следующие строки:

```
options {
        directory "/var/named";
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "127.0.0";
};
```

Строка `directory` указывает `bind` где искать файлы. Все файлы, используемые впоследствии, будут иметь путь относительно этого каталога.

Строка `zone "0.0.127.in-addr.arpa"` показывает, что `bind` также отвечает за обратную зону для подсети 127.*.*.*, является в ней мастером, и файл описания зоны — 127.0.0.

Секция `zone "."` самая важная. Она описывает, в каком файле лежат адреса корневых DNS-серверов, которые отвечают за зоны первого уровня.

Файл, названный /var/named/root.hints, должен находиться в указанном каталоге и содержать приблизительно следующую информацию:

```
.                       6D IN NS        G.ROOT-SERVERS.NET.
.                       6D IN NS        J.ROOT-SERVERS.NET.
.                       6D IN NS        K.ROOT-SERVERS.NET.
.                       6D IN NS        L.ROOT-SERVERS.NET.
.                       6D IN NS        M.ROOT-SERVERS.NET.
.                       6D IN NS        A.ROOT-SERVERS.NET.
.                       6D IN NS        H.ROOT-SERVERS.NET.
.                       6D IN NS        B.ROOT-SERVERS.NET.
.                       6D IN NS        C.ROOT-SERVERS.NET.
.                       6D IN NS        D.ROOT-SERVERS.NET.
.                       6D IN NS        E.ROOT-SERVERS.NET.
.                       6D IN NS        I.ROOT-SERVERS.NET.
.                       6D IN NS        F.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
```

Этот файл описывает имена корневых серверов имен по всему миру. Их список периодически изменяется. Поэтому данный файл необходимо время от времени корректировать.

Для получения файла root.hints существует по меньшей мере два пути: либо забрать его по FTP с сервера internic, либо выполнить команду:

```
dig @rs.internic.net . ns >root.hints
```

/etc/127.0.0

127.0.0 — это файл, который отвечает за преобразование чисел IP-адреса в имена.

Файл 127.0.0 должен выглядеть так:

```
@            IN      SOA   ns.ogpu.odessa.ua. hostmaster.ogpu.odessa.ua. (
                                  1       ; Serial
                                  8H      ; Refresh
                                  2H      ; Retry
                                  1W      ; Expire
                                  1D)     ; Minimum TTL
                        NS        ns.ogpu.odessa.ua.
1                       PTR       localhost.
```

Эта запись обозначает следующее:

- `@` указывает, что описываем сами себя;
- описываемая зона держится сервером с именем ns.ogpu.odessa.ua;
- отвечает за нее человек, доступный по адресу **hostmaster@ogpu.odessa.ua** (первая точка заменяет `@`);
- у зоны серийный номер равен 1 (обычно для него используют дату последней правки зоны — на него опираются другие серверы, которые берут информацию с вашего);
- другие серверы будут обновлять информацию о вашем с периодичностью в восемь часов;
- при неудачном обновлении следующая попытка будет произведена через два часа;
- зона будет считаться содержащей недостоверную информацию на кэширующих серверах через одну неделю;
- зона будет считаться содержащей недостоверную информацию на кэширующих серверах не менее, чем через один день;
- строка `IN NS ns.ogpu.odessa.ua.` показывает, что авторитетным сервером за эту зону является `ns.ogpu.odessa.ua.`, и именно ему надо рассылать обновления зоны ns.ogpu.odessa.ua.;
- строка `1 PTR localhost.` описывает что машина с адресом 1 в зоне 127.0.0. имеет имя localhost.

Запуск named

Теперь можно запускать сервер. Наберите `ndc start` без опций и нажмите клавишу <Enter>.

Затем запускаем программу nslookup:

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

>_
```

Если на мониторе это выглядит так, значит система работает. Мы так надеемся. Если вы видите что-то другое, то вернитесь назад и все проверьте. Каждый раз, когда вы изменяете файл named.conf, необходимо перезапустить named, используя команду `ndc restart`.

Теперь проверим, как функционирует ваш кэширующий сервер — введем `user7.ogpu.odessa.ua`:

```
> user7.ogpu.odessa.ua
Server:  localhost
Address:  127.0.0.1

Name:    user7.ogpu.odessa.ua
Address:  195.66.195.31
```

При этом nslookup попросил ваш named посмотреть информацию о данном компьютере. Если вы повторно попросите узнать адрес компьютера user7.ogpu.odessa.ua, то получите такой ответ:

```
> user7.ogpu.odessa.ua
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:    user7.ogpu.odessa.ua
Address:  195.66.195.31
```

В это раз вы получили сообщение "Non-authoritative answer". Это значит, что named во второй раз не делал запрос к внешним серверам имен, а произвел поиск в своем кэше и нашел нужную запись. Поскольку вы увидели это сообщение, ваш кэширующий DNS-сервер функционирует верно. Получив положительный результат, можно завершить работу nslookup дав команду `exit`.

Настройка DNS-сервера

Настройка полнофункционального DNS-сервера несколько сложнее, чем кэширующего, но в основном файлы и записи те же самые. Для чистоты

эксперимента рекомендуется произвести настройку для несуществующего домена. У нас он будет называться ivan.petrov.

/etc/named.conf

Для нашего сервера DNS он должен содержать следующие строки:

```
options {
        directory "/var/named";
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "127.0.0";
};
zone "ivan.petrov" {
        notify no;
        type master;
        file "ivan.petrov";
};
zone "0.168.192.in-addr.arpa" {
        notify no;
        type master;
        file "192.168.0";
};
```

Как можно видеть, по сравнению с кэширующим сервером добавилась только секция `zone "ivan.petrov"` и секция `zone "0.168.192.in-addr.arpa"`.

Секция `zone "ivan.petrov"` описывает, что наш DNS-сервер держит зону ivan.petrov, является в ней мастером `type master;` (то есть другие серверы лишь синхронизируют по нему свои записи по зоне ivan.petrov), при изменении записей в зоне не извещает другие сервера `notify no;` (засорять Интернет вашими тестовыми записями нет нужды) и использует для описания зоны файл ivan.petrov.

Секция `zone "0.168.192.in-addr.arpa"` описывает, что наш DNS-сервер держит реверсную зону 0.168.192.in-addr.arpa, является в ней мастером `type master;` (то есть другие серверы лишь синхронизируют по нему свои записи

по зоне 0.168.192.in-addr.arpa), при изменении записей в зоне не извещает другие серверы `notify no;` и использует для описания зоны файл 192.168.0.

/etc/named/ivan.petrov

В файле зоны ivan.petrov поместим следующие данные:

```
@   IN   SOA   ns.ivan.petrov. hostmaster.ivan.petrov. (
               199802151       ; serial, todays date + todays serial #
               8H              ; refresh, seconds
               2H              ; retry, seconds
               1W              ; expire, seconds
               1D )            ; minimum, seconds
;
    NS      ns                      ; интернет-адрес сервера имен
    MX      10 mail.ivan.petrov.    ; Основной почтовый сервер
    MX      20 mail2.ivan.petrov.   ; Дополнительный почтовый сервер
;
localhost       A       127.0.0.1
ns              A       192.168.0.1
mail            A       192.168.0.40
```

Этот файл зоны содержит 4 записи ресурсов (Resource Records, RR):

- **SOA RR**. SOA — это сокращение для слов Начала Полномочий (Start Of Authority). Запись SOA находится в преамбуле каждого из файлов зон, и она должна быть первой записью в файле. Она описывает зону — откуда она появляется (машина, названная ns.ivan.petrov), кто отвечает за содержимое зоны (**hostmaster@ivan.petrov**), какая версия файла зоны текущая (serial: 1) и другие вещи, которые надо сделать для кэширующих и вторичных серверов DNS;
- **NS RR**. NS — это RR для сервера имен (Name Server);
- **MX RR**. MX — запись ресурса Почтовый Сервер (Mail eXchanger). Запись MX сообщает почтовой системе, куда посылать почту, адресованную любому адресату в домене ivan.petrov, в нашем случае — серверам mail. ivan.petrov или mail2. ivan.petrov. Число перед каждым именем системы — это приоритет записи MX RR. Запись ресурса с наименьшим номером (10) — это компьютер, куда почта должна посылаться в первую очередь. Если происходит ошибка, то почта может быть послана на машину с большим номером. И так далее. Таким образом, можно указать несколько почтовых серверов, что поможет вам в случае форс-мажорных обстоятельств не потерять ваши почтовые сообщения;

- **A RR.** A (Address) — адрес (IP-адрес)

```
localhost       A       127.0.0.1
ns              A       192.168.0.1
mail            A       192.168.0.40
```

Эти строки устанавливают соответствие IP-адресам имен mail и ns в зоне ivan.petrov.

/etc/192.168.0

Для нормального функционирования DNS-сервера требуется обратная (реверсная) зона, которая дает возможность DNS преобразовывать IP-адреса в имена хостов. Эти имена используются серверами различного рода (FTP, IRC, WWW и т. п.) Поэтому обратная зона требуется для полного доступа к различным сервисам в Интернете.

Далее представлен файл /etc/192.168.0:

```
@       IN      SOA     ns.ivan.petrov. hostmaster. ivan.petrov. (
                        199802151 ; Serial, todays date + todays serial
                        8H      ; Refresh
                        2H      ; Retry
                        1W      ; Expire
                        1D)     ; Minimum TTL
                NS      ns.linux.bogus.

2               PTR     gw.ivan.petrov.
1               PTR     ns.ivan.petrov.
3               PTR     petya.ivan.petrov.
40              PTR     mail.ivan.petrov.
5               PTR     ftp.ivan.petrov.
```

Вышеприведенный файл в принципе мало чем отличается от файла описания прямой зоны. Появились только следующие строки:

```
2               PTR     gw.ivan.petrov.
1               PTR     ns.ivan.petrov.
3               PTR     petya.ivan.petrov.
40              PTR     mail.ivan.petrov.
5               PTR     ftp.ivan.petrov.
```

Эти строки описывают, что машина с адресом 2 в зоне 192.68.0. имеет имя gw.ivan.petrov, а компьютер с адресом 40 — mail.ivan.petrov.

Вот собственно и все. Перезапускаем сервер и проверяем правильность функционирования нашей системы.

Некоторые тонкости

Как вы видите, глубоко в тонкости функционирования мы не погружались. Во-первых, этого вполне достаточно, а во-вторых, решать проблемы следует по мере их возникновения. Тем не менее, несколько полезных вещей необходимо знать.

Записи ресурсов (RR) службы DNS

Давайте рассмотрим несколько расширенный файл описания зоны:

```
gw              A       192.168.0.2
                HINFO   "i486" "RH 4.2"
                TXT     "The router"

ns              A       192.168.196.1
                MX      10 mail
                HINFO   "Pentium3" "RH 7.2"
www             CNAME   ns
donald          A       192.168.196.3
                MX      10 mail
                HINFO   "p3"  "Windows2000"
                TXT     "Developer computer home tel 223344"
```

Помимо знакомых вам строчек появились строки, содержащие `HINFO`, `CNAME` и `TXT`.

- `HINFO` — информация о компьютере (Host INFOrmation) состоит из двух частей: первая часть — это информация об оборудовании машины, а вторая — описывает программное обеспечение и операционную систему данной машины. Помимо этой информации не рекомендуется вносить ничего другого. Пример:

  ```
  HINFO   "Pentium3" "RH 7.2"
  ```

  Из этой строки видно, что наш DNS-сервер собран на базе процессора Pentium III и на нем установлена операционная система Linux Red Hat 7.2;

- `CNAME` — каноническое имя (Canonical NAME) — это способ присвоить каждой машине несколько имен. При использовании `CNAME` необходимо следовать правилу, что записи `MX`, `CNAME` или `SOA` *никогда* не должны ссылаться на имя, указанное как запись `CNAME`;

- `TXT` — произвольная текстовая информация. Обычно используется в качестве расширенного комментария для описания хоста. Пример:

  ```
  TXT     "Developer computer home tel 223344"
  ```

Из содержимого строчки понятно, что это компьютер разработчика, а его домашний телефон — 223344.

Существует еще один тип записи — RP (Responsible Party, группа ответственных). В принципе эта же информация может храниться и в записях TXT, однако применение записи RP ускоряет поиск данных об ответственных лицах. Список основных записей ресурсов службы DNS приведен в табл. 17.1.

Таблица 17.1. *Основные записи ресурсов (RR) службы DNS*

Обозначение записи	Содержание записи	Номер RFC или автор проекта
A	IP-адрес хоста	RFC1035
AAAA	Адрес IPv6	Проект, автор Thomson
CNAME	Каноническое имя домена	RFC1035
GPOS	Географическое положение	RFC1712
HINFO	Информация о хосте (процессор и ОС)	RFC1035
ISDN	Адрес ISDN	RFC1183
KEY	Ключ шифрования	(проект, автор Eastlake)
LOC	Расположение	(проект, автор Vixie)
MX	Имя хоста или домена для переадресации почты	RFC1035
NSAP	SAP-адрес (адрес A в формате NSAP)	RFC1706
NSAP-PTR	Аналог записи PTR для адреса NSAP	RFC1706
NULL	Пустая запись ресурса	RFC1035 (экспериментальный стандарт)
NXT	Следующий домен	(проект, автор Eastlake)
PTR	Указатель на имя домена	RFC1035
RP	Ответственные лица	RFC1183
SIG	Цифровая подпись	(проект, автор Eastlake)
SRV	Выбор сервера	(проект, автор Vixie)
TXT	Произвольный текст	RFC1035
WKS	Описание подключенных сервисов	RFC1035
X25	Адрес X.25	RFC1183

Реверсная зона

Не забывайте об обратной (реверсной) зоне! Очень неприятно, когда по этой причине вы не сможете воспользоваться FTP-сервером или получите сообщения о нарушениях системы защиты. Не поленитесь — потратьте час на описание реверсной зоны.

Два сервера DNS

Существует множество причин, по которым не желательно раскрывать всю информацию о вашей сети через службу DNS. Поэтому рекомендуется создать два сервера DNS: один для внутренних пользователей, другой для внешних. Для этого необходимо обеспечить два различных набора IP-адресов: для внутренних клиентов и для внешнего мира.

Иерархические поддомены

Если в вашей организации используется более одной подсети, то вам придется задать несколько доменов in-addr.arpa. Создание поддоменов, подчиненных первичному домену, целесообразно также при наличии в вашей организации нескольких отделов или подразделений. Это облегчит мониторинг сети, а также упростит организацию доступа в сеть и установку защитных фильтров. Конечно, если ваша сеть состоит всего из нескольких машин, смысла в создании иерархии доменов просто нет.

Вторичные DNS-серверы

Если у вас большая сеть или если вы занимаетесь хостингом сайтов, вы обязательно должны помимо первичного сервера DNS иметь еще и вторичный сервер DNS. Это позволит уменьшить время отклика на запрос, а также повысить отказоустойчивость сети.

Используйте серверы кэширования

Если вы занимаетесь обслуживанием сети, рекомендуется установить кэширующие DNS-серверы если не на каждый компьютер, то в каждой подсети. Быстродействие, которое обеспечивает такой подход, становится заметным уже в сетях средней сложности.

Инструменты

Для тех, кто не хочет подробно изучать настройку DNS с помощью конфигурационных файлов, существуют доступные инструменты, позволяющие вносить изменения, особо не задумываясь. Вообще говоря, мы не сторонники такого подхода. Однако... Произведите поиск и вы наверняка найдете

десяток-другой программ для удаленного администрирования DNS-сервера, в т. ч. и имеющих графическую "дружественную" оболочку. В частности, исходный код на языке HTML для создания инструментария по управлению службой DNS можно найти в Интернете по адресу **webdns.lcs.mit.edu/cgi-bin/webdns**/. Существует также универсальная программа для администрирования множества сервисов через Интернет — webmin (рис. 17.1).

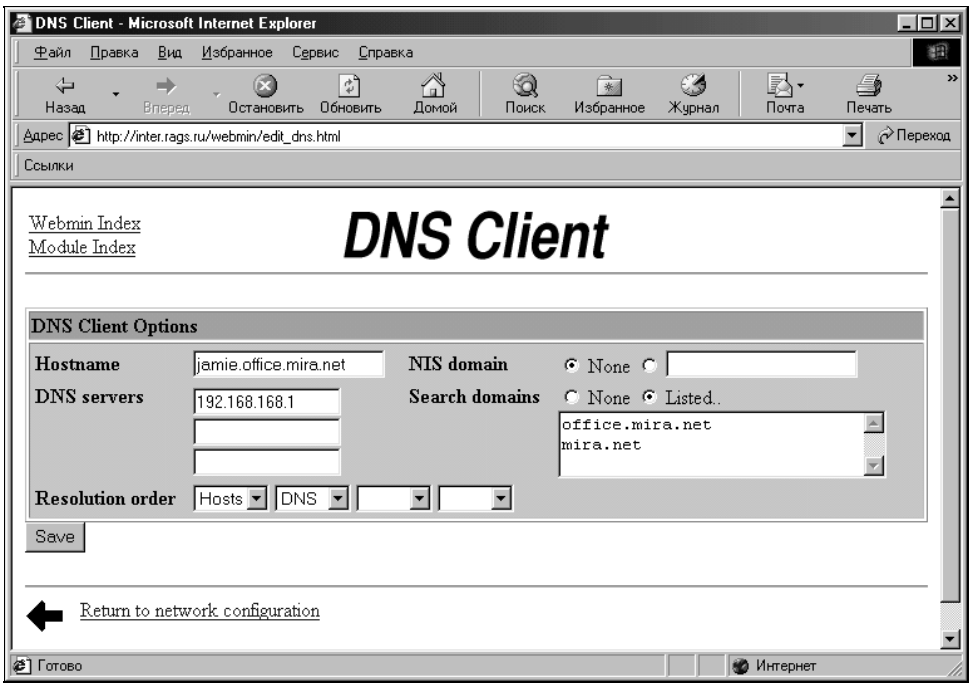

Рис. 17.1. webmin — редактирование записи DNS

Ссылки

- **http://www.biblioteka.agava.ru/nastroyka_dns.htm** — Вячеслав Калошин, "Настройка DNS".
- **http://www.4com.ru/support/DNSAdvanSetup.html** — Эрик Холл, "Тонкая настройка DNS".
- **http://linux.webclub.ru/bind/pers_dns.html** — В. Водолазкий, к. т. н., "Мой личный сервер DNS".
- **http://www.webmin.com/webmin/** — сайт программы webmin.
- DNS-HOWTO *(см. гл. 13)*.

Глава 18

Почта

Эта глава посвящена электронной почте, тому, с чего и начинался Интернет. Функционирование электронной почты очень похоже на свой бумажный прототип. Давайте представим как работает обычная почта. Человек пишет письмо, подписывается, указывает на конверте адрес отправителя и получателя, запечатывает конверт и бросает его в почтовый ящик. Специальная служба, условно назовем ее курьерской, периодически объезжает почтовые ящики, собирает письма и отвозит их на почтамт. Там их сортируют и отправляют на почтамты в города назначения. На этих почтамтах письма сортируются по районам и адресатам, и почтальоны доставляют их по почтовым ящикам адресатов.

Приблизительно по такому же принципу работает и электронная почта. Есть программа — почтовый клиент, в которой происходит подготовка почты к отправке и получение почты человеком. Есть программа — транспортный агент, которая отвечает за доставку электронной почты от компьютера к компьютеру, и есть программы, выполняющие роль почтамтов — они получают почту, сортируют ее по адресатам и раскладывают по почтовым ящикам.

В качестве почтового клиента выступают десятки программ, среди которых — mail, Pine, Kmail, Balsa, stuphead, Mozilla и многие другие. Для транспортировки почты используется MTA, Mail Transport Agent — почтовый транспортный агент. Старейшим и наиболее распространенным транспортным агентом является программа sendmail. Так же получили популярность программы Qmail, postfix, exim.

Основой почтовой службы является система адресов. В Интернете принята система адресов, которая базируется на доменном адресе машины, подключенной к сети. Например, для пользователя ivan машины с адресом ogpu.odessa.ua почтовый адрес будет выглядеть так: **ivan@ogpu.odessa.ua**.

Почтовый адрес состоит из двух частей: идентификатора пользователя, который записывается перед знаком "коммерческого at" — "@", и доменного адреса компьютера, который записывается после знака "@". Существует еще один вариант задания почтового адреса — адрес UUCP, который записыва-

ется в виде: **odessa.ua!ogpu!ivan**. Правда протокол UUCP сейчас почти не используется.

Для работы электронной почты разработан специальный протокол Simple Mail Transfer Protocol (SMTP) — простой почтовый протокол, который является протоколом прикладного уровня и использует транспортный протокол TCP.

Протокол SMTP

Simple Mail Transfer Protocol был разработан для обмена почтовыми сообщениями в сети Интернет. SMTP не зависит от транспортной среды и может использоваться для доставки почты в сетях с протоколами, отличными от TCP/IP.

Взаимодействие в рамках SMTP строится по принципу двусторонней связи, которая устанавливается между отправителем и получателем почтового сообщения. При этом отправитель инициирует соединение и посылает запросы на обслуживание, а получатель на эти запросы отвечает.

Как и множество других протоколов, команды и ответы протокола SMTP передаются в ASCII-кодах и представляют собой небольшой набор английских слов. Приведем небольшой пример отправки почтового сообщения по протоколу SMTP:

```
Отправитель: MAIL FROM: <ivan@ogpu.odessa.ua>
Получатель: 250 Ok
Отправитель: RCPT TO: <vano@mail.ru>
Получатель: 250 Ok
Отправитель: DATA
Получатель: 354 Start mail input; end with <CRLF>.<CRLF>
Текст почтового сообщения
Отправитель:
Получатель: 250
```

Протокол, помимо отправки почты, поддерживает переадресацию, прямую посылку сообщения на терминал, обработку ошибок и некоторые другие возможности.

Протокол POP3

Протокол обмена почтовой информацией POP3 (Post Office Protocol) предназначен для получения почты из почтовых ящиков пользователей на их рабочие места при помощи программ-клиентов. Таким образом, по протоколу SMTP пользователи отправляют корреспонденцию, а по протоколу POP3 — получают корреспонденцию из своих почтовых ящиков на почто-

вом сервере в локальные файлы. Этот протокол так же основан на установлении двусторонней связи, команды и ответы протокола передаются в ASCII-кодах и представляют собой небольшой набор английских слов.

Протокол IMAP

Еще одним протоколом разбора почты является протокол IMAP — почтовый протокол интерактивного доступа (Interactive Mail Access Protocol), который по своим возможностям похож на POP3, но разрабатывался как более надежная альтернатива последнему и обладает более широкими возможностями по управлению процессом обмена сообщениями с сервером.

Главным отличием от POP3 является возможность поиска нужного сообщения и разбор заголовков сообщения непосредственно на почтовом сервере.

Формат почтового сообщения (RFC-822)

Формат почтового сообщения определен в документе RFC-822. Почтовое сообщение состоит из трех частей: конверта, заголовка и тела сообщения. Конверт используется программами доставки почтового сообщения, а заголовок и тело сообщения предназначены для его получателя. Заголовок находится перед телом сообщения, отделен от него пустой строкой и состоит из определенных стандартом полей. Поля состоят из имени поля и содержания поля. Имя поля отделяется от содержания символом ":". Для доставки сообщения можно воспользоваться только частью полей заголовка — такими как `Date`, `From`, `Cc` или `To`, например:

- `Date:  26 Aug 76 1429 EDT`
- `From:  1@mail.ru`
- `To:  Sm2@chat.ru`

Поле `Date` определяет дату отправки сообщения, поле `From` — отправителя, а поля `Cc` и `To` — получателей. Однако, если следовать установленным правилам, необходимо определять все поля заголовка, описанные в стандарте:

- `Date:  27 Aug 76 0932`
- `From:  Motya <1@mail.ru>`
- `Subject:  Re: Ответ на письмо`
- `Sender:  K@Other-host`
- `Reply-To:  Sam.Irving@R.org.ru`
- `To:  Geo <J@chat.ru>`
- `Cc:  Sm3@chat.ru`
- `Comment:  Sam is away on bisiness`

- In-Reply-To: <some.string@DBM.Group>, George`s message
- Message-ID: <4331.629.XYzi-What@Other-Host

Поле Subject определяет тему сообщения, Reply-To — пользователя, которому отвечают, Comment — комментарий, In-Reply-To — показывает, что сообщение относится к типу "В ответ на Ваше сообщение, отвечающее на сообщение, отвечающее ...", Message-ID — уникальный идентификатор письма, используемый почтовыми программами.

Формат сообщения постоянно дополняется и совершенствуется. В частности, в RFC-1327 введены дополнительные поля для совместимости с почтой X.400.

Спецификация MIME (Multipurpose Internet Mail Extension)

Спецификация MIME (Multipurpose Internet Mail Extension), приведенная в стандарте RFC-1341, предназначена для описания тела почтового сообщения Интернета. Необходимость в этом стандарте возникла в силу того, что по стандарту RFC-822 в тело почтового сообщения не могут быть включены некоторые специальные символы и восьмибитные символы.

Стандарт RFC-822 подробно описывает в заголовке почтового сообщения текстовое тело письма и механизм его рассылки, а MIME сориентирован на описание в заголовке письма структуры тела почтового сообщения и возможности составления письма из информационных единиц различных типов.

В стандарте зарезервировано несколько способов представления разнородной информации. Для этого используются специальные поля заголовка почтового сообщения:

- поле версии MIME, которое используется для идентификации сообщения, подготовленного в новом стандарте;
- поле описания типа информации в теле сообщения, которое позволяет обеспечить правильную интерпретацию данных;
- поле типа кодировки информации в теле сообщения, указывающее на тип процедуры декодирования;
- два дополнительных поля, зарезервированных для более детального описания тела сообщения.

Стандарт MIME разработан как расширяемая спецификация, в которой подразумевается, что число типов данных будет расти по мере развития форм представления данных.

Рассмотрим некоторые из полей MIME.

MIME-Version

Поле версии указывается в заголовке почтового сообщения и позволяет определить, что сообщение подготовлено в стандарте MIME. Формат поля:

```
MIME-Version: 1.0
```

Поле версии указывается в общем заголовке почтового сообщения и относится ко всему сообщению целиком.

Content-Type

Поле типа используется для описания типа данных, которые содержатся в теле почтового сообщения. Это поле сообщает программе чтения почты, какие преобразования необходимы для того чтобы сообщение правильно проинтерпретировать. Эта же информация используется и программой рассылки при кодировании/декодировании почты. Стандарт MIME определяет семь типов данных, которые можно передавать в теле письма. Для важнейших из них приведем краткие описания.

- Текст (text). Этот тип указывает на то, что в теле сообщения содержится текст. Основным подтипом типа text является plain — плоский текст. Под этим подразумевается неразмеченный текст. Для определения размеченного текста используют подтип richtext, а для определения гипертекста — подтип html;
- Смешанный тип (multipart). Этот тип определяет смешанный документ, который может состоять из фрагментов данных разного типа. Данный тип имеет ряд подтипов;
- Сообщение (message). Данный тип предназначен для работы с обычными почтовыми сообщениями, которые напрямую не могут быть переданы по почте. Существует несколько подтипов:
  - partial — подтип предназначен для передачи одного большого сообщения по частям для последующей автоматической сборки у получателя;
  - External-Body — подтип позволяет ссылаться на внешние информационные источники;
  - rfc822 — стандартный подтип типа message. Определяет сообщения стандарта RFC-822;
- Графический образ (image);
- Аудиоинформация (audio);
- Видеоинформация (video);
- Приложение (application).

Content-Transfer-Encoding

Тип кодирования сообщения. Поскольку передача сообщений происходит в неоднородной среде, неизбежны перекодирования почтового сообщения. Для того чтобы при получении данные были бы правильно распакованы и используется данное поле.

Программное обеспечение

Как и многое другое, взаимодействие между участниками обмена почтового сообщения основано на технологии клиент-сервер. Можно выделить три независимых этапа:

- взаимодействие по протоколу SMTP между почтовым клиентом и почтовым транспортным агентом;
- взаимодействие между транспортными агентами в процессе доставки почты;
- получение сообщения из почтового ящика пользователя почтовым клиентом по протоколу POP3 или IMAP.

Программа sendmail

Основным средством рассылки почты является программа sendmail, хотя она и является одной из старейших и сложных в конфигурации. Sendmail позволяет организовать почтовую службу локальной сети и обмениваться почтой с другими серверами почтовых служб через специальные шлюзы. Sendmail может быть сконфигурирована для работы с различными почтовыми протоколами. Обычно это протоколы UUCP (UNIX-UNIX-CoPy) и SMTP (Simple Mail Transfer Protocol).

Sendmail может интерпретировать два типа почтовых адресов:

- почтовые адреса SMTP;
- почтовые адреса UUCP.

Sendmail можно настроить для поддержки:

- списка адресов-синонимов;
- списка адресов рассылки пользователя;
- автоматической рассылки почты через шлюзы;
- очередей сообщений для повторной рассылки почты в случае отказов при рассылке;
- работы в качестве SMTP-сервера;

- доступа к адресам машин через сервер доменных имен BIND;
- доступа к внешним серверам имен

и многого другого.

Принцип работы программы sendmail

Sendmail идеологически копирует обычную почтовую службу — почта отправляется с заданной периодичностью, перед этим сообщения собираются в очереди и только затем отсылаются.

Как уже упоминалось ранее, каждое сообщение состоит из трех частей: конверта, заголовка и тела сообщения:

- *конверт* состоит из адреса отправителя, адреса получателя и специфической информации, которая используется программами подготовки, рассылки и получения почты. Конверт остается невидимым для отправителя и получателя почтового сообщения;
- *заголовок* состоит из стандартных текстовых строк, которые содержат адреса, информацию о рассылке и данные. Данные из заголовка могут использоваться для оформления конверта сообщения;
- *тело сообщения* следует после первой пустой строки вслед за заголовком сообщения. Все, что следует после этой строки, называется телом сообщения и передается по почте без изменений.

После постановки почтовых сообщений в очередь начинается ее рассылка. При этом выполняются следующие действия:

- адреса отправителя и получателя преобразуются в формат сети — получателя почты;
- если необходимо, то в заголовок сообщения добавляются отсутствующие данные;
- почта передается одной из программ рассылки почты.

Настройка программы sendmail

Настройка программы sendmail происходит при помощи конфигурационного файла /etc/sendmail.cf. Этот файл состоит из нескольких частей:

- описания компьютера (local information) — в данной секции описываются имя компьютера и т. п;
- описания макроопределений sendmail, отвечающих за работу в локальной сети;
- групп имен, которые используются программой для рассылки почты;
- номера версии файла конфигурации;
- опций команды sendmail — опции определяют режимы работы программы;

- доверенных пользователей;
- описания формата заголовка почтового сообщения — в данной секции определяются поля и их формат, которые отображаются в заголовке;
- правил преобразования адресов;
- описания программ рассылки;
- общего набора правил преобразования адресов;
- машинно-зависимой части общего набора правил преобразования адресов.

Обычно после инсталляции sendmail изменения, которые вносятся в файл конфигурации, касаются только имени хоста, домена и шлюзов. В современных дистрибутивах (таких как Red Hat) иногда не приходится делать даже этого.

Подробно о конфигурировании sendmail здесь рассказано не будет — разобраться в конфигурационном файле, который имеет около 100 Кбайт текста, весьма не просто. Для детального ознакомления с конфигурацией sendmail рекомендуется почитать книгу "UNIX — руководство системного администратора", а также документацию, идущую в комплекте с sendmail.

Для примера приведем небольшую секцию локальной конфигурации программы sendmail:

```
##################
#   local info   #
##################
Cwlocalhost
CP.
# UUCP relay host
DYucbvax.Berkeley.EDU
CPUUCP
#  BITNET relay host
#DBmailhost.Berkeley.EDU
DBrelay.kiae.su
CPBITNET
# "Smart" relay host (may be null)
DSrelay.kiae.su
# who I send unqualified names to (null means deliver locally)
DR
# who gets all local email traffic ($R has precedence for unqualified
names)
DH
# who I masquerade as (null for no masquerading)
```

```
DM
# class L: names that should be delivered locally, even if we have a re-
lay
# class E: names that should be exposed as from this host, even if we
masquerade
#CLroot
CEroot
# operators that cannot be in local usernames (i.e., network indicators)
CO @ % !
# a class with just dot (for identifying canonical names)
C..
# dequoting map Kdequote dequote
```

Тестирование отправки почты sendmail

Для проверки правильности функционирования программы sendmail можно запустить ее с ключом `-v` (режим verbose). При этом режиме процесс обмена между транспортными почтовыми агентами выводится на консоль или записывается в файл. Таким образом можно исключить большую часть ошибок в настройке sendmail.

Тестирование обслуживания по протоколу SMTP

Для проверки сервиса SMTP используют программу telnet, подключаемую к 25-му порту:

```
telnet ivan.petrov 25
```

Если на компьютере установлен SMTP-сервер — в ответ получим строку приглашения протокола SMTP, после чего можно вводить команды SMTP:

```
MAIL FROM: user
250 user... Sender ok
RCPT TO: user
250 user... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
This is a test message!!!
...
250 JAA24856 Message accepted for delivery
quit
221 ivan.petrov closing connection
Connection closed by foreign host.
You have new mail.
#
```

В приведенном примере мы отправили сами себе сообщение. Команда MAIL FROM: указывает адрес отправителя почтового сообщения. Затем вводится команда RCPT TO: для указания адреса получателя почтового сообщения. Команда DATA разрешает ввод почтового сообщения. Конец режима редактирования обозначается символом "." в первой позиции строки. Более подробную информацию следует искать в документации по sendmail, а также в табл. 18.1, где приведены команды протокола SMTP, и в табл. 18.2, содержащей коды возврата протокола SMTP.

Команды и коды возврата протокола SMTP

Для тестирования работы SMTP-сервера необходимо знать команды протокола SMTP (табл. 18.1) и его коды возврата (табл. 18.2) и воспользоваться программой telnet.

Таблица 18.1. *Команды протокола SMTP*

Команда	Описание
HELO <SP> <domain> <CRLF>	Открыть сессию взаимодействия по протоколу SMTP. <domain> — доменное имя машины
MAIL <SP> FROM:<reverse-path> <CRLF>	Сообщить адрес отправителя <reverse-path>. Обязательная команда, которую надо выдать перед отправкой сообщения
RCPT <SP> TO:<forward-path> <CRLF>	Сообщить адрес получателя <forward-path>. Обязательная команда, которую выдают после MAIL FROM, но перед DATA
DATA <CRLF>	Начать передачу тела почтового сообщения. Тело сообщения должно завершаться точкой "." в первой позиции строки
RSET <CRLF>	Конец операции
SEND <SP> FROM:<reverse-path> <CRLF>	Послать сообщение на терминал пользователя, который определяется командой RCPT
SOML <SP> FROM:<reverse-path> <CRLF>	SEND OR MAIL. Послать в почтовый ящик или на терминал пользователя
SAML <SP> FROM:<reverse-path> <CRLF>	SEND AND MAIL. Послать в почтовый ящик и на терминал пользователя
VRFY <SP> <string> <CRLF>	Получить информацию о пользователе, имя которого указывается в качестве аргумента команды <string>

Таблица 18.1 *(окончание)*

Команда	Описание
EXPN <SP> <string> <CRLF>	Получить информацию о пользователях, зарегистрированных в качестве получателей корреспонденции
HELP [<SP> <string>] <CRLF>	Краткая справка по командам протокола
NOOP <CRLF>	Нет операции
QUIT <CRLF>	Завершить сессию
TURN <CRLF>	Поменять местами сервер и клиент

Таблица 18.2. *Коды возврата протокола SMTP*

Код возврата	Текстовое пояснение сервера	Описание
211	System status, or system help reply	Статус системы или помощь
214	Help message. [Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user]	Краткая справка
220	<domain> Service ready	SMTP-сервис готов к работе
221	<domain> Service closing transmission channel	Сервис закрыл канал передачи данных
250	Requested mail action okay, completed	Соединение установлено
251	User not local; will forward to <forward-path>	Пользователь не местный. Выполнить перенаправление запроса
354	Start mail input; end with <CRLF>.<CRLF>	Начать ввод почтового сообщения
421	<domain> Service not available, closing transmission channel [This may be a reply to any command if the service knows it must shut down]	Сервис отсутствует. Канал передачи данных закрыт
450	Requested mail action not taken: mailbox unavailable [E.g., mailbox busy]	Нет возможности записать данные в почтовый ящик
451	Requested action aborted: local error in processing	Ошибка при обработке запроса

Таблица 18.2 *(окончание)*

Код возврата	Текстовое пояснение сервера	Описание
452	`Requested action not taken: insufficient system storage`	Запрос не выполнен — недостаточно памяти
500	`Syntax error, command unrecognized [This may include errors such as command line too long]`	Синтаксическая ошибка — нет такой команды
501	`Syntax error in parameters or arguments`	Синтаксическая ошибка в аргументах команды
502	`Command not implemented`	Данная команда не может быть выполнена
503	`Bad sequence of commands`	Неправильная последовательность команд
504	`Command parameter not implemented`	Параметр команды не может быть использован в данном контексте
550	`Requested action not taken: mailbox unavailable [E.g., mailbox not found, no access]`	Не найден соответствующий почтовый ящик
551	`User not local; please try <forward-path>`	Пользователь не найден; можно попробовать отправить почту по другому адресу
552	`Requested mail action aborted: exceeded storage allocation`	Превышены квоты на использование ресурсов памяти
553	`Requested action not taken: mailbox name not allowed [E.g., mailbox syntax incorrect]`	Имя почтового ящика неправильное
554	`Transaction failed`	Аварийное завершение

Тестирование обслуживания по протоколу POP3

Аналогично тестированию обслуживания по протоколу SMTP с помощью программы telnet можно проверить функционирование и POP3-протокола. Для этого необходимо подключиться к нашему серверу по порту 110.

```
telnet ivan.petrov 110
  user user
```

```
+OK Password required for user.
pass 12345623432
+OK user has 3 messages (33276 octets).
list
+OK 3 messages (33276 octets)
1 11276
2 11000
3 11000
.
dele 3
+OK Message 3 has been deleted.
quit
+OK
Connection closed by foreign host.
```

Очень похоже на протокол SMTP. Подключились к порту 110. Производим "опознание" пользователя с помощью команд `user` и `pass`. Затем командой `list` узнаем количество сообщений в почтовом ящике и их размер. Командой `dele` отмечаем сообщение к удалению, которое произойдет по окончании сеанса. Команда `Quit` завершает сеанс работы с сервером. Все просто.

Команды протокола POP3

Для тестирования работы POP3-сервера необходимо знать его команды (табл. 18.3) и воспользоваться программой telnet.

Успешное выполнение команды заканчивается выводом сообщения "`+OK`", а неуспешное "`-ERR`" соответственно.

Таблица 18.3. *Команды протокола POP3*

Команда	Назначение	Возможные возвращаемые значения (кроме `+OK` или `-ERR`)
`USER <имя пользователя>`	Посылка имени пользователя серверу	
`PASS <пароль>`	Посылка пароля серверу	
`QUIT`	Окончание сеанса работы	
`STAT`	Получить состояние почтового ящика	`+OK <кол-во сообщений> <общий размер всех сообщений>`

Таблица 18.3 *(продолжение)*

Команда	Назначение	Возможные возвращаемые значения (кроме +OK или -ERR)
UST [<номер сообщения>]	Получить параметры всех сообщений в ящике пользователя. Если задан номер сообщения, то будут получены только его параметры	+OK <параметры сообщений> Возвращаемые параметры сообщений зависят от того, был ли задан номер сообщения. Если да, то сразу после +OK следует сообщение сервера. Затем строка за строкой передаются параметры всех сообщений в формате <номер сообщения> <размер сообщения>
RETR <номер сообщения>	Получить сообщение с сервера	+OK <тест запрошенного сообщения> — если команда прошла удачно -ERR <комментарий сервера> — если запрошенное сообщение отсутствует на сервере
DELE <номер сообщения>	Пометить сообщение на сервере как удаленное. Реально оно будет удалено после команды QUIT	+OK <комментарий сервера> — если сообщение было помечено на удаление -ERR <комментарий сервера> — если сообщение не существует или уже отмечено как удаленное
NOOP	Пустая операция	+OK
RSET	Отменить удаление сообщений, помеченных как удаленные	
TOP <номер сообщения> <кол-во строк>	Считать заголовок сообщения и первые строки в количестве, заданном параметром <кол-во строк>	+OK Далее строка за строкой передается заголовок сообщения. За ним следует пустая строка и, если имеется второй параметр, передаются начальные строки сообщения

Таблица 18.3 *(окончание)*

Команда	Назначение	Возможные возвращаемые значения (кроме +OK или -ERR)
UIDL [<номер сообщения>]	Получить уникальные идентификаторы всех сообщений в ящике пользователя. Если задан номер сообщения, то будет получен только его идентификатор	+OK <параметры сообщений> Возвращаемые параметры сообщений зависят от того, был ли задан номер сообщения. Если да, то сразу после +OK идут номер запрошенного сообщения и его идентификатор. Если команда вызвана без параметра, то после статуса +OK следует сообщение сервера. Затем строка за строкой передаются параметры всех сообщений в формате <номер сообщения> <идентификатор>
APOP <имя пользователя> <дайджест>	Осуществляет подключение к почтовому серверу по закодированной алгоритмом MD5 строке, защищая транзакцию от разглашения пароля пользователя	+OK <комментарий сервера> — если имя пользователя или дайджест соответствуют имеющемуся почтовому ящику пользователя

Рассмотрим теперь более современные интерфейсы подготовки почтовых сообщений bml и elm. Обе эти программы подготовки почты работают в режиме полноэкранных интерфейсов.

Почтовые клиенты

Сегодня существует несколько десятков почтовых клиентов — простейшие текстовые, сложные текстовые, графические и даже Web-клиенты. Трудно охватить все разнообразие, поэтому приведем примеры нескольких почтовых клиентов. Для настройки всех почтовых клиентов необходимо знать ряд параметров:

- ❒ логин пользователя;
- ❒ пароль пользователя;
- ❒ адрес SMTP-сервера;
- ❒ порт SMTP-сервера;
- ❒ адрес POP3-сервера;
- ❒ порт POP3-сервера

или те же параметры IMAP-сервера.

Зная эти параметры и при правильных настройках сети на вашем компьютере не составляет труда настроить практически любого почтового клиента.

mail

Одна из первых программ — почтовых клиентов. Не блещет красотой интерфейса (его просто нет), достаточно примитивна, зато не занимает много места и поэтому присутствует практически на любом хосте. Представление о программе mail можно получить из рис. 18.1.

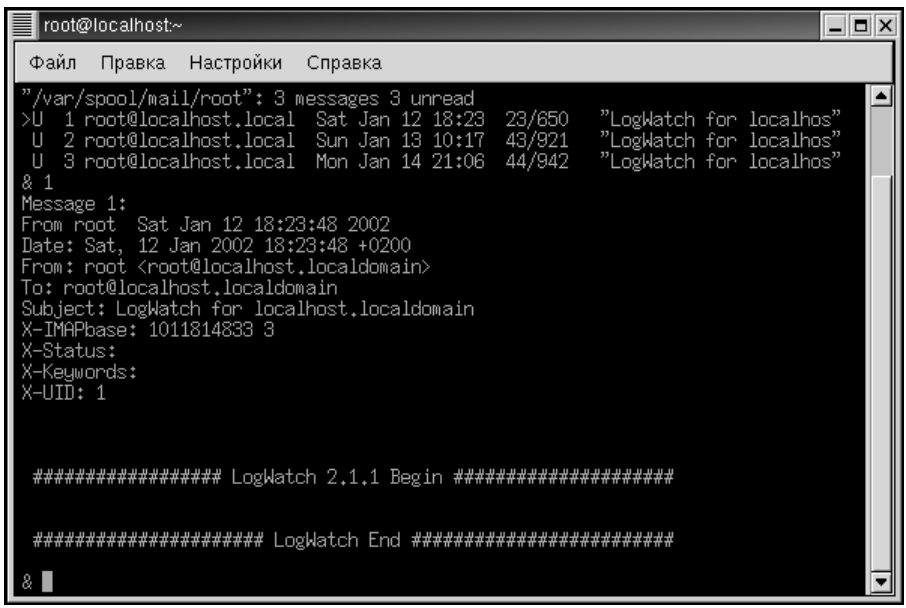

Рис. 18.1. Почтовый клиент mail

Pine

Один из самых "навороченных" текстовых почтовых клиентов, который так же позволяет работать с сообщениями новостей (news). Удобный, приятный в использовании интерфейс. Внешний вид программы pine изображен на рис. 18.2.

Mozilla

Аналог Netscape Communicator. Достаточно устойчивая и надежная почтовая программа. Является частью программного комплекса Mozilla — Web-браузер, почтовый клиент, клиент чата. Почтовый клиент Mozilla представлен на рис. 18.3.

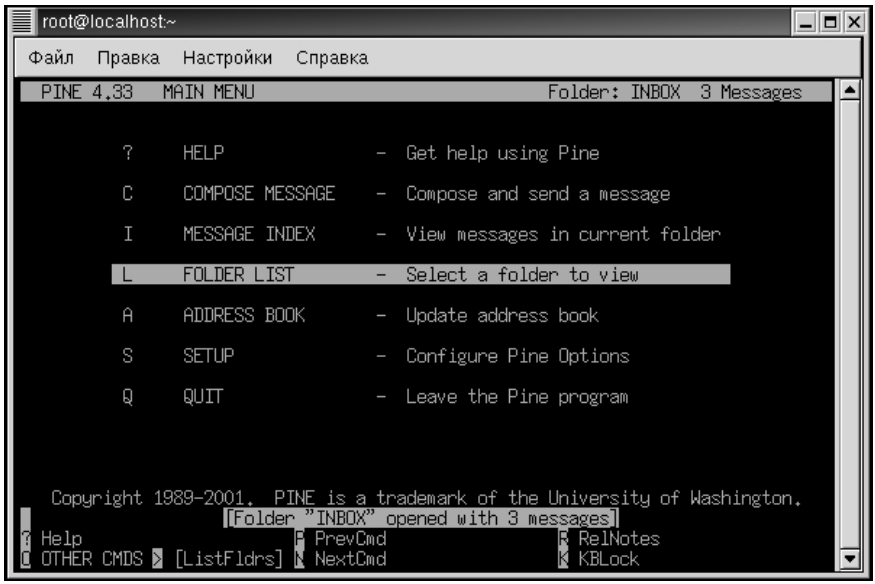

Рис. 18.2. Почтовый клиент pine

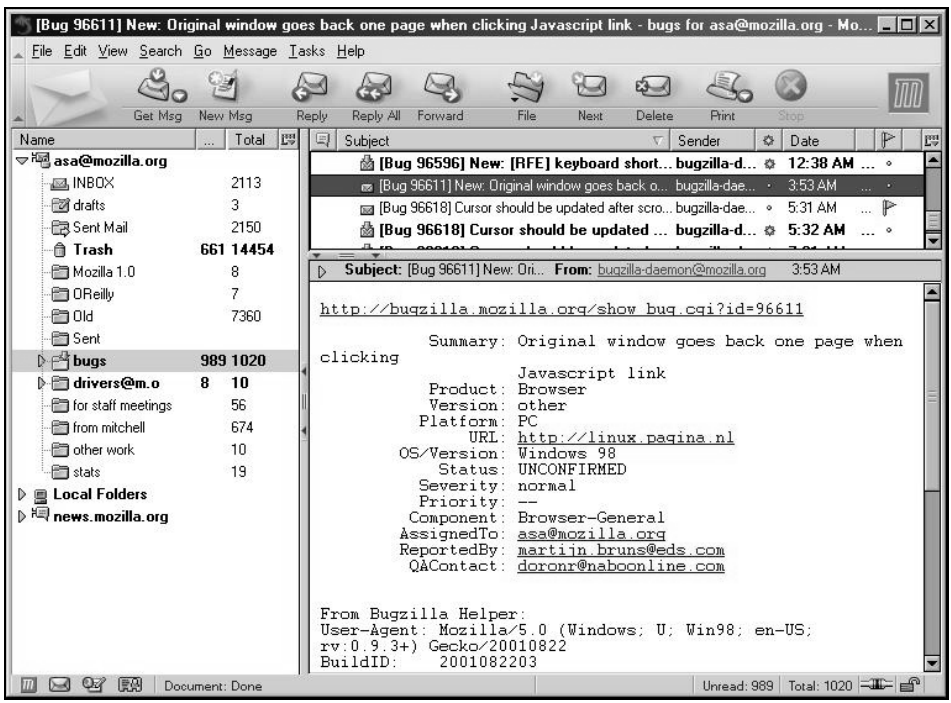

Рис. 18.3. Почтовый клиент Mozilla

Balsa

Стандартный почтовый клиент GNOME. Понятный и удобный интерфейс (рис. 18.4). Могут быть некоторые проблемы с русификацией.

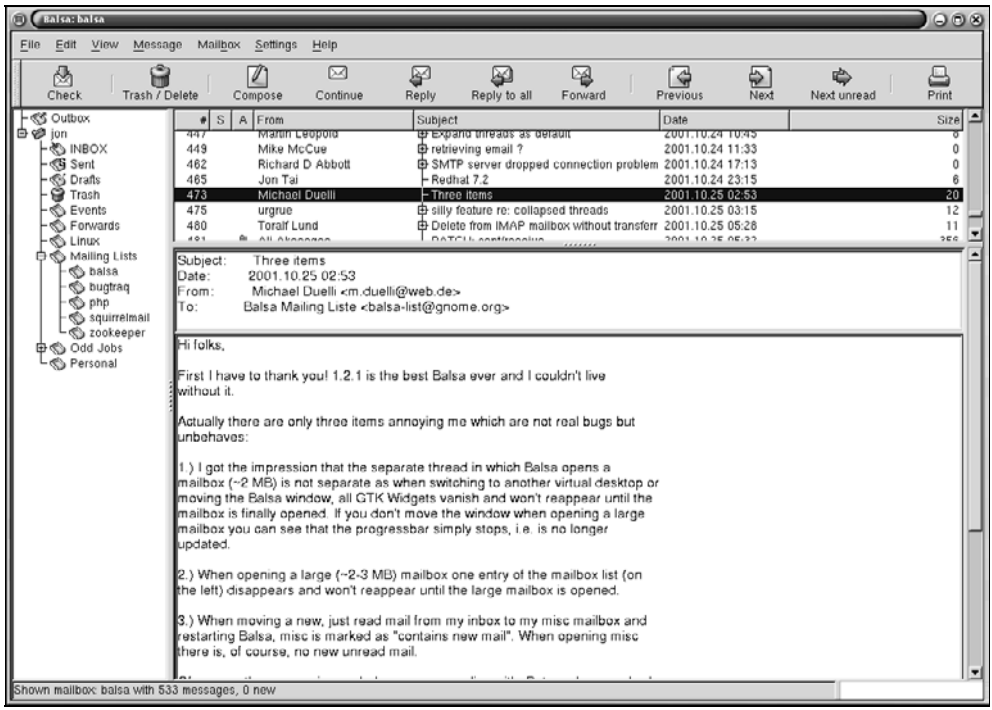

Рис. 18.4. Почтовый клиент Balsa

Stuphead

Модификация почтового клиента, написанного японским программистом, сделанная компанией ALT Linux. Пока получается достаточно неплохо и с русским языком проблем нет (рис. 18.5).

Evolution

Попытка программистов создать нечто подобное Microsoft Outlook — почтовый клиент (рис. 18.6), органайзер (рис. 18.7), дневник и записную книжку в одном комплекте. Получился довольно "увесистый" программный пакет.

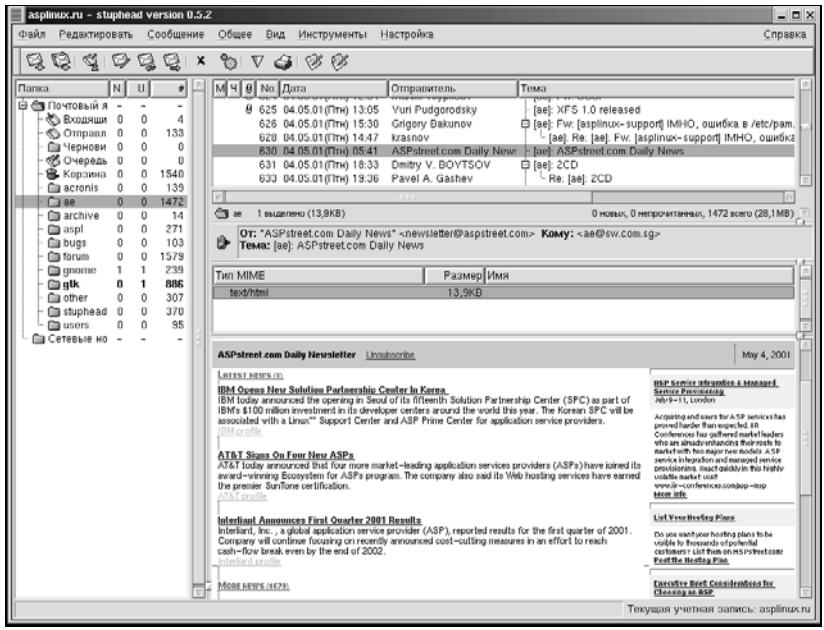

Рис. 18.5. Почтовый клиент Stuphead

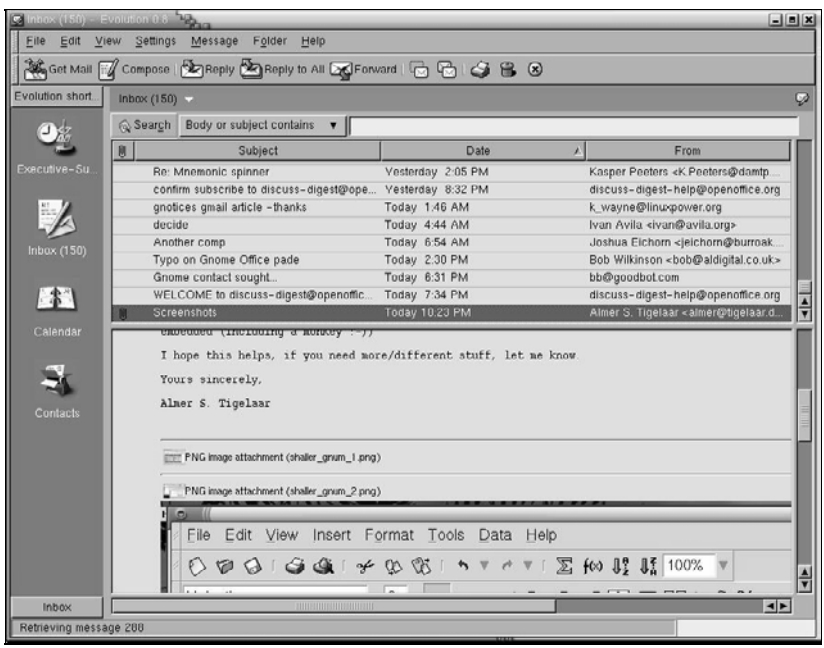

Рис. 18.6. Почтовый клиент Evolution

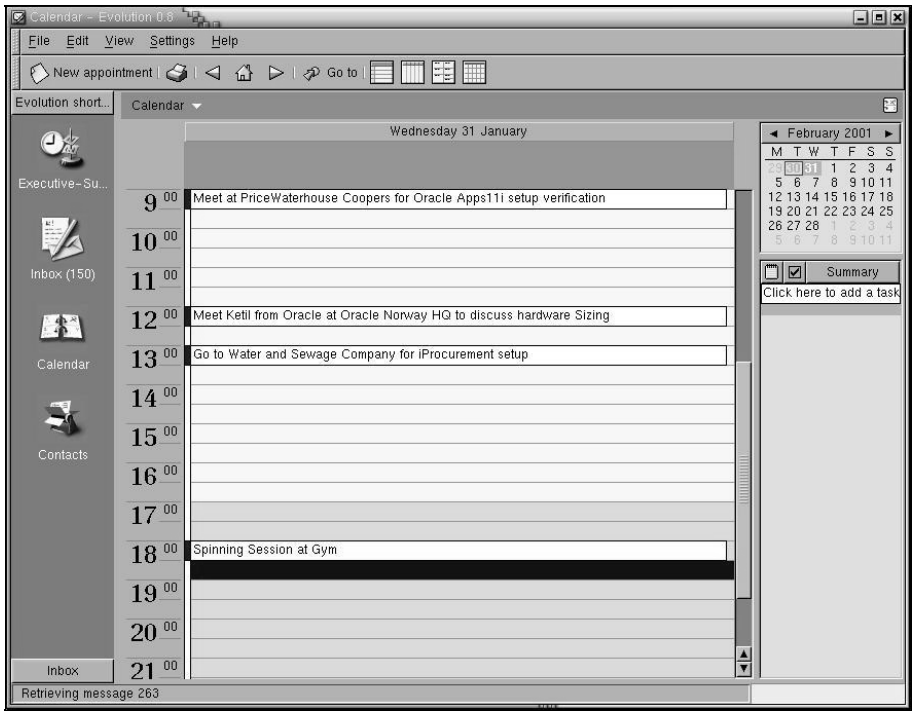

Рис. 18.7. Почтовый клиент Evolution — планировщик встреч

Kmail

Очень хороший почтовый клиент (рис. 18.8). Хорошо понимает различные кодировки, удобный и понятный интерфейс. Является стандартным почтовым клиентом для KDE.

Ссылки

- **www.citforum.ru/internet/servers/** — Павел Храмцов. Организация и администрирование почтовых и файловых серверов Internet. Центр Информационных Технологий.
- Соответствующие HOWTO *(см. гл. 13)*:
  - Linux Mail-Queue mini-HOWTO;
  - Sendmail+UUCP HOWTO;
  - Sendmail address rewriting mini-HOWTO.

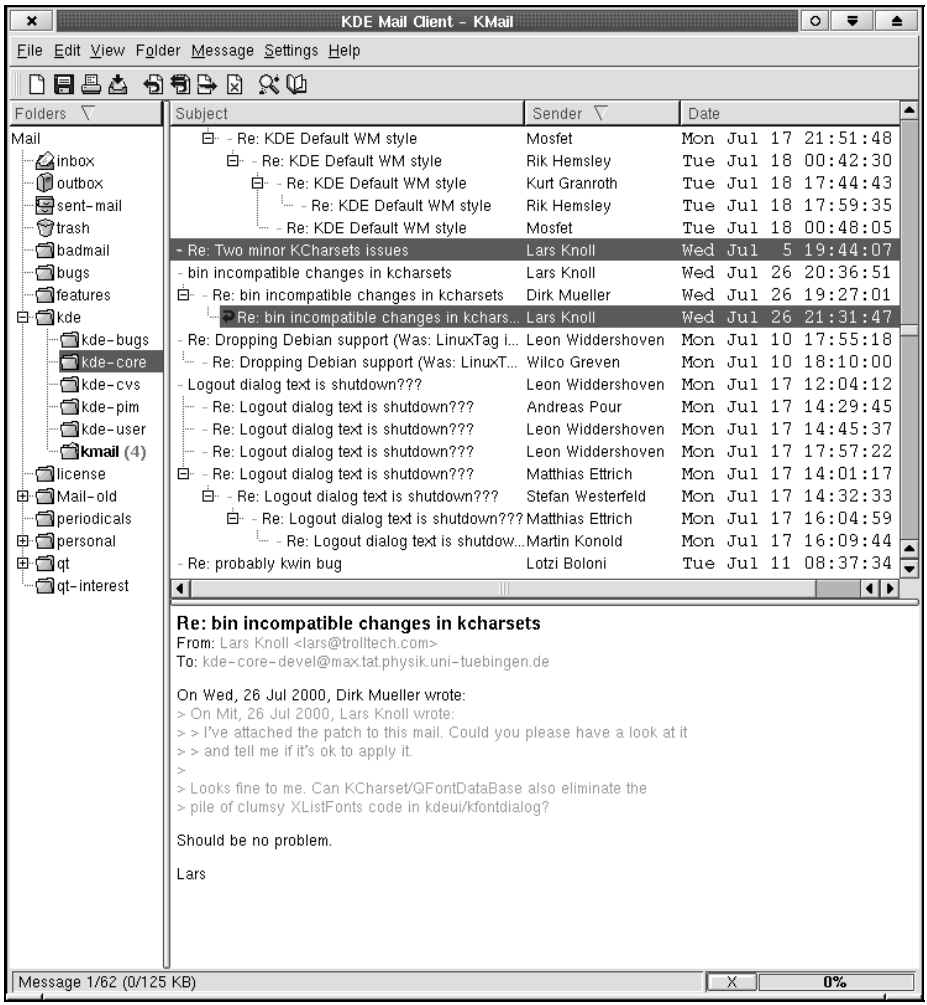

Рис. 18.8. Почтовый клиент Kmail

Глава 19

Web-сервер Apache

Для создания Web-сервера (HTTP-сервера) в мире Linux в основном используется бесплатный (лицензия GNU) Web-сервер Apache. По статистическим данным до недавнего времени более пятидесяти процентов Web-серверов в Сети были построены на базе сервера Apache.

Чем же привлекателен этот сервер? Во-первых, большое количество возможностей — использование CGI-скриптов, шифрования, доступ по паролю, перекодирование страниц "на лету", поддержка виртуальных хостов и многое другое. Во-вторых, малая требовательность к ресурсам и большая производительность. В-третьих, многоплатформенность — Apache есть для Linux, для различных клонов UNIX, для Windows. В-четвертых, он бесплатный и с открытым исходным кодом. Список можно продолжать. Конечно, есть и недостатки, к примеру, некоторые сложности с конфигурированием. Но в целом — этот сервер не зря получил столь большую популярность.

В качестве альтернативы для Linux-платформы в последнее время выдвинулся Web-сервер TUX, который использует особенности ядра Linux, что позволило резко увеличить количество обрабатываемых запросов за единицу времени. Однако у этого сервера есть несколько минусов, в том числе:

- платформозависимость;
- неустоявшийся код;
- мало дополнительных возможностей по сравнению с Apache.

Далее мы рассмотрим конфигурирование сервера. Существует так называемый "Русский Apache" — адаптированный для русскоязычной аудитории. Мы, конечно, коснемся этого вопроса, но в целом будем рассматривать стандартный Apache.

Конфигурация

Установка сервера для дистрибутивов, использующих rpm-пакеты, не составляет особого труда — необходимо скачать нужный пакет и произвести установку сервера командой

```
rpm -I имя_пакета
```

С конфигурированием труднее. Мы рассмотрим только наиболее распространенные директивы и их параметры, поскольку полный перечень слишком большой. Если вы производите переконфигурирование при рабочем сервере Apache, вам необходимо заставить сервер перечитать конфигурационные файлы. Сервер перечитывает конфигурационные файлы при запуске или при получении сигнала `-HUP` или `-uSR1`. Если сервер находится в рабочем состоянии, то при изменении конфигурации его рекомендуется перезапустить командой `kill -USR1`, поскольку в этом случае текущие соединения не закрываются принудительно и завершаются обычным образом, а следующие клиенты работают уже с новыми конфигурационными файлами.

Конфигурация сервера задается в файлах httpd.conf, srm.conf, access.conf и .htaccess. Файл httpd.conf предназначен для общей конфигурации сервера, srm.conf содержит описание доступных ресурсов, а access.conf — права доступа к ресурсам. Однако в современных версиях сервера любая директива конфигурации может лежать в любом из этих файлов. Сейчас de-facto все директивы конфигурации содержатся в файле httpd.conf.

Файлы .htaccess могут находиться в каталогах и переопределять права доступа и другие параметры данных каталогов. Некоторые модули могут иметь свои отдельные файлы конфигурации (например, mod_charset требует файлы, хранящие таблицы перекодировки).

Используемые обозначения

Ниже показаны обозначения, используемые при описании параметров конфигурации сервера:

- `S` — директива действует на поведение сервера целиком;
- `V` — действует, если запрос касается данного виртуального хоста;
- `D` — определяет свойства только данного каталога;
- `A` — определяет свойства только данного каталога.

Права доступа и свойства объекта

Права доступа к данному каталогу и его свойства определяются следующими директивами.

- `DA allow from {host}`

  Определяет, с каких хостов разрешен доступ к данному каталогу:

  - `all` — для всех;
  - доменное имя — с тех хостов, имя которых заканчивается этой строкой;
  - полный IP-адрес;

- частичный IP-адрес — 1, 2 или 3 байта IP-адреса;
- a.b.c.d/e.f.g.h — сеть/сетевая маска;
- a.b.c.d/nnn — сеть/подсеть.

❒ `DA allow from env=имя_переменной`

Доступ разрешается, только если определена соответствующая переменная окружения.

❒ `D AllowOverride {None | All | AuthConfig | FileInfo | Indexes | Limit | Options}`

Определяет, какие директивы из .htaccess в данном каталоге могут перекрывать конфигурацию сервера.

❒ `D AuthName домен_авторизации`

Определяет, какой домен авторизации клиент должен использовать при определении имени и пароля.

❒ `DA deny from {host}`

Определяет, с каких адресов запрещен доступ к данному каталогу:

- `all` — для всех;
- доменное имя — с тех хостов, имя которых заканчивается этой строкой;
- полный IP-адрес;
- частичный IP-адрес — 1, 2 или 3 байта IP-адреса;
- a.b.c.d/e.f.g.h — сеть/сетевая маска;
- a.b.c.d/nnn — сеть/подсеть.

❒ `DA deny from env=имя_переменной`

Доступ не разрешается если определена соответствующая переменна окружения.

❒ `SV <Directory имя_каталога> ... </Directory>`

Внутри этой пары тэгов определяются права и свойства данного каталога. В качестве имен используется полный путь к каталогу.

❒ `SV <DirectoryMatch регулярное_выражение> ... </DirectoryMatch>`

Внутри пары тэгов определяются права и свойства данного каталога. В качестве имени используется регулярное выражение.

❒ `SV DocumentRoot путь`

Определяет, где находится корневой каталог документов сервера или виртуального сервера.

❒ `SVDA ErrorDocument error-code document`

Определяет, какой документ выдавать в случае ошибки с указанным кодом.

- `SVA <Files имя_файла> ... </Files>`

  Внутри пары тэгов определяются права и свойства файлов. Может находиться внутри секции Directory или .htaccess.

- `SVA <FilesMatch имя_файла> ... </FilesMatch>`

  Внутри пары тэгов определяются права и свойства файлов, в качестве имен используется регулярное выражение. Может находиться внутри секции Directory или .htaccess.

- `SVDA <Limit {метод}> ... </Limit>`

  Эта пара тэгов для группы директив, управляющих доступом. Методы — `GET`, `POST`, `PUT`, `DELETE`, `CONNECT` или `OPTIONS`.

- `SV <Location URL> ... </Location>`

  Пара тэгов для определения свойств и прав доступа для данного URL.

- `SV <LocationMatch URL> ... </LocationMatch>`

  Пара тэгов для определения свойств и прав доступа для данного URL (регулярное выражение).

- `SVDA Options [+|-]option ...`

  Определяет возможности сервера в данном каталоге:

  - `ALL` — все кроме MultiView;
  - `ExecCGI` — разрешается выполнение CGI;
  - `FollowSymLinks` — разрешено ходить по символьным ссылкам;
  - `Includes` — использовать SSI (Server Side Include);
  - `IncludesNOEXEC` — использовать SSI, кроме exec и include CGI;
  - `Indexes` — генерировать список содержимого каталога, если отсутствует файл index.html;
  - `MultiViews` — определять представление ресурса в зависимости от предпочтений клиента;
  - `SymLinksIfOwnerMatch` — следовать по символьным ссылкам, только если владелец целевого файла совпадает с владельцем ссылки.

- `DA order option`

  Определяет очередность, в которой применяются директивы `allow` и `deny`:

  - `deny, allow` — первой применяются директивы `deny`, затем `allow` (начальное состояние — доступ разрешен);
  - `allow, deny` — первой применяются директивы `allow`, затем `deny` (начальное состояние — запрещено);

- `mutual-failure` — доступ только с тех хостов, которые перечислены в `allow` и не перечислены в `deny`.

❐ `DA require entity-name entity entity...`

Какой пользователь может иметь доступ к каталогу:

- `user {userid}` — только пользователи с данными именами;
- `group {group-name}` — только пользователи из данной группы;
- `valid-user` — любой аутентифицированный пользователь.

❐ `DA satisfy [all|any]`

Если для ограничения доступа используется логин/пароль и IP-адрес, то сервер будет требовать соответствия обоих критериев (`all`) или любого из них (`any`). По умолчанию — `all`.

Общие характеристики сервера

Общие характеристики сервера определяются следующими директивами.

❐ `SV ErrorLog filename | syslog:facility`

Определяет, куда выводить сообщения об ошибках.

❐ `SV Group группа`

Определяет, с правами какой группы будет обрабатываться запрос.

❐ `SVD HostNameLookups on | off | double`

Указывает, определять ли имя клиента по его IP-адресу.

❐ `SVDA <IfDefine [!]parameter-name> ... </IfDefine>`

Условная конфигурация, в зависимости, определен параметр или нет.

❐ `SVDA <IfModule [!]module-name> ... </IfModulee>`

Условная конфигурация, в зависимости, включен модуль в состав сервера или нет.

❐ `S Include имя_файла`

Вставить содержимое файла в состав конфигурационного файла в данном месте.

❐ `S KeepAlive on | off`

Обслуживать несколько запросов, не прерывая TCP-соединения с клиентом.

❐ `SV LogLevel emerg | alert | crit | error | warn | notice | info | debug`

Определяет, что писать в журнал ошибок.

❐ `S MaxClients число`

Определяет максимальное количество одновременно обслуживаемых клиентов.

- `S MaxKeepAliveRequests число`

  Определяет, максимальное количество запросов.

- `S MaxRequestsPerChild число`

  Определяет максимальное количество одновременно обслуживаемых запросов одним процессом.

- `S MaxSpareServers число`

  Определяет максимальное число процессов, не осуществляющих в данный момент соединения.

- `S MinSpareServers число`

  Определяет минимальное число процессов, не осуществляющих в данный момент соединения.

- `S Port номер_порта`

  Определяет, по какому порту производится соединение (по умолчанию — 80-й порт).

- `SV RLimitCPU soft-limit  max-resource-limit`

  Задает максимальное число секунд CPU для любого процесса. Оба параметра могут быть числом или словом `max`.

- `SV RlimitMEM soft-limit  max-resource-limit`

  Задает максимальное число байтов, которое может использовать каждый процесс. Оба параметра могут быть числом или словом `max`.

- `SV RlimitNPROC soft-limit  max-resource-limit`

  Задает максимальное число процессов, которое может запустить каждый пользователь. Оба параметра могут быть числом или словом `max`.

- `SV ServerAdmin email-address`

  Электронный адрес администратора Web-сервера.

- `SV ServerName имя`

  Полное доменное имя, используется для перенаправления.

- `S ServerRoot полное-имя-каталога`

  Указывает место, где лежат все файлы сервера по умолчанию.

- `SVDA ServerSignature Off | On | Email`

  Определяет, какую информацию включать в конце документов, генерируемых сервером:

  - `Off` — отсутствие информации;
  - `On` — имя сервера и версия;
  - `EMail` — имя сервера, версия и почтовый адрес администратора.

- `S ServerTokens Minimal|OS|Full`

  Определяет, что сервер сообщает о себе в заголовке `Server`;

- `S ServerType standalone | initd`

  Определяет тип сервера — постоянно находящийся в оперативной памяти или вызываемый демоном `initd`.

- `S StartServers number`

  Определяет, сколько дочерних процессов запускать при начальном старте сервера.

- `S TimeOut секунд`

  Количество секунд, определяющее тайм-аут.

- `SVDA UseCanonicalName on|off`

  Используется при генерации URL, ссылающихся на этот же сервер:

  - `On` — использовать имя, определенное в `ServerName` и `Port`;
  - `Off` — использовать параметры из запроса пользователя.

- `SV User uid`

  Определяет, с правами какого пользователя будет работать сервер.

Виртуальные серверы

Общие характеристики виртуальных серверов определяются следующими директивами.

- `S NameVirtualHost addr[:port]`

  Задает пару соответствия виртуальный хост — адрес/порт.

- `V ServerAlias host1 host2 ...`

  Задает альтернативные имена для виртуального хоста.

- `V ServerPath путь`

  Все запросы, которые начинаются с `путь`, будут обслуживаться этим виртуальным сервером.

- `S <VirtualHost {адрес[:порт]}> ... </VirtualHost>`

  Пара тэгов определяет описание виртуального сервера. Адрес и порт определяют адрес, по которому он будет отзываться. Внутри используются любые директивы с признаком `V`.

Преобразование адресов

Преобразование адресов определяется следующими директивами.

- `SV Alias URL dirname-filename`

Запрос, начинающийся с `URL`, будет отображен на файл, начинающийся с `dirname-filename`.

- `SV AliasMatch регулярное_выражение dirname-filename`

  Аналогична директиве `Alias`, но сравнение производится в соответствии с регулярным выражением.

- `SV ScriptAlias url-path directory-filename`

  Аналогична директиве `Alias`, но дополнительно пометить каталог как содержащий CGI.

- `SV ScriptAliasMatch regex directory-filename`

  Аналогична директиве `AliasMatch`, но дополнительно пометить каталог как содержащий CGI.

Преобразование HTTP-заголовков

Преобразование HTTP-заголовков определяется следующими директивами.

- `SVDA MetaFiles on/off`

  Включить/выключить преобразование для данного каталога.

- `SVDA MetaDir directory-name`

  Определяет имя каталога, в котором лежат метафайлы.

- `SVDA MetaSuffix suffix`

  Определяет суффикс, который добавляется к имени файла, чтобы найти метафайл для него.

- `SVDA ExpiresActive on|off`

  Определяет, посылать ли заголовок `Expire` (срок хранения документа в кэше).

- `SVDA Header unset header`

  Предписывает удалить заголовок.

Безопасность

Безопасность сервера определяется следующими директивами.

- `DA AuthGroupFile filename`

  Определяет имя файла, в котором хранится список групп пользователей.

- `DA AuthUserFile filename`

  Определяет имя файла, в котором хранится список пользователей.

- `D AuthType [Basic | Digest]`

  Определяет тип аутентификации.

- `DA AuthAuthoritative on | off`

  Если установлено `off`, то в процессе авторизации, если отсутствует имя пользователя в текущей базе данных, происходит обращение к модулю аутентификации нижнего уровня.

- `DA AuthDBMGroupFile filename`

  Аналогична `AuthGroupFile`, но использует `dbm`.

- `DA AuthDBMUserFile filename`

  Аналогична `AuthUserFile`, но использует `dbm`.

Индекс каталога

Индекс каталога определяется следующими директивами.

- `SVDA AddAlt string file file...`

  Определяет, какой текст показывать вместо иконки, если на стороне клиента отключена загрузка картинок.

- `SVDA AddDescription string file file...`

  Определяет текстовое описание файла.

- `SVDA AddIcon icon name name ...`

  Определяет, какую картинку показать для файла, соответствующего `name`.

- `SVDA DefaultIcon url`

  Определяет, какая картинка будет использоваться, если нет соответствующей.

- `SVDA DirectoryIndex local-url local-url ...`

  Задает имя файла (относительно запрашиваемого каталога), в котором находится индексный файл каталога.

- `SVDA HeaderName filename`

  Определяет, что в качестве заголовка индекса будет вставлен указанный файл.

- `SVDA IndexIgnore file file ...`

  Определяет список файлов, которые надо скрывать.

- `SVDA IndexOptions [+|-]option [+|-]option ...`

  Определяет параметры сортировки и оформления:

  - `FancyIndexing` — сортировка по столбцам;
  - `IconHeight=pixels` — высота иконки;
  - `IconWidth=pixels` — ширина иконки;
  - `NameWidth=[n | *]` — ширина колонки.

- `SVDA ReadmeName filename`

  В конец индекса будет вставлен указанный файл (сначала ищется файл filename.html, затем просто filename).

Перекодировка (русификация)

Для перекодирования документов из одной кодовой страницы в другую используются нижеприведенные директивы.

Определение кодировки и таблиц перекодировки:

- `SV CharsetDecl имя_кодировки [ S ]`

  Флаг `S` подавляет выдачу `charset=...` клиенту.

- `SV CharsetRecodeTable из_какой в_какую имя_файла_с_таблицей [ имя_файла_с_обратной_таблицей ]`

  Задает, из какой кодировки в какую производится перекодирование.

- `SV CharsetWideRecodeTable из_какой в_какую имя_файла_с_таблицей`

  Используется для перекодировок из символа в строку, например, для транслитерации.

- `SVDLA CharsetAlias официальное_имя синоним ...`

  Определяет синонимы для имени кодировки.

Определение кодировки хранения:

- `SVDLA CharsetSourceEnc имя_кодировки`

  Определяет, в какой кодировке хранятся документы.

- `SVDLA CharsetByExtension имя_кодировки .ext1 ...`

  Разрешает определение кодировки по расширению.

- `SVDLA CharsetProcessType mime-type`

  Определяет, какие типы файлов надо обрабатывать; всегда обрабатываются — text/*.

Определение кодировки клиента:

- `SVDLA CharsetPriority имя_кодировки1 ...`

  Определение приоритета, если клиент задает несколько `Accept`.

- `SVDLA CharsetBrokenAccept Agent-Substring accept_charset_string`

  Игнорировать данный заголовок `Accept` от данного клиента — использовать другие механизмы для определения типа клиентской кодировки.

- `SVDLA CharsetSelectionOrder Rule1 ...`

  Устанавливает приоритет способов определения кодировки клиента:

  - `Portnumber` — по номеру порта;

- `Hostname` — если каноническое имя хоста начинается с имени кодировки или его синонима, то выбирается данная кодировка;
- `URIHostname` — если имя в заголовке `Host:` начинается с имени кодировки или его синонима, то выбирается данная кодировка;
- `EnvVariable` — по переменной `FORCE_CHARSET`, определенной внешними модулями;
- `Dirprefix` — по началу имени каталога;
- `Useragent` — по HTTP-заголовку `User-Agent`.

❐ `SVDLA CharsetDefault имя_кодировки`

Принимается в качестве кодировки клиента, если все остальные способы не помогли.

❐ `SVDLA CharsetByPort имя_кодировки номер_порта`

Определяет кодировку по номеру порта, к которому произошло подключение.

Дополнительная обработка специфических случаев:

❐ `SVDLA AddHandler strip-meta-http .ext1 ...`

Удалять тэги `"META HTTP-EQUIV=... charset=..."` из HTML-файлов перед передачей их клиенту.

❐ `SVDLA CharsetBadAgent шаблон ...`

Для клиентских программам, подпадающих под шаблон, не будет выдаваться строка `charset=` в HTTP-заголовке `Content-type`.

❐ `SVDLA CharsetErrReject On | Off`

Если клиент запрашивает неизвестную кодировку в директиве `Accept/Accept-charset`, выдавать сообщение об ошибке или попытаться определить правильную кодировку.

❐ `SVDLA CharsetDisable On | Off`

Выключить модуль для данного сервера/каталога.

❐ `SVDLA CharsetRecodeFilenames On | Off`

Перекодировать имена файлов.

❐ `SVDLA CharsetOverrideExpires On | Off`

Если включен (`On`) — заменять заголовки `Expires`, сгенерированные другими модулями, на свои.

❐ `SVDLA CharsetDisableForcedExpires On | Off`

Если выключен (`Off`) — сервер выдает заголовок **Expires: 1 Jan 1970** для того, чтобы документ не кэшировался, если его кодировка определилась по `User-Agent` или `Accept-charset`.

- `SVDLA CharsetRecodeMethodsIn метод1 ...`

  Включить обработку запроса для данного метода: `GET`, `POST`, `PUT`, `ALL`, `NONE`.

- `SVDLA CharsetRecodeMethodsOut метод1 ...`

  Включить обработку ответа для данного метода: `GET`, `POST`, `PUT`, `ALL`, `NONE`.

Это далеко не все параметры, используемые при конфигурации сервера Apache. Для более полного описания конфигурационных директив смотрите документацию, идущую в комплекте с сервером Apache. А сейчас перейдем к рассмотрению непосредственно конфигурационных файлов.

Файл access.conf

В access.conf содержатся директивы, описывающие права доступа к каталогам и файлам Web-сервера. Обычно создается каталог /www/<имя_сервера>/, потому что при такой организации проще ориентироваться в структуре файлов.

Файл access.conf содержит секции `Directory`, `Location` и `Files`, которые ограничены одноименными директивами. В параметрах этих директив могут использоваться символы "`?`" и "`*`", а также регулярные выражения, предваряемые тильдой. В секции `Directory` помещаются инструкции, относящиеся к определенному каталогу на диске, в секции `Location` — относящиеся к виртуальному пути, в секции `Files` — относящиеся к файлу или группе файлов.

```
<Directory /www/r.com.ua>
# директивы, относящиеся ко всем документам, хранящимся в
каталоге /www/r.com.ua и вложенных в него
</Directory>

<Location /cgi-bin>
# директивы, относящиеся ко всем документам, доступным по
адресу http://<имя_сервера>/cgi-bin/ <путь_к_файлу>
</Location>

<Files /www/r.com.ua/form.htm>
# директивы, относящиеся к файлу form.htm из каталога
 /www/r/com.ua
</Files>
```

Различие между секциями `Directory` и `Location` состоит в том, что первая относится к каталогам на диске, вторая — к виртуальному пути (URL), ко-

торый браузер запрашивает у Web-сервера. И в той, и в другой могут присутствовать директивы `order`, `allow` и `deny`, которые позволяют ограничить доступ к каталогу или URL с различных машин.

При отсутствии специальных требований к безопасности можно указать `Options All` в секции `<Directory /www>`, иначе нужно описать параметры каждого каталога отдельно.

Пример файла access.conf

```
## access.conf — Apache HTTP server configuration file
##
# access.conf: Global access configuration
# Online docs at http://www.apache.org/

<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>

<Directory /www>
Options All
AllowOverride All
order allow,deny
allow from all
</Directory>
```

Файл srm.conf

Файл srm.conf содержит директивы, связанные с общими настройками структуры каталогов сервера. Обычно они не изменяются.

Файл httpd.conf

Конфигурационный файл httpd.conf является основным и содержит настройки, связанные с работой Web-сервера, виртуальных серверов, а также всех его программных модулей. Кроме того, именно в нем настраивается перекодирование русских букв при передаче от сервера к клиенту и обратно.

Директива `Port`, помещенная в самом начале файла, определяет номер порта для HTTP-сервера; по умолчанию это 80. При необходимости можно приписать серверу другой порт или несколько портов.

Директива `HostnameLookups` с параметром `on` или `off` включает или отключает преобразование численных IP-адресов клиентов, получивших данные с сервера, в доменные имена.

Директивы `User` и `Group` задают пользователя, который будет администрировать сервер. С точки зрения безопасности нежелательно указывать здесь существующего пользователя, имеющего доступ к каким-либо другим ресурсам или файлам. Лучше создать отдельного пользователя и группу специально для HTTP-сервера.

Директивы `ServerRoot`, `ErrorLog`, `CustomLog` определяют корневой каталог HTTP-сервера, путь к журналу регистрации ошибок (error_log) и путь к общему журналу обращений к серверу (access_log).

Настройка виртуальных серверов в файле httpd.conf

Обычно на одном физическом Web-сервере размещаются несколько так называемых "виртуальных" Web-серверов. Обычно это делается по нескольким причинам — экономия денег (для чего покупать отдельные серверы, когда можно на существующем сервере запустить сотню виртуальных серверов), экономия IP-адресов (и в конечном итоге — денег), проще с администрированием.

Виртуальные серверы могут иметь один и тот же IP-адрес и разные доменные имена, а могут иметь и разные IP-адреса. Для описания адресов и доменных имен виртуальных серверов служат следующие директивы: `ServerName`, `ServerAlias`, `NameVirtualHost` и `VirtualHost`. Они необходимы только если вам нужно установить более одного виртуального сервера.

Директива `ServerName`, находящаяся вне секций `VirtualHost`, определяет имя основного сервера, корневой каталог которого задан директивой `DocumentRoot` в файле srm.conf. Виртуальные серверы наследуют настройки основного; при необходимости специальной настройки соответствующие директивы помещаются в секции `VirtualHost`, относящейся к данному серверу.

Ниже приведен фрагмент конфигурационного файла для виртуальных серверов с различными IP-адресами:

```
...
ServerName www.lug.net

<VirtualHost 192.168.0.2>
DocumentRoot /www/lug.net
ServerName www.lug.net
ErrorLog /var/log/error_log.lug.net
```

```
CustomLog /var/log/access_log.lug.net combined
...
</VirtualHost>

<VirtualHost 192.168.11.6>
DocumentRoot /www/r.com
ServerName www.r.com
ErrorLog /var/log/error_log.r.com
CustomLog /var/log/access_log.r.com combined
...
</VirtualHost>
```

Ниже приведен фрагмент конфигурационного файла для виртуальных серверов с одинаковыми IP-адресами:

```
...
ServerName www.lug.net
NameVirtualHost 192.168.0.2

<VirtualHost 192.168.0.2>
DocumentRoot /www/lug.net
ServerName www.lug.net
ErrorLog /var/log/error_log.lug.net
CustomLog /var/log/access_log.lug.net combined
...
</VirtualHost>

<VirtualHost 190.168.0.2>
DocumentRoot /www/nug.net
ServerName www.nug.net
ServerAlias *.nug.net
ErrorLog /var/log/error_log.nug.net
CustomLog /var/log/access_log.nug.net combined
...
</VirtualHost>
```

Ссылки

- **http://bog.pp.ru/work/apache.html** — Apache: HTTP-сервер. Установка, настройка и русификация.

- **http://www.cs.ifmo.ru/education/documentation/rapacheman/index.shtml** — Артем Подстрешный. Работа с Web-сервером Russian Apache.
- **http://www.apache.org** — официальный сервер Apache.
- **http://apache.lexa.ru** — сервер группы разработчиков русского модуля Apache.
- Соответствующие HOWTO *(см. гл. 13)*:
  - Apache-Overview-HOWTO — Обзор Web-сервера Apache;
  - Building a Secure RedHat Apache Server HOWTO — настройка безопасного Apache;
  - Apache+DSO+mod_ssl+mod_perl+php+mod_auth_nds+mod_auth_mysql +mod_fastcgi mini-HOWTO — инсталляция Apache Web-сервера с поддержкой модулей mod_perl, mod_ssl и php;
  - Linux Apache SSL PHP/FI frontpage mini-HOWTO — настройка Web-сервера, поддерживающего динамически изменяемое содержимое.

Глава 20

FTP

Эта глава посвящена протоколу FTP, настройке сервера FTP, проблемам конфигурации и безопасности сервера.

Протокол FTP

Протокол FTP (File Transfer Protocol, протокол передачи файлов) предназначен для передачи файлов в сети Интернет. Этот протокол был разработан на заре эры Интернета и по сию пору остается востребованным и актуальным. Конечно, он несколько устарел, частично функции передачи файлов взял на себя Web-протокол HTTP, но, несмотря на это, протокол FTP, похоже, будет использоваться еще долгое время.

Передача файлов заключается в копировании целого файла с одного компьютера на другой. Для использования FTP-сервиса необходимо иметь учетную запись на сервере для авторизованного доступа, или воспользоваться анонимным доступом к FTP (anonymous FTP).

Протокол FTP, в отличие от большинства других протоколов, для пересылки файла использует два TCP-соединения. Одно соединение собственно для пересылки файла, а второе — для управления процессом передачи (управляющее соединение). Порт 20 предназначен для пересылки данных, а порт 21 для управляющего соединения. FTP-протокол может использовать как TCP-соединение, так и UDP-соединение.

Представление данных

Протокол передачи файлов допускает различные способы представления файлов и управления передачей. Ниже приведены критерии, от выбора которых зависит корректность передачи файлов по протоколу FTP.

Тип файла

Протокол должен знать, каков тип передаваемого файла. От этого зависит корректное представление его на компьютере (и в операционной системе) получателя.

- *ASCII-файлы*. Текстовый файл передается как NVT ASCII. При этом требуется, чтобы программа-отправитель конвертировала текстовый файл в NVT ASCII, а программа-получатель производила обратное преобразование. Конец каждой строки передается в виде NVT ASCII-символа возврата каретки (CR), после которого следует перевод строки (LF). Если отправитель текстового файла установит тип файла как бинарный — программы не будут преобразовывать передаваемый файл. Это вызывает проблемы несовместимости между текстовыми файлами DOS/Windows и текстовыми файлами UNIX. В DOS/Windows принято конец текстовой строки обозначать парой символов "возврат каретки/перевод строки" (CR LF), а в UNIX — перевод строки (LF).
- *EBCDIC-файлы*. Альтернативный способ передачи текстовых файлов, когда на обоих концах системы EBCDIC.
- *Бинарные файлы*. Данные между FTP-сервером и клиентом передаются как непрерывный поток битов.
- *Локальный тип файлов*. Способ передачи бинарных файлов между компьютерами, которые имеют различный размер байта. Количество битов в байте определяется отправителем. Обычно не используется.

Управление форматом

Применяется только для передачи ASCII- и EBCDIC-файлов.

- *Nonprint*. Файл не содержит информацию вертикального форматирования.
- *Telnet format control*. Файл содержит управляющие символы вертикального форматирования telnet, которые интерпретируются принтером.
- *Fortran carriage control*. Первый символ каждой строки это Fortran-символ управления форматированием.

Структура

Способы передачи структуры данных приведены ниже.

- *Структура файла*. Пересылаемый файл воспринимается в виде непрерывного потока байтов.
- *Структура записи*. Эта структура используется только в случае текстовых файлов.
- *Структура страницы*. Каждая страница передается с номером страницы (не рекомендуется использовать эту структуру).

Режим передачи

Определяет способ передачи файла по соединению данных.

- *Потоковый режим*. Передача файла осуществляется как поток байтов.
- *Блочный режим*. Передача файла осуществляется как последовательность блоков. Блок имеет управляющий заголовок и собственно пересылаемые данные.
- *Режим сжатия*. При передаче осуществляется замена неоднократно встречающихся повторяющихся байтов на байт и число его повторений.

Как видите, протокол FTP поддерживает большое количество представлений данных. Однако в реальной жизни большинством программного обеспечения наиболее часто используется нижеприведенное ограниченное подмножество представлений:

- тип файла — ASCII или двоичный;
- управление форматом — только nonprint;
- структура — только структура файла;
- режим передачи — только потоковый режим.

Управляющие команды FTP

Управляющие команды и ответы передаются по управляющему соединению между клиентом и сервером в формате NVT ASCII. В конце каждой строки присутствует пара символов "возврат каретки/перевод строки" (CR/LF).

В полном наборе команд насчитывается более 30-ти. В табл. 20.1 приведены наиболее часто используемые команды. Полный список команд можно посмотреть в соответствующем RFC.

Таблица 20.1. *Управляющие команды протокола FTP*

Команда	Описание
`ABOR`	Прервать последнюю команду FTP и любую передачу данных
`LIST` *список файлов*	Список файлов или каталогов
`PASS` пароль	Передача пароля пользователя
`PORT a, b, c, d, e, f`	IP-адрес клиента (`a.b.c.d`) и порт (`e×256 + f`)
`QUIT`	Разорвать соединение
`RETR` *имя файла*	Получить файл
`STOR` *имя файла*	Выгрузить файл

Таблица 20.1 *(окончание)*

Команда	Описание
SYST	Сервер возвращает тип системы
TYPE *тип*	Указать тип файла: A для ASCII, I для бинарного
USER *имя пользователя*	Передача имени пользователя (логин)

Ответы на управляющие FTP-команды

Ответы на управляющие FTP-команды состоят из трехзначного числа в формате ASCII и необязательного текстового сообщения, которое следует за числом.

Каждая из трех цифр в коде ответа имеет собственное значение. Расшифровка первой и второй цифр кода приведена в табл. 20.2.

Таблица 20.2. *Значения первой и второй цифр в коде ответа на управляющие команды*

Ответ	Описание
1xx	Положительный предварительный отклик. Действие началось, однако необходимо дождаться еще одного отклика перед отправкой следующей команды
2xx	Положительный отклик о завершении. Может быть отправлена новая команда
3xx	Положительный промежуточный отклик. Команда принята, однако необходимо отправить еще одну команду
4xx	Временный отрицательный отклик о завершении. Требуемое действие не произошло, однако ошибка временная, поэтому команду необходимо повторить позже
5xx	Постоянный отрицательный отклик о завершении. Команда не была воспринята и повторять ее не стоит
x0x	Синтаксическая ошибка
x1x	Информация
x2x	Соединения. Отклики имеют отношение либо к управляющей команде, либо к соединению данных
x3x	Аутентификация и бюджет. Отклик имеет отношение к регистрации пользователя в системе или командам, связанным с бюджетом
x4x	Не определено
x5x	Состояние файловой системы

Третья цифра дает уточняющее определение сообщению об ошибке. В табл. 20.3 приведены соответствующие коды и пояснения.

Таблица 20.3. *Значения третьей цифры в коде ответа на управляющие команды*

Ответ	Описание
125	Соединение данных уже открыто; начало передачи
200	Команда выполнена
214	Сообщение о помощи
331	Имя пользователя принято, необходимо ввести пароль
425	Невозможно открыть соединение данных
452	Ошибка записи файла
500	Неизвестная команда
502	Нереализованный тип `MODE`

Обычно каждая FTP-команда генерирует однострочный ответ. Если необходим ответ, состоящий из нескольких строк, то первая строка содержит дефис вместо пробела после трехзначного кода отклика, а последняя строка содержит тот же самый трехзначный код отклика, за которым следует пробел.

Управление соединением

Использовать соединение данных можно тремя способами:

1. Отправка файлов от клиента к серверу.
2. Отправка файлов от сервера к клиенту.
3. Отправка списка файлов или каталогов от сервера к клиенту.

Третий способ необходимо пояснить. FTP-сервер посылает список файлов по соединению данных — при таком использовании канала данных появляется возможность избежать любых ограничений в строках, накладывающихся на размер списка каталога, и несколько упрощает обмен информацией.

Управляющее соединение остается в активизированном состоянии все время, пока установлено соединение клиент-сервер, но соединение данных может устанавливаться и отключаться по необходимости. Рассмотрим, как выбираются номера портов для соединения данных, и кто осуществляет активное открытие, а кто пассивное.

Основной режим передачи — потоковый режим. В этом режиме конец файла обозначает закрытие соединения данных. Из этого вытекает, что для пе-

редачи каждого файла требуется новое соединение данных. Обычная процедура выглядит следующим образом:

1. Создание соединения данных осуществляется клиентом.
2. Клиент выбирает динамически назначаемый номер порта на компьютере клиента для своего конца соединения данных и осуществляет пассивное открытие с этого порта.
3. Клиент посылает номер порта на сервер по управляющему соединению с использованием команды `PORT`.
4. Сервер принимает номер порта с управляющего соединения и осуществляет активное открытие на этот порт компьютера клиента. Сервер всегда использует порт 20 для соединения данных.

Сервер всегда осуществляет *активное открытие* соединения данных. Обычно сервер также осуществляет *активное закрытие* соединения данных, за исключением тех случаев, когда клиент отправляет файл на сервер в потоковом режиме, который требует, чтобы клиент закрыл соединение.

Если клиент не выдает команду `PORT`, сервер осуществляет активное открытие на тот же самый номер порта, который использовался клиентом для управляющего соединения.

Программное обеспечение

Для работы по протоколу FTP необходимы две программы — сервер и клиент. Клиентских программ — очень много. От простейших, работающих в командной строке, до имеющих весьма развитый графический интерфейс. Любой современный Web-браузер способен выступать в роли FTP-клиента. Поэтому на клиентских программах останавливаться не будем, а перейдем сразу к программному обеспечению сервера FTP.

Сегодня стандартом de-facto для множества дистрибутивов является использование в качестве программного обеспечения пакета wu-ftp (Washington University at Saint Louis FTP daemon).

Пакет wu-ftp

Программный пакет написан в Вашингтонском университете. Достаточно гибок и настраиваем. Обычно поставляется вместе с дистрибутивом, поэтому установка его не представляет сложности. Основная задача — правильно сконфигурировать систему и настроить доступ.

Команды

Как уже упоминалось ранее, FTP-серверы имеют свои наборы команд, иногда несколько отличающиеся друг от друга. В табл. 20.4 приведен список стандартных команд сервера wu-ftp.

Таблица 20.4. *Стандартные команды FTP-сервера wu-ftp*

Команда	Описание
ABOR	Прервать предыдущую команду
APPE	Добавить к файлу
CDUP/XCUP	Подняться на каталог вверх
CWD /XCWD	Поменять текущий каталог
DELE	Удалить файл
HELP	Получить справочную информацию
LIST	Получить список файлов и каталогов в текущем каталоге
MKD /XMKD	Создать каталог
MDTM	Показать время последнего изменения файла
MODE	Задать режим пересылки файла
NLST	Получить список файлов
PASS	Передать пароль пользователя
PASV	Вход в "пассивный" режим передачи
PORT	Задает порт для последующей передачи данных
QUIT	Окончание сеанса
REST	Продолжить прерванную передачу данных
RETR	Получить файл
RMD/XRMD	Удалить каталог
RNFR	Исходное имя переименовываемого файла
RNTO	Новое имя переименовываемого файла
SIZE	Получить размер файла
STAT	Показать состояние сервера
STOR	Сохранить файл
STOU	Сохранить файл с уникальным именем
STRU	Задает структуру передачи
SYST	Вывести тип операционной системы, на которой работает сервер

Таблица 20.4 *(окончание)*

Команда	Описание
TYPE	Задать тип передачи
USER	Передать имя пользователя

Помимо вышеприведенных команд, сервер wu-ftp имеет несколько специфичных, которые приведены в табл. 20.5.

Таблица 20.5. *Нестандартные команды FTP-сервера wu-ftp*

Команда	Описание
SITE EXEC	Запустить программу на выполнение
SITE GROUP	Сменить группу
SITE GPASS	Передать пароль группы
SITE IDLE	Задать время неактивности пользователя, по истечении которого соединение разрывается
SITE MINFO	Показать список файлов более новых, чем указанная дата. Команда выдает более расширенную информацию, чем NEWER
SITE NEWER	Показать список файлов более новых, чем указанная дата
SITE UMASK	Задать umask для файлов, сохраняемых пользователем на сервере

Конфигурирование сервера

Конфигурирование сервера wu-ftp проводится в два этапа. Первый — компилирование сервера со специфическими для вашего случая свойствами. Этот вариант мы описывать не будем, поскольку для создания простого сервера достаточно rpm-пакета, входящего в дистрибутив. Второй этап — использование конфигурационных файлов сервера, на которых мы сейчас и остановимся.

Как вы уже знаете, конфигурационные файлы находятся в каталоге /etc. В идеале сервер wu-ftp использует следующие конфигурационные файлы:

- ftpaccess
- ftphosts
- ftpservers
- ftpgroups
- ftpusers
- ftpconversion

Рассмотрим подробно каждый конфигурационный файл.

Файл ftpaccess

Этот конфигурационный файл, используется для определения прав доступа к серверу. Здесь определяется, какие и сколько пользователей, могут получить доступ к серверу, а также важные элементы настройки безопасности сервера.

Рассмотрим подробно конфигурационные параметры, используемые в этом файле.

Управление правами доступа:

- `autogroup имя_группы класс ...` — в том случае, если анонимный пользователь является членом указанного класса, то сервер использует заданную группу, что позволяет анонимным пользователям из разных классов получать доступ к различным наборам каталогов;
- `class класс typelist шаблон_адресов ...` — позволяет закрепить клиента за указанным классом, исходя из IP-адреса и типа клиента, где:
  - `typelist` — список из ключевых слов, обычно `anonymous`, `guest` и `real` (зарегистрированные на локальном хосте — /etc/passwd), через запятую;
  - `шаблон_адресов` — шаблон имени или адреса хоста клиента или адрес:маска или имя файла (имя файла должно начинаться с /, а файл — содержать шаблоны адресов);
- `deny шаблон_адресов файл_с_текстом_сообщения` — запретить доступ клиентов с указанного адреса с выдачей текста сообщения;
- `guestgroup имя_группы ...` — если реальный пользователь является членом указанной группы, то с ним поступают так же, как с анонимным. Вместо имени группы можно использовать номер, перед которым надо поставить знак процента, или интервал номеров, или звездочку для всех групп;
- `guestuser имя_пользователя ...` — аналогично `guestgroup`, но используется имя реального пользователя;
- `realgroup имя_группы ...` — инвертирует действие `guestgroup` и `guestuser`;
- `realuser имя_пользователя ...` — инвертирует действие `guestgroup` и `guestuser`;
- `defumask umask [ класс ]` — задание `umask`, применяемой при создании файлов;
- `keepalive { yes | no }` — установить `TCP SO_KEEPALIVE`;
- `timeout accept секунд` — сколько ожидать входного соединения для передачи данных (`PASV`);

- `timeout connect секунд` — сколько ожидать установления выходного соединения для передачи данных (`PORT`);
- `timeout data секунд` — максимальный период неактивности пользователя при передаче данных;
- `timeout idle секунд` — сколько ожидать следующей команды;
- `timeout maxidle секунд` — поскольку клиент имеет возможность установить `idle` самостоятельно, параметр `maxidle` позволяет установить верхний предел для клиента;
- `timeout RFC931 секунд` — максимальное время ожидания ответа для протокола `ident`;
- `file-limit [ raw ] { in | out | total } число [ класс ]` — ограничивает число передаваемых файлов;
- `byte-limit [ raw ] { in | out | total } число [ класс ]` — ограничивает число передаваемых байтов;
- `limit-time { * | anonymous | guest } минут` — ограничение времени сессии. Реальные пользователи не ограничиваются никогда;
- `guestserver [ имя_серверного_хоста ]` — гостевой и анонимный доступ предоставляется только к указанному хосту. Имеет смысл, если сервер обслуживает несколько виртуальных доменов;
- `limit класс число временной_интервал имя_файла_с_сообщением` — ограничение на число одновременно работающих клиентов из данного класса. Проверка производится только в момент входа. Если к сеансу применимо несколько команд `limit`, то используется первая;
- `noretrieve [ absolute | relative ] { class=класс } имя_файла ...` — запретить клиенту читать указанные файлы. Если имя начинается с /, то только этот файл, иначе любой файл с соответствующим именем. Если указан каталог, то любой файл из этого каталога;
- `allowretrieve [ absolute | relative ] { class=класс } имя_файла ...` — отменить действие директивы `noretrieve`;
- `loginfails число` — после указанного числа неудачных попыток зайти на сервер, сделать запись в журнале и разорвать соединение;
- `private { yes | no }` Нестандартные команды `SITE GROUP` и `SITE GPASS` позволяют пользователю поменять текущую группу.

Выдача сообщений клиенту:

- `greeting { full | brief | terse | text строка }` — определяет, какой текст будет выдаваться в строке приветствия:
  - `full` — имя хоста и версия сервера;

- `brief` — имя хоста;
- `terse` — ничего, кроме факта готовности к обслуживанию;
- `text` — произвольная строка текста;

- `banner имя_файла` — определяет текст сообщения, выдаваемого клиенту до ввода имени/пароля;
- `hostname имя_хоста` — определяет имя хоста по умолчанию (имя локального хоста);
- `email адрес` — адрес администратора;
- `message имя_файла { LOGIN | CWD=имя_каталога { класс } }` — содержимое файла выдается клиенту при входе или смене каталога;
- `readme имя_файла { LOGIN | CWD=имя_каталога { класс } }` — при входе или смене каталога сервер информирует клиента о наличии указанного файла и дате его создания или последней модификации.

Журнализация:

- `log commands ???сок_типов` — выводить в журнал все команды клиента, где `список_типов` — список через запятую слов `real`, `guest` и `anonymous`;
- `log transfers список_типов список_направлений` — выводить в журнал пересылки файлов, где `список_типов` — список через запятую слов `real`, `guest` и `anonymous`; `список_направлений` — список через запятую слов `incoming` и `outbound`;
- `log security список_типов` — выводить в журнал нарушения правил безопасности, где `список_типов` — список через запятую слов `real`, `guest` и `anonymous`;
- `log syslog` — перенаправлять сообщения о пересылках в syslog вместо файла xferlog;
- `log syslog+xferlog` — направлять сообщения о пересылках в syslog и файл xferlog.

Виртуальные серверы:

- `daemonaddress ip-адрес` — прислушиваться к соединениям только по указанному адресу;
- `virtual ip-адрес { root | banner | logfile } имя_файла` — определить соответственно: корень файловой системы, файл, содержащий баннер приветствия, и журнал для указанного виртуального сервера;
- `virtual ip-адрес { hostname | email } строка` — определить имя хоста (отображаемое в приветствии) и адрес администратора для указанного виртуального сервера;

- `virtual ip-адрес private` — закрыть анонимный доступ по указанному адресу;
- `virtual ip-адрес incmail email-адрес` — кого извещать в случае анонимной загрузки файлов;
- `virtual ip-адрес mailfrom email-адрес` — какой обратный адрес подставлять при рассылке сообщений об анонимной загрузке файлов;
- `defaultserver { deny | allow } имя_пользователя ...` — по умолчанию доступ разрешен всем;
- `defaultserver private` — закрыть анонимный доступ;
- `defaultserver incmail email-адрес` — кого извещать в случае анонимной загрузки файлов;
- `defaultserver mailfrom email-адрес` — какой обратный адрес подставлять при рассылке сообщений о анонимной загрузке файлов.

Права доступа:

- `{ chmod | delete | overwrite | rename | umask } { yes | no } список_типов` — разрешить/запретить пользователям выполнять соответствующее действие. По умолчанию — все разрешено. `Список_типов` — список через запятую слов `anonymous`, `guest`, `real` или `class=имя_класса`;
- `passwd-check { none | trivial | rfc822 } ( { enforce | warn } )` — уровень проверки правильности вводимых анонимными пользователями в качестве пароля email-адресов и реакция сервера в случае ошибки:
  - `none` — никакой проверки;
  - `trivial` — строка должна содержать @;
  - `rfc822` — полная проверка согласно стандарту rfc-822;
  - `warn` — если обнаружена ошибка, то выдавать предупреждение;
  - `enforce` — если обнаружена ошибка, то не впускать пользователя;
- `deny-email email-адрес` — считать данный адрес неправильным;
- `path-filter список-типов имя_файла_сообщения шаблон_допустимых_имен шаблон_недопустимых ...` — когда пользователь типа из списка типов пытается загрузить файл на сервер, то сервер проверяет имя файла на соответствие регулярному выражению, указанному в шаблоне допустимых имен, и на несоответствие ни одному из регулярных выражений в шаблонах недопустимых имен;
- `upload [ absolute | relative ] [ class=имя-класса ]... [ – ] корень шаблон)_каталога { yes | no } owner group mode [ dirs | nodirs ] [ dir_mode ]` — определяет каталоги, в которые разрешено/запрещено записывать файлы пользователям из указанного класса.

Все создаваемые файлы будут иметь соответствующие права доступа и принадлежность;

- `throughput` — позволяет задать скорость передачи определенных файлов на определенные хосты;
- `anonymous-root` *`корень [ класс ] ...`* — определяет корневой каталог (`chroot`) для анонимных пользователей указанного класса и их домашний каталог;
- `guest-root корень [ интервал-uid ] ... корень` — определяет аргумент `chroot` для гостевых пользователей и их домашний каталог. Можно задавать отдельные `uid` или интервалы через дефис;
- `deny-uid интервал ...` — запрещает доступ к серверу определенным пользователям и может использоваться вместо файла ftpusers;
- `deny-gid интервал ...` — запрещает доступ к серверу определенным группам пользователей и может использоваться вместо файла ftpusers;
- `allow-uid интервал ...` — разрешает доступ к серверу определенным пользователям и может использоваться вместо файла ftpusers;
- `allow-gid интервал ...` — разрешает доступ к серверу определенным группам пользователей и может использоваться вместо файла ftpusers;
- `restricted-uid интервал ...` — разрешить реальному или гостевому пользователю доступ вовне его домашнего каталога;
- `restricted-gid интервал ...` — разрешить группе пользователей доступ вовне его домашнего каталога;
- `unrestricted-uid интервал ...` — запретить реальному или гостевому пользователю доступ вовне его домашнего каталога;
- `unrestricted-gid интервал ...` — запретить группе пользователей доступ вовне его домашнего каталога;
- `site-exec-max-lines число [ класс ] ...` — ограничивает число строк, посылаемых командой `SITE EXEC`;
- `dns refuse_mismatch файл_с_сообщением [ override ]` — выдавать сообщение, если прямой и обратный адреса клиента не совпадают. Если не указано `override`, то прекращать сеанс;
- `dns refuse_no_reverse файл_с_сообщением [ override ]` — выдавать сообщение, если клиент не имеет обратного адреса. Если не указано `override`, то прекращать сеанс.

Разное:

- `alias строка имя_каталога` — позволяет переходить в указанный каталог по команде `cd строка` из любого каталога;

- `cdpath имя_каталога` — добавляет каталог к переменной `cdpath`, которая используется в качестве списка поиска для команды `cd`;
- `compress { yes | no } шаблон_классов ...` — разрешить/запретить компрессию/декомпрессию для классов, подпадающих под шаблон;
- `tar { yes | no } шаблон_классов ...` — разрешить/запретить использование `tar` для классов, подпадающих под шаблон;
- `shutdown имя_управляющего_файла` — файл содержит описание для остановки сервера;
- `passive address возвращаемый_ip-адрес cidr_шаблон` — если клиент выдает команду `PASS`, то сервер определяет возвращаемый адрес исходя из соответствия IP-адреса клиента CIDR-шаблону;
- `pasive ports cidr_шаблон min max` — определяется интервал портов, из которых сервер выбирает порт для прослушивания случайным образом и передает его номер клиенту;
- `pasv-allow класс шаблон_адресов` — позволяет пользователям указанного класса соединяться не только с исходного адреса, но и с заданных шаблоном адресов;
- `port-allow класс шаблон_адресов` — позволяет пользователям данного класса указывать в команде `PORT` адрес, подходящий под шаблон;
- `lslong команда [ параметры ]` — какую команду и параметры использовать для генерации расширенного списка файлов в каталоге;
- `lsshort команда [ параметры ]` — какую команду и параметры использовать для генерации списка файлов в каталоге;
- `lsplain команда [ параметры ]` — какую команду и параметры использовать для генерации списка файлов в каталоге;
- `incmail email-адрес` — кого извещать в случае анонимной загрузки файлов;
- `mailserver имя-хоста` — какой почтовый сервер использовать для рассылки сообщений об анонимной загрузке файлов;
- `mailfrom email-адрес` — какой обратный адрес подставлять при рассылке сообщений об анонимной загрузке файлов.

Файл ftpservers

Этот файл определяет набор файлов конфигурации для каждого виртуального сервера. Каждая строка в этом конфигурационном файле описывает виртуальный сервер и состоит из двух полей:

- имя и IP-адрес виртуального сервера;

- имя каталога, содержащего конфигурационные файлы. Имена файлов фиксированы: ftpaccess, ftpusers, ftpgroups, ftphosts, ftpconversions. Если какой-либо конфигурационный файл отсутствует, то вместо него используется конфигурационный файл основного сервера.

Файл ftpconversions

В этом файле каждая строка описывает возможное преобразование файлов "на лету" и состоит из 8-ми полей, разделенных двоеточиями:

- удаляемый префикс;
- удаляемый суффикс;
- добавляемый префикс;
- добавляемый суффикс;
- используемая для преобразования внешняя программа и ее параметры;
- типы преобразуемого файла: T_REG — обычный файл, T_ASCII — текстовый, T_DIR — каталог или сочетание перечисленных типов;
- опции: `O_COMPRESS`, `O_UNCOMPRESS`, `O_TAR` или их сочетание;
- комментарий к строке преобразования.

Файл ftpgroups

Этот файл используется для поддержки функционирования нестандартных команд типа `SITE GROUP` и `SITE GPASS`. В файле ftpgroups находятся строки, состоящие из трех полей, разделенных двоеточием:

- задаваемое клиентом имя группы;
- зашифрованный пароль группы;
- реальное имя группы.

Файл ftphosts

Этот файл предназначен для ограничения доступа к FTP-серверу с определенных хостов. Используется всего две команды:

- `allow имя_пользователя шаблон_IP-адреса ...` — разрешить доступ;
- `deny имя_пользователя шаблон_IP-адреса ...` — запретить доступ.

Файл ftpusers

Этот файл предназначен для того чтобы запретить некоторым реальным пользователям доступ к FTP-серверу. Используется обычно для повышения безопасности системы, чтобы исключить доступ пользователей типа root, news и т. п.

Параметры запуска программ, входящих в пакет

ftpd

Эта программа — собственно, сервер. При запуске можно использовать следующие ключи (приведены только основные):

- `-d` — выдавать отладочную информацию;
- `-l` — вести протокол по каждой сессии;
- `-t число_секунд` — время бездействия клиента, после которого сервер автоматически разрывает соединение (может быть изменен клиентом);
- `-T число_секунд` — время бездействия клиента, после которого сервер автоматически разрывает соединение;
- `-a` — использовать файл ftpaccess;
- `-A` — не использовать ftpaccess;
- `-i` — вести протокол о полученных файлах в файле xferlog;
- `-I` — запрещает использовать протокол ident;
- `-o` — записывать имена переданных файлов в xferlog;
- `-X` — делать записи о полученных и переданных файлах в файле syslog;
- `-u umask` — маска файла по умолчанию;
- `-w` — записывать заходы в wtmp;
- `-W` — не записывать заходы в wtmp;
- `-s` — самостоятельный запуск без использования inetd;
- `-S` — самостоятельный запуск без использования inetd, отсоединиться от терминала;
- `-p порт` — управляющий порт, по умолчанию берется FTP-порт из файла /etc/services, при использовании inetd не применяется;
- `-P порт` — порт данных, по умолчанию берется значение `ftp-data 20` (20-й порт) из файла /etc/services;
- `-q` — использовать файлы для хранения номеров процессов;
- `-Q` — не использовать файлы для хранения номеров процессов; при использовании этого параметра не будет работать ограничение на количество пользователей в классе;
- `-r rootdir` — сделать chroot (определение корневого каталога для программы) немедленно после запуска, не дожидаясь ввода имени пользователя; используется для построения защищенной (может быть избыточно) системы.

ftpwho

Эта утилита показывает информацию о каждом подключенном в данный момент клиенте.

ftpcount

Утилита показывает текущее и максимальное количество пользователей для каждого класса пользователей.

ftpshut

Утилита используется для безаварийного завершения работы FTP-сервера. Представляют интерес следующие ключи запуска:

- `-l минуты` — позволяет задать время, за сколько минут до завершения работы сервера запрещать установку новых соединений;
- `-d минуты` — задает время, за сколько минут до завершения работы сервера разрывать текущие соединения;
- `время_завершения` — время завершения работы сервера. Может быть задано в следующем виде:
  - `now` — немедленно завершить работу сервера;
  - `+минут` — через сколько минут завершить работу сервера;
  - `ЧЧММ` — время завершения работы сервера.

ftprestart

Утилита производит запуск FTP-сервера, если он был завершен командой `stop`.

ckconfig

Утилита, позволяющая проверить конфигурацию FTP-сервера. Позволяет выявить случаи явных ошибок.

Формат файла журнала xferlog

Как и положено, FTP-сервер ведет журнал событий. Файл журнала событий называется xferlog, и в нем протоколируется любой прием или передача файла. Информация о событии записывается строкой, состоящей из более чем десятка полей. Ниже приведено описание полей записи.

- Название дня недели, например `Sat`.
- Название месяца.
- День.

- Часы:минуты:секунды.
- Год.
- Продолжительность передачи в секундах.
- Имя удаленного хоста.
- Размер файла в байтах.
- Имя файла.
- Тип передачи:
  - `a` — текстовый;
  - `b` — бинарный.
- Действие над файлом в процессе передачи:
  - `C` — сжат;
  - `U` — разархивирован;
  - `T` — обработан программой tar;
  - `_` (символ подчеркивания) — не было произведено никаких действий.
- Направление передачи:
  - `o` — с сервера;
  - `i` — на сервер.
- Тип пользователя:
  - `a` — анонимный;
  - `g` — guest;
  - `r` — real.
- Имя реального пользователя или идентификационная строка для анонимного или гостевого пользователя.
- Имя сервиса.
- Способ аутентификации:
  - `0` — отсутствует;
  - `1` — ident (rfc931).
- Аутентифицированный идентификатор пользователя. Если аутентификация не использовалась — `*`.
- Состояние передачи:
  - `c` — передача была закончена;
  - `i` — не закончена.

Безопасность

После (а лучше во время) конфигурации очень желательно подумать о безопасности FTP-сервера. Зачастую неправильно сконфигурированный FTP-сервер становится тем слабым местом, через которое осуществляется прорыв безопасности вашей операционной системы.

Чрезвычайно важно, чтобы ваши анонимные и гостевые пользователи FTP не имели доступа к реальному командному процессору. Тогда, даже если они по каким-либо причинам смогут покинуть окружение FTP, то не смогут выполнить никаких посторонних задач. Для обеспечения этого требования убедитесь, что в файле /etc/passw у пользователей guest и anonymous в поле, где находится командная оболочка пользователя, находится что-то типа /dev/null.

Файл ftpusers должен содержать список следующих псевдо-пользователей, которым будет отказано в подключении к FTP-серверу:

```
root        lp          mail        operator
bin         sync        news        games
daemon      shutdown    uucp        nobody
adm         halt
```

Обычно FTP-сервер разрешает загрузку файлов на сервер (upload) всем пользователям. Однако необходимо запретить пользователям загружать свои файлы в некоторые каталоги (а иногда и во все). Для этого в файле ftpaccess необходимо прописать опцию `upload` с ключом `no` и указать каталог, на который налагается запрет.

Иногда желательно запретить пользователям получение с FTP-сервера некоторых каталогов и файлов. Для этого в файле ftpaccess добавляем строку `noretrieve` с каталогом, куда необходимо запретить доступ пользователям.

Ссылки

- RFC 959 — RFC, описывающий FTP-протокол.
- **www.bog.pp.ru/work/ftpd.html** — описание конфигурирования сервера wu-ftp.
- **ftp.wu-ftpd.org** — исходный текст пакета wu-ftp.
- **www.westnet.com/providers/multi-wu-ftpd.txt** — описание настройки виртуальных FTP-серверов.
- **ftp.fni.com/pub/wu-ftpd/guest-howto** — HOWTO по настройке анонимного доступа на FTP-сервер.

Глава 21

Сервер новостей INN

Одним из популярных сервисов, доступных в Интернете, является Usenet (News, новости, телеконференции, эхо-конференции). Это похоже на электронную доску объявлений или Web-форумы. В Usenet минимальной единицей информации является статья. Статья помещается в конференцию. Каждая конференция имеет свою тему. Конференций может быть множество — несколько десятков тысяч. Конференции имеют иерархическую структуру. Имя образуется из имени родительской иерархии, к которому через точку добавляется имя конференции. К примеру — **fido7.ru.linux**, где fido7 — корень иерархии, показывающий, что группа новостей импортирована из эхо-конференций FIDO, ru — русскоязычная (российская, ранее было su — Советский Союз), linux — конференция посвящена Linux. Для приема и передачи статей используются News-серверы (Usenet-серверы). Эти серверы производят синхронизацию (обмен статьями) между собой. Для передачи и приема статей используется протокол NNTP (Network News Transfer Protocol, сетевой протокол передачи новостей).

Каждая телеконференция имеет свой каталог статей, собранных в соответствующем разделе. Каждая статья в телеконференции имеет оригинальные атрибуты:

- ❒ идентификационный номер статьи;
- ❒ автор статьи;
- ❒ длина статьи;
- ❒ тема статьи.

Принятые статьи накапливаются, но не более чем за определенный период (свой для каждой телеконференции). Устаревшие материалы уничтожаются.

Наиболее популярным программным обеспечением для создания сервера телеконференций является пакет InterNetNews, INN.

Сервер новостей InterNetNews (INN)

Пакет INN является одним из старейших пакетов программного обеспечения, предназначенного для создания сервера новостей. Использует стандартный протокол NNTP. Новости хранятся на сервере в дереве каталогов, имена которых формируются из имен телеконференций и повторяют их иерархическую структуру.

Работа пакета INN

Основной процесс — innd. Постоянно запущен в системе, ожидая и принимая поток статей по протоколу NNTP от серверов новостей. Прослушивает порт 119 на наличие входящих соединений. Ведет список активных групп, список статей, статьи, базу заголовков статей, пакеты статей для рассылки по серверам новостей, журналы.

При соединении клиентов для чтения новостей программа innd передает управление демону nnrpd. Этот демон просматривает файл nnrp.access для определения прав доступа к локальной базе статей.

Для управления работой innd — добавления, удаления групп, статей, серверов новостей, изменения параметров работы используется программа ctlinnd.

Удалением старых статей с истекшим сроком хранения занимаются программы expire и expireover, которые удаляют устаревшие файлы, не останавливая innd.

Для автоматического обновления списка новостей используются управляющие сообщения.

Управляющие сообщения

Представляют собой обычные статьи в обычной группе новостей, имеющие заголовок "Control:". Встретив такую статью, innd обрабатывает записанную в ней команду и сохраняет статью. Управляющие сообщения могут содержать следующие команды:

- `cancel Message-ID` (обрабатывается самим innd);
- иначе первое слово рассматривается как имя программы (ищется в каталоге ${BIN}/control). Если программа не существует, то вызывается программа default.

Сообщения запоминаются в псевдогруппе control. Если создать подгруппу control.имя_команды, то все соответствующие статьи будут помещаться в эту подгруппу (очень рекомендуется создать control.cancel, control.newgroup, control.rmgroup, control.checkgroups).

Настройка системы INN

Сервер новостей INN по возможностям и сложности настройки весьма напоминает пакет sendmail. Исходя из этого не приходится удивляться огромному количеству конфигурационных файлов и параметров конфигурации. Находятся конфигурационные файлы в каталоге /etc/news. Рассмотрим наиболее значимые из них:

- /etc/news/actsync.cfg — этот файл, как и следующий, используется для конфигурации автоматического изменения списка групп новостей. Обычно, чтобы в последствии не разгребать завалы неведомо откуда взявшихся групп новостей с огромным трафиком, добавление новых групп новостей возлагается на администратора системы;
- /etc/news/actsync.ign — этот файл, как и предыдущий, используется для конфигурации автоматического изменения списка групп новостей;
- /etc/news/control.ctl — в этом файле описывается, как обрабатывать управляющие сообщения. Каждая его строка задает действие. Строки состоят из четырех полей, разделенных двоеточием. Первое поле задает команду, к которой применяется действие (можно указать ключевое слово `all`), последнее поле — действие. Строки просматриваются по порядку. Используется последняя подошедшая. Возможные действия:
  - `doit`
  - `doifarg`
  - `doit=отдельный_журнал`
  - `doit=mail`
  - `doit= (без журнализации)`
  - `drop`
  - `log` (запись в журнал — `errlog`)
  - `log=отдельный_журнал`
  - `mail`

  Описание действий смотрите в документации на сервер INN. Для увеличения безопасности и устойчивости системы рекомендуется не использовать управляющие сообщения, а в файл control.ctl записать единственную строку `all:*:*:drop` — не делать никакой обработки вообще;
- /etc/news/cycbuff.conf — файл содержит конфигурацию метода хранения CNFS, обычно не используется (подробности смотрите в документации на пакет INN);
- /etc/news/distrib.pats — файл используется программами посылки статей, в частности, inews — для определения области распространения статьи.

Область распространения определяется по шаблону группы новостей и приоритету. Обычно файл не используется;

- /etc/news/expire.ctl — файл определяет, через какое время статьи в базе устаревают. Использование файла зависит от метода хранения статей. В частности, метод хранения CNFS самостоятельно удаляет старые статьи. В этом же файле определяется, сколько времени хранить в "истории" информацию об удаленных или отвергнутых статьях.

  В начале файла обязательно должна находиться строка, определяющая срок хранения записи об идентификаторах статей в файле history после удаления тела статьи. Это позволяет отклонить статью, если поставщик новостей вновь предложит ее в определенный промежуток времени. Эта строка имеет следующий формат: `/remember/:время`, где `время` — срок хранения в днях, по истечении которого из системы удаляются идентификаторы старых статей.

  Здесь можно задать для различных групп или набора иерархий групп различные сроки хранения статей. Правила хранения статей задаются следующей строкой, состоящей из пяти полей, разделенных двоеточием:

  ```
  Шаблоны_имени_группы_через_запятую:флаг:min:default:max
  ```

  - первое поле в строке задает группу или иерархию, удовлетворяющую шаблону;
  - второе поле содержит флаг, который определяет, к какому типу групп применять данное условие:
    - `A` — все группы;
    - `M` — только модерируемые;
    - `U` — только немодерируемые;
    - `X` — все группы. Если статья была послана в несколько групп и удовлетворяет данному шаблону, то она удалится не только из данной группы, но и из всех остальных групп, в которые была отослана;
  - третье поле задает минимальное число дней хранения. Так же можно использовать ключевое слово `never`;
  - четвертое поле определяет число дней хранения по умолчанию. Так же можно использовать ключевое слово `never`;
  - пятое поле определяет максимальное число дней хранения статьи в базе. Так же можно использовать ключевое слово `never`;

- /etc/news/incoming.conf — в файле определяется, кто может служить для нашего сервера поставщиком новостей. Определяющая строка имеет вид: имя, двоеточие, пробел, значение. В качестве имени используются следующие слова:
  - `Hostname`. В качестве значения — список полных доменных имен хостов или десятичных IP-адресов через запятую;

- `streaming`. В качестве значения — `true` или `false`; параметр определяет, разрешен ли потоковый режим;
- `max-connections` — параметр определяет максимальное число параллельных соединений;
- `password` — если сервер новостей требует авторизации, здесь прописывается пароль, обычно не используется;
- `patterns`. В качестве значения — шаблон групп, принимаемых с указанного хоста;
- `noresendid`. В качестве значения — `true` или `false`; определяет, должен ли сервер новостей посылать ответ `431 RESENDID` в потоковом режиме и `436 Retry later` в непотоковом в ответ на попытку послать статью, которая уже была принята;

❐ /etc/news/inn.conf — файл содержит глобальные параметры сервера новостей и параметры, используемые при формировании заголовков статей, создаваемых на этом сервере. Все изменения, сделанные в этом файле, считываются демоном innd только после перезагрузки сервера новостей. Строки в конфигурационном файле имеют следующий формат:

```
имя: пробел значение
```

Ниже описываются имена параметров и их значения:

- `fromhost` — параметр используется при формировании заголовка `From:`, если его нет. Переменная окружения `FROMHOST` переопределяет это значение. По умолчанию это полное доменное имя локальной машины;
- `moderatormailer` — имя хоста, содержащего псевдонимы для всех модерируемых групп. Рекомендуется использовать файл moderators;
- `organization` — определяет содержимое заголовка `Organization:`, если таковой отсутствует. Если определена переменная окружения `ORGANIZATION`, то она переопределяет это значение;
- `pathhost` — определяет, какое имя локального узла помещается в заголовок `Path:`. По умолчанию это полное доменное имя локальной машины;
- `server` — определяет имя NNTP-сервера, на котором должны публиковаться созданные статьи. В том случае, если определена переменная окружения `NNTPSERVER`, то она изменяет это значение;
- `domain` — определяет имя домена, к которому принадлежит локальная машина;
- `overviewmmap` — определяет, будут ли программы expire, nnrpd и makehistory использовать mmap для доступа к файлу overview;

- `storageapi` — определяет способ хранения статей: `false` для традиционного метода хранения статей; `true` — для хешированных имен, `cnfs` — для кольцевых буферов:
  - ◊ традиционный метод — каждая статья в отдельном файле; каждая группа — в каталоге с соответствующим именем;
  - ◊ хешированные имена — каждая статья хранится в отдельном файле, но имена выбираются исходя из ускорения доступа к файлам;
  - ◊ CNFS — все статьи хранятся в кольцевых буферах; есть возможность группировки статей по определенным критериям;

- ❒ `maxforks` — определяет максимально возможное количество одновременно запущенных демонов innd;
- ❒ `maxartsize` — определяет максимально возможный размер статьи;
- ❒ `nicekids` — определяет приоритет процессам, порождаемым программой nnrpd;
- ❒ `nicenewnews` — определяет еще более низкий приоритет программе nnrpd, обрабатывающей команду `newnews`;
- ❒ `mta` — определяет программу, используемую для посылки почтой модерируемых статей;
- ❒ `mailcmd` — определяет программу для посылки отчетов;
- ❒ `logcancelcomm` — определяет, сбрасывать ли в стандартную систему журнализации событий (syslog) сообщения о выполнении команды `cancel`;
- ❒ `wanttrash` — определяет, сохранять ли статьи для несуществующей группы в группе junk;
- ❒ `remembertrash` — определяет, запоминать ли отвергнутые статьи в файле history;
- ❒ `linecountfuzz` — определяет, исправлять ли заголовок `Lines`;
- ❒ `logartsize` — указывает серверу запоминать в журнале размер статьи;
- ❒ `logipaddr` — определяет, записывать ли в журнал событий IP-адрес вместо значения из заголовка `Path`;
- ❒ `logsitename` — определяет, сохранять ли имя хоста в журнале полученных статей;
- ❒ `overviewname` — задает имя файла для хранения истории сообщений; для каждой группы — свой; по умолчанию имя файла — .overview;
- ❒ `extendeddbz` — ускоряет работу с overview за счет увеличения DBZ-файла; требует определенного параметра `storageapi`;

- `nnrpdoverstats` — позволяет сохранять в стандартную систему журнализации событий syslog статистку истории сообщений для nnrpd;
- `storeonxref` — при применении нестандартного метода хранения использовать `Xref:` вместо `Newsgroup:`;
- `nnrpdcheckart` — благодаря этому значению nnrpd будет не только читать overview, но и проверять реальное наличие статьи;
- `storemsgid` — разрешает хранить идентификатор сообщения (Message-ID);
- `usecontrolchan` — позволяет использовать канал для обработки управляющих статей;
- `refusecybercancel` — указывает серверу отвергать статьи, идентификатор сообщения (Message-ID) которых начинается с `cancel`;
- `activedenable`, `activedupdate`, `activedport` — указывают использовать вспомогательный процесс для буферизации доступа nnrpd к файлу active;
- `pathnews`, `pathbin`, `pathfilter`, `pathcontrol`, `pathdb`, `pathetc`, `pathrun`, `pathlog`, `pathhttp`, `pathtmp`, `pathspool`, `patharticles`, `pathoverview`, `pathoutgoing`, `pathincoming`, `patharchive`, `pathuniover` — вышеперечисленные параметры указывают серверу пути к различным составляющим сервера новостей — исполняемым файлам, базам сообщений, журналам событий и т. п.;
- `backoff` — задает ограничение на количество статей, посылаемых локальными клиентами с помощью nnrpd;
- `strippostcc` — указывает nnrpd удалять поля `To:`, `Cc:` и `Bcc:`;
- `nnrpperlauth` — указывает серверу аутентифицировать читателя nnrpd с помощью внешней программы на Perl;
- `pathalias` — указывает, какую строку добавлять перед `pathhost`;
- `nnrpdposthost`, `nnrpdpostport` — программы nnrpd и rnews будут отправлять статьи на заданный сервер;
- `wireformat` — указывает серверу хранить статьи в том же формате, что и при передаче `CR LF` в конце каждой строки и удвоении точки в начале строки;
- `status` — позволяет производить регулярную выдачу статистики на стандартную систему журнализации событий syslog;
- `timer` — позволяет производить регулярный вывод информации о загруженности сервера на стандартную систему журнализации событий syslog;
- `peertimeout` — определяет, сколько секунд входной канал может оставаться неактивным, прежде чем innd его закроет;
- `chaninacttime`, `chanretrytime` — параметры определяют, сколько секунд канал может быть неактивным, прежде чем innd его закроет;

- `maxconnections` — задает число одновременных NNTP-соединений;
- `artcutoff` — задает количество дней для хранения статей: статьи, старше указанного числа дней удаляются;
- `nntplinklog` — разрешает записывать в журнал сообщения nntplink;
- `nntpactsync` — задает, сколько статей обрабатывать между записями в журнал;
- `badiocount` — определяет, сколько ошибок ввода/вывода допускать, не закрывая канал;
- `pauseretrytime` — задает паузу между проверками канала на неактивность;
- `sourceaddress` — определяет, какой адрес будут иметь исходящие пакеты; если указано `any` — будет выбран операционной системой;
- `port` — задает порт, который будет прослушиваться;
- `localmaxartsize` — определяет максимальный размер посылаемых через nnrpd статей;
- `mimeversion` — разрешает nnrpd добавлять MIME-заголовки;
- `mimecontenttype` — если добавляются MIME-заголовки, то здесь определяется значение заголовка `Content-Type`;
- `mimeencoding` — если добавляются MIME-заголовки, то здесь определяется значение заголовка `Content-Transfer-Encoding`;
- `spoolfirst` — если задано `true`, то nnrpd помещает статью от клиента в спул, даже не пытаясь обратиться к innd; если `false` — помещает ее в спул только при получении сообщения об ошибке при посылке;
- `articlemmap` — разрешает использовать mmap при доступе к статье в спуле;
- `clienttimeout` — определяет, сколько секунд клиент nnrpd может не проявлять активность до разрыва соединения;
- `innflags` — задает флаги, передаваемые innd при запуске;
- `doinnwatch` — определяет, запускать ли программу innwatch;
- `innwatchsleeptime` — задает промежуток между проверками innwatch в секундах;
- `controlfailnotice` — определяет, посылать ли администратору письма об ошибках обработки управляющих сообщений;
- `logcycles` — задает, сколько копий старых журналов сохранять;
- `innwatchpauseload` — содержит среднюю загрузку, умноженную на 100, при которой innwatch будет переводить innd в режим ожидания;

- `innwatchhiload` — определяет среднюю загрузку, умноженную на 100, при которой innwatch будет переводить innd в режим throttle;
- `innwatchloload` — средняя загрузка, умноженная на 100, при которой innwatch будет возвращать innd в нормальный режим;
- `innwatchspoolspace` — размер свободного места на устройстве, хранящем articles и overview, в единицах inndf, при достижении которого innwatch переводит innd в режим throttle;
- `innwatchbatchspace` — размер свободного места на устройстве, хранящем исходящие сообщения, в единицах inndf, при достижении которого innwatch переводит innd в режим throttle;
- `innwatchlibspace` — размер свободного места на устройстве, хранящем файлы db-history, active в единицах inndf, при достижении которого innwatch переводит innd в режим throttle;
- `docnfsstat` — определяет, запускать ли cnfsstat; нужен только при использовании метода хранения статей CNFS;
- /etc/news/innfeed.conf — конфигурационный файл для программы innfeed. Более подробную информацию следует искать в документации к серверу новостей;
- /etc/news/innreport.conf — конфигурационный файл для программы innreport. Более подробную информацию следует искать в документации к серверу новостей;
- /etc/news/innwatch.ctl — конфигурационный файл для программы innwatch. Каждая строка определяет одну проверку, состоит из семи полей, разделенных одним символом, и начинается с того же символа. Разделитель полей един для всей строки и выбирается из списка: восклицательный знак, запятая, двоеточие, `@`, точка с запятой или вопросительный знак; в зависимости от того, какой знак из вышеперечисленных не встречается внутри полей в этой строке. Более подробную информацию следует искать в документации к серверу новостей;
- /etc/news/moderators — файл, который хранит имя модерируемой группы и электронный адрес модератора. Когда nnrpd или inews получает статью от клиента и выясняется, что она послана в модерируемую группу, то вместо того, чтобы послать ее innd, он посылает ее по электронной почте модератору этой группы. В данном файле задаются шаблоны для определения адреса модератора по имени группы. Каждая строка состоит из двух полей, разделенных двоеточием. В первом поле указывается шаблон имени группы. Во втором поле указывается электронный адрес модератора конференции;
- /etc/news/news2mail.cf — конфигурационный файл для программы news2mail;

- /etc/news/newsfeeds — файл содержит информацию о том, какие статьи и каким образом необходимо пересылать на соседние NNTP-узлы. Для каждого узла, с которым вы обмениваетесь новостями, должно быть соответствующее описание в этом файле.

  Каждая строка представляет собой отдельное правило, состоящее из 4-х полей, разделенных двоеточиями:

  - `имя_сайта/список_исключений_через_запятую` — первым сайтом в файле должен быть сайт с именем ME. Если он имеет список шаблонов групп, то этот список добавляется в начало списков остальных сайтов:
    - `имя_сайта` получателя записывается в журнал; если имя сайта уже встречается в `Path:`, то статья на него не посылается; для локальных имен (программ обработки типа overchan, archive и т. д.) рекомендуется добавлять восклицательный знак в конце, чтобы не пересечься с реальным именем сайта; в качестве имени сайта получателя обычно выбирается то имя, которое этот сайт вставляет в `Path:` при обработке статьи;
    - `список_исключений` — список имен сайтов через запятую; для каждого имени делается аналогичная проверка — не встречается ли он в `Path:`. Часто используются имена генераторов управляющих сообщений: cyberspam, spewcancel, bincancel;
  - `список_шаблонов_имен_групп_через_запятую/список_областей_распределения_через_запятую`:
    - `список_шаблонов` определяет, какие группы будут посылаться на сайт получателя. Восклицательный знак в начале шаблона означает отрицание. Наибольший приоритет имеет последнее соответствие. Если вместо `!` использовать `@`, то статья из соответствующей группы не будет посылаться на данный сайт, даже если она отсылается в группу, подлежащую посылке;
    - область распространения дополнительно ограничивает список рассылаемых статей — если статья имеет заголовок `Distribution:` и определен список областей распространения для данного сайта получателя, то они должны соответствовать друг другу. Правила записи аналогичны правилам записи шаблонов. Если статья имеет несколько областей распространения, то используется логическое "ИЛИ";
  - `список_флагов`:
    - `<size` — статья посылается, если ее размер меньше указанного числа байтов;
    - `>size` — статья посылается, если ее размер больше указанного числа байтов;

- `Ac` — не посылать управляющие сообщения;
- `AC` — посылать только управляющие сообщения;
- `Ad` — только статьи с заголовком `Distribution:`;
- `Ae` — только если заголовок статьи `Newsgroups:` содержит только те группы, которые имеются в списке активных групп;
- `Ap` — не проверять наличие имени сайта получателя в `Path:` до отсылки сообщения;
- `Fимя_файла` — задает имя файла для спула;
- `Gчисло` — посылать статью, если она послана не более чем в указанное число групп;
- `Hчисло` — посылать статью только если в `Path:` накопилось не более указанного числа хостов;
- `Iразмер` — величина внутреннего буфера, после которого данные начинают сбрасываться в файл;
- `Nm` — только модерируемые группы;
- `Nu` — только немодерируемые группы;
- `Pприоритет` — число от 0 до 20, которое будет назначено программе или каналу;
- `Oшаблон` — требуется наличие заголовка `X-Trace`, и первое поле в нем должно соответствовать шаблону;
- `Sразмер` — если в очереди к данному сайту находится больше указанного размера байтов, то innd переходит в режим спулинга — сбрасывает статью во временный файл;
- `Tтип` — способ передачи статей на сайт:
  - `c` — канал;
  - `f` — файл;
  - `l` — только запись в журнал (очень удобно собирать статистику);
  - `p` — программа;
- `Wполе` — если передача происходит через файл или канал, то здесь указывается, какую информацию туда записывать. Можно использовать несколько флагов. Поля будут записаны в указанном порядке и разделяться пробелами. Программы понимают только поле `*`:
  - `b` — размер статьи в байтах;
  - `f` — полное имя файла статьи;
  - `g` — имя первой группы;
  - `h` — hash-ключ Message-ID;

- m — Message-ID;
- n — имя файла статьи относительно спула;
- p — время посылки статьи;
- s — откуда пришла статья;
- t — время получения статьи;
- * — имена всех сайтов, получающих данную статью;
- D — значение заголовка Distribution: ("?", если не было);
- H — все заголовки;
- N — заголовок Newsgroups:;
- P — заголовок Path:;
- R — данные для репликации.

- параметры — формат зависит от способа посылки статей на сайт.

  Способы посылки статей:

  ◊ журнал — делается только запись в журнале /var/log/news/news;

  ◊ файл — для каждой статьи в файл, определяемый полем Параметры, записывается одна строка. По умолчанию, имя файла — outgoing/имя_сайта;

  ◊ программа — для каждой статьи запускается новый экземпляр программы;

  ◊ канал — в поле Параметры задается полное имя программы, которая запускается при старте innd. На каждую статью запущенный процесс получает одну строку на стандартный ввод. Стандартный вывод, ошибки, UID и GID — как для случая программы. Если процесс уже запущен, он перезапускается. Если процесс не удается запустить, то образуется спул в outgoing/имя_сайта;

  ◊ exploder — особый подтип канала, кроме обычных статей на него могут быть посланы команды. Команда предваряется восклицательным знаком. Автоматически генерируются команды:

  - newgroup имя_группы
  - rmgroup имя_группы
  - flush
  - flush имя_сайта

  ◊ funnel — слияние нескольких потоков в один. Поле Параметр определяет реального получателя;

- /etc/news/nnrp.access — файл определяет права доступа к данному NNTP-узлу. Все строки состоят из пяти полей, разделенных двоеточием и имеют следующий формат:

  `шаблон_хостов:права_доступа:имя_пользователя:пароль:шаблон_имен_групп`

  - `шаблон_хостов` — задает шаблон для сравнения с хостом клиента и может использовать как имена, так и адреса с сетевой маской;
  - `права_доступа` — перечень букв, которые определяют права клиента, зашедшего с соответствующего адреса:
    - `R` — клиент имеет право на чтение;
    - `P` — клиент имеет право на посылку;
    - `N` — клиент может использовать команду `NEWNEWS`, несмотря на глобальный запрет;
    - `L` — клиент может посылать статьи в группы с запретом на локальную посылку;
    - `полное_имя_файла` — формат файла такой же, как и основного, права доступа уточняются, исходя из него;
  - `имя_пользователя` — пустое, если аутентификация клиента не нужна;
  - `пароль` — пустой, если аутентификация клиента не нужна;
  - `шаблон_имен_групп` — список шаблонов имен групп через запятую, к которым клиент должен иметь доступ;
- /etc/news/nnrpd.track — файл позволяет nnrpd записывать в журнал доступа определенную строку текста вместо имени или адреса хоста клиента. Состоит из строк вида:

  `шаблон_имен_или_адресов_хостов:строка_идентифицирующая_пользователя`

- /etc/news/nntpsend.ctl — файл определяет список хостов, на которые nntpsend будет рассылать статьи, если имя хоста не указано явно при запуске. Каждая строка определяет отдельный хост и имеет вид:

  `сайт:fqdn:size:параметры`

  - `сайт` — имя, указанное в newsfeeds;
  - `fqdn` — полное доменное имя хоста, на который должны быть посланы статьи;
  - `size` — размер для обрезания пакета заданий, если он станет слишком большим;
  - `параметры` — параметры для innxmit;
- /etc/news/overview.ctl — файл используется для создания файла истории сообщений overview при использовании новых способов хранения статей;

- /etc/news/overview.fmt — файл определяет, какие заголовки будут храниться в файле истории сообщений overview;
- /etc/news/passwd.nntp — в этом файле хранятся пароли для доступа к NNTP-серверам;
- /etc/news/storage.conf — файл определяет параметры для нестандартных методов хранения статей. Для каждого класса определяется своя структура хранения.

Файл active

Этот файл содержит список групп новостей, которые принимает локальный сервер. Все статьи, опубликованные в группы новостей, которые не указаны в файле active, отвергаются локальным сервером новостей. Строки в этом файле имеют следующий формат:

```
имя    старшая_метка  младшая_метка  флаги
```

где:

- `имя` — имя группы новостей;
- `старшая_метка` — номер самой новой статьи в данной группе новостей на локальном сервере. Это число увеличивается при получении новых статей;
- `младшая_метка` — номер самой старой статьи в данной группе новостей на локальном сервере. Это число изменяется в результате удаления старых статей на диске;
- `флаги` — это поле определяет один из шести возможных флагов:
  - `y` — для данной группы новостей разрешена локальная публикация;
  - `n` — для данной группы новостей не разрешена локальная публикация;
  - `m` — данная группа модерируемая, и все публикации должны быть одобрены модератором;
  - `j` — статьи из данной группы новостей не хранятся на локальном сервере, а только передаются через него;
  - `x` — статьи не могут посылаться в данную группу новостей;
  - `=news.group` — статьи для данной группы новостей помещаются локально в группу news.group.

Основные операции, которые должен время от времени выполнять администратор, включают в себя добавление новых групп, удаление ненужных групп, изменение флагов текущих групп новостей. Все эти операции должны находить свое отображение в файле active.

Существуют два основных подхода к выполнению указанных выше операций с группами новостей.

- Первый подход — использование соответствующих подкоманд команды `ctlinnd` — `newgroup`, `rmgroup` и `changegroup`.
- Второй подход — непосредственное редактирование файла active. Такой подход удобен для операций с большим количеством групп.

Файлы базы данных и журналы

Список файлов базы данных и их стандартное размещение приведено ниже.

- /var/lib/news/.news.daily
- /var/lib/news/active
- /var/lib/news/active.times
- /var/lib/news/distributions
- /var/lib/news/history
- /var/lib/news/newsgroups
- /var/lib/news/subscriptions

Список файлов журналов и их стандартное размещение приведено ниже.

- /var/log/news
- /var/log/news/OLD
- /var/log/news/news.crit
- /var/log/news/news.err
- /var/log/news/news.notice

Сами статьи находятся в следующих файлах:

- /var/spool/news/archive
- /var/spool/news/articles
- /var/spool/news/incoming
- /var/spool/news/incoming/bad
- /var/spool/news/innfeed
- /var/spool/news/outgoing
- /var/spool/news/overview
- /var/spool/news/uniover

Настройка списка получаемых групп новостей

Попробуем выяснить, что нам может предложить провайдер (или любые хосты, которые согласны снабжать нас новостями). Для этого получим список новостей, на которые провайдер подписан. Один из способов получения списка следующий. Воспользуемся командой пакета INN:

```
getlist -h newsserver.our.pro > active.provider
```

Созданный этой командой файл active.provider содержит список групп новостей, на которые подписан наш провайдер. Выберем из списка те группы, на которые мы действительно хотим подписаться, и пропишем их в нашем файле active. Например, если вы хотите подписаться на конференцию relcom.humor, добавьте в этот файл примерно следующее:

```
relcom.humor    0000000000    0000000001    y
```

Если вы хотите принимать все (или почти все) группы новостей, на которые подписан ваш провайдер, то файл active можно получить из active.provider, выполнив для него следующие команды (обнуляются два средних поля каждой строки):

```
#!/bin/sh
sed < active.provider > active \
-e 's/^\([^ ]*\) [0-9]* [0-9]* \([^ ]*\)$/\1 0000000000 0000000000 \2/'
```

Нужный файл active готов (он содержит строки для всех групп, которые поддерживает наш сервер), но надо сообщить и провайдеру о нашем выборе (чтобы он знал, какие группы новостей ему нужно пересылать на наш хост).

Даже если провайдер пропишет нас в своей конфигурации сервера новостей, он не сможет пересылать нам новости по NNTP. Мы должны дать ему разрешение на это. Для этого добавим строчку в файл hosts.nntp:

```
newsserver.our.provider:
```

Здесь надо заметить, что мы полагаемся на провайдера — знаем, что он будет снабжать нас только теми конференциями, о которых мы его попросили. Если же вы не доверяете своим NNTP-соседям, то можно указать конкретно шаблон конференций, которые вы принимаете на локальный диск от конкретного NNTP-соседа. Например, мы хотим принимать от провайдера newsserver.our.badprovider только relcom-группы новостей:

```
newsserver.our.badprovider::relcom.*
```

Отредактируем файл newsfeeds, указав всех NNTP-соседей, которых мы хотим снабжать статьями. Не забудем указать в этом файле своего провайдера. Ниже приведены два примера этого файла.

- В первом случае мы планируем снабжать статьями хост newsserver.our.provider по NNTP:

  ```
  ME:*, !junk, !control*, !local*/!local::
  newsserver.our.provider:*, !junk, !control*, !local*:Tf,
  Wnm:newsserver.our.provider
  ```

- Во втором случае мы хотим снабжать этот же хост по UUCP (имя этой UUCP-системы `provider`), используя программу sendbatch:

  ```
  ME:*, !junk, !control*, !local*/!local::
  provider/newsserver.our.provider:*, !junk, !control*, !local*:Tf, Wnb:
  ```

Затем назначим различные глобальные параметры сервера новостей (имя сервера, имя домена) и параметры, используемые при формировании заголовков статей, публикуемых у нас. Эта информация хранится в файле inn.conf.

Определимся теперь с клиентами нашего сервера новостей (хосты, которые через программу чтения новостей общаются с нашим сервером). Например,

мы хотим ограничить пространство пользования ресурсами нашего сервера новостей своей интранет-сетью (192.168.111.0/255.255.255.0) и нашей внешней сетью (домен our.domain), причем пользователям этих сетей мы разрешаем и читать новости, и публиковать их на нашем сервере. При этом надо помнить о партнерах из домена partner.domain (правда, им нечего делать в наших локальных конференциях). Ну, а для остальных поместим первым правило, запрещающее любой доступ. Для этого добавим в файл nnrp.access строки:

```
*:: -no- : -no- :!*
192.168.111.*:Read Post:::*
*.our.domain:Read Post:::*
*.partner.domain:Read Post:::*, !local*
```

Как только мы начнем получать статьи на локальный диск, надо будет следить за сроком их хранения на диске и удалять старые (диск же не резиновый). К счастью, за нас это будет делать программа expire, а от нас требуется только дать ей соответствующие указания в файле expire.ctl (ну и конечно, запускать механизм очистки). В этом файле следует указать:

- срок хранения идентификаторов статей в файле history (это делается для того, чтобы не принимать заново удаленные статьи);
- срок хранения самих тел статей.

Пример ниже показывает, что запись об идентификаторе статей хранится в файле history 14 дней после удаления тела этих статей, тела статей из локальных телеконференций хранятся на системе от 5 до 7 дней (по умолчанию 6), а для всех остальных телеконференций тела хранятся от 3 до 5 дней (по умолчанию — 4 дня):

```
/remember/:14
*:A:3:4:5
local*:A:5:6:7
```

Заметим, что значение по умолчанию (образец `*`) должно фигурировать раньше, чем строки для отдельных групп, поскольку применяется последнее соответствие образцу в первом поле.

Важным шагом после редактирования конфигурационных файлов является проверка корректности сделанных нами изменений. Система INN имеет ряд средств, помогающих нам в решении этой задачи. Вот некоторые из них.

- Для поиска ошибок в файле newsfeeds можно дать следующую команду:

  ```
  innd -s
  ```

  Например, если вы получили в ответ следующее:

  ```
  Found 1 errors --see syslog
  ```

  то это значит, что командой обнаружена одна ошибка, о которой сообщается через syslog в файлах news.err и news.notice.

- Для проверки файла active на наличие неверных строк можно дать следующую команду:

  ```
  expire -n -x -t
  ```

  Например, если в ответ получено следующее:

  ```
  /var/news/etc/active: line 5 wrong number of fields
  ```

  то это значит, что вы ошиблись с количеством полей в 5-й строке этого файла (их должно быть 4). Однако это не лучший способ проверки файла active. В частности, expire не замечает отсутствие флага для группы новостей (в отличие от inncheck).

Итак, обратим внимание на inncheck — Perl-сценарий, предназначенный для проверки всех рассматриваемых нами конфигурационных файлов. Помимо проверки файлов на наличие синтаксических ошибок, он может осуществлять проверку прав доступа к файлам их владельцев. Возвращаясь к примеру выше (отсутствие флага в конце строки файла active), inncheck сообщит вам об этой ошибке:

```
/var/news/etc/active:5: ends with whitespace
```

Запущенный без параметров, inncheck проверит синтаксис всех файлов (которые может проверить), с выводом на экран сообщений об ошибках. Если мы укажем опцию `-v` (режим verbose), то inncheck расскажет нам о том, что он просматривает. Мы можем ограничить работу inncheck проверкой синтаксиса конкретного файла, дав команду `inncheck имя_файла`. Для того чтобы проверить корректность прав доступа к файлам и корректность владельцев и групп файлов, можно дать команду `inncheck -perm`. Ту же информацию, да еще и с указанием того, какие команды надо выполнить, чтобы устранить ошибки, дает команда `inncheck -f -perm`.

Последний шаг настройки — периодически запускать программу отправки статей с нашей машины, программу чистки каталога статей и обобщения log-файлов. Для этого отредактируем таблицу заданий пользователя news для демона cron:

```
crontab -u news -e
```

Ваш редактор (определенный переменной окружения `EDITOR`) откроет файл /var/cron/tabs/news. Ежедневно в 4 часа утра мы будем запускать сценарий news.daily, в функции которого входит обобщение и ротация файлов регистрации, прогон программы expire и др. Далее, в 1-ую минуту и 28-ую минуту каждого часа мы будем запускать программу nntpsend для отправки потоков статей по NNTP нашим соседям.

```
0    4  *  *  *   /usr/news/bin/news.daily > /dev/null 2>&1 &
1, 28 *  *  *  *  /usr/news/bin/nntpsend > /dev/null 2>&1 &
```

Наконец, если мы планируем отправлять потоки новостей по UUCP на UUCP-систему provider, то в 37 минут каждого часа из cron будем вызывать программу sendbatch:

```
37  *  *  *  *  /usr/news/bin/sendbatch -c provider > /dev/null 2>&1 &
```

Ну что ж, теперь можно запустить демон innd (rc.news поможет нам в этом) и насладиться его работой!

Журналирование пакета INN

Пакет INN использует стандартный способ — стандартную систему журнализации событий syslog. Помимо этого, можно использовать дополнительные журналы сообщений, в частности:

- news.crit — содержит сообщения о критических ошибках, требующих внимания от администратора сервера новостей;
- news.err — содержит сообщения о фатальных ошибках сервера;
- news.notice — используется для записи информации о соединении удаленных NNTP-хостов, активности клиентов, в этом же файле информируют о своей работе программы ctlinnd, innxmit, rnews.

Система INN имеет помимо log-файлов, поддерживаемых системой syslog, встроенные log-файлы — errlog и news (по умолчанию они расположены в каталоге /var/log/news):

- файл errlog содержит стандартный вывод и стандартные ошибки любых программ, порождаемых демоном innd;
- файл news регистрирует все статьи, поступающие к innd для обработки.

Помимо перечисленных выше файлов регистрации, ряд программ системы INN ведет собственные файлы регистрации (expire.log, send-uucp.log, nntpsend.log и др.).

Программы пакета INN

Поскольку пакет INN очень велик, то в этом разделе приведены некоторые программы, имеющие отношение к пакету с небольшими комментариями:

- /usr/bin/actived — вспомогательный демон для nnrpd, хранит в памяти проиндексированный файл active;
- /usr/bin/actmerge — утилита, позволяющая произвести слияние двух файлов active;
- /usr/bin/actsync — утилита для синхронизации, сравнения или слияния файлов active;
- /usr/bin/archive — утилита для создания архивной копии части статей;

- /usr/bin/batcher — программа разбивает на пакеты указанного размера список статей, подготовленных для посылки на хост;
- /usr/bin/controlchan — программа позволяет передать обработку управляющих сообщений из innd внешней программе;
- /usr/bin/convdate — утилита для преобразования формата времени;
- /usr/bin/ctlinnd — интерфейс для управления работающим innd;
- /usr/bin/cvtbatch — преобразует Usenet-пакеты в формат INN;
- /usr/bin/expire — утилита для удаления старых статей без прерывания работы innd;
- /usr/bin/expireindex — удаление старых статей из списка заголовков статей группы;
- /usr/bin/expireover — удаление старых статей из списка статей группы;
- /usr/bin/fastrm — быстрое удаление группы файлов;
- /usr/bin/getlist — получение списков от NNTP-сервера;
- /usr/bin/grephistory — быстрое извлечение статьи по ее индексу;
- /usr/bin/inncheck — проверка конфигурационных файлов;
- /usr/bin/innd — основной сервер, принимающий данные и изменяющий базу данных;
- /usr/bin/inndstart — пусковая программа для innd;
- /usr/bin/innreport — обработка журналов;
- /usr/bin/innstat — выдать состояние сервера;
- /usr/bin/innwatch — мониторинг сервера inn;
- /usr/bin/innxbatch — послать статьи в формате Usenet другому NNTP-серверу;
- /usr/bin/innxmit — пересылка пакета статей другому NNTP-серверу;
- /usr/bin/mailpost — поместить письмо в news-группу;
- /usr/bin/makeactive — восстановление файла active по спулу;
- /usr/bin/news.daily — подготовка ежедневного отчета;
- /usr/bin/news2mail — превращение статей в письма;
- /usr/bin/nnrpd — отдельный процесс, предоставляющий клиентам доступ к статьям;
- /usr/bin/nntpsend — оболочка для innxmit;
- /usr/bin/overchan — заполнение данных списка заголовков статей группы;
- /usr/bin/parsecontrol — анализ управляющих сообщений;
- /usr/bin/pgpverify — проверка управляющих сообщений;

- /usr/bin/scanlogs — обработка журналов;
- /usr/bin/send-nntp — подготовка и рассылка пакетов с помощью innxmit;
- /usr/bin/sendxbatches — подготовка и рассылка пакетов с помощью innxbatch;
- /usr/bin/writelog — запись в журнал inn.

Утилиты

newsprune

Утилита просматривает все каталоги, соответствующие файлу active, и генерирует список файлов, для которых нет соответствующей строки в индексном файле.

findmissing.pl

Создает список файлов, найденных в спуле, но отсутствующих в индексном файле.

Ссылки

- **malik.bishkek.su/doc/UNIX/innd/inn.htm** — Юрий Савин. Сервер новостей InterNetNews (INN).
- **www.bog.pp.ru/work/inn.html** — конфигурирование сервера INN.
- **www.isc.org/products/INN** — официальный сайт INN.
- **www.switch.ch/switch/netnews/wg/newstools.html** — утилиты для пакета INN.
- **www.mibsoftware.com/userkt/inn/0346.htm** — утилиты для пакета INN.

Глава 22

Proxy-сервер

При подключении к любому провайдеру вам выдаются параметры настройки — адрес сервера DNS, адрес почтового сервера и сервера новостей, а так же — адрес proxy-сервера.

Что собой представляет proxy-сервер? Если вы настроите свой браузер для работы через proxy-сервер, то при запросе некоторого документа из Интернета, если некоторое время назад кто-то уже обращался с подобным запросом, вы получите документ незамедлительно, с максимальной скоростью, на которую способно ваше сетевое подключение, потому что вы получите копию документа, взятую из кэша proxy-сервера. Если же proxy-сервер не имеет в своем кэше данного документа, то proxy-сервер запросит удаленный WWW-сервер, хранящий оригинал, и выдаст документ вам, одновременно положив копию документа в свой кэш.

Чем больше пользователей пользуются proxy-сервером, тем более существенной становится его помощь. Согласно статистике, количество обращений пользователей к одним и тем же документам в сети Интернет приближается к 60%.

Многие proxy-серверы обладают еще одним интересным свойством — они могут обмениваться информацией с соседними proxy-серверами, что существенно ускоряет доступ к данным, хранящимся на удаленных или сильно загруженных серверах.

Proxy-сервер предоставляет следующие возможности:

- ❒ централизованный выход в Интернет через один сервер в сети;
- ❒ локальное хранение часто просматриваемых документов для увеличения скорости загрузки страниц;
- ❒ возможность регулировать пропускную способность канала в зависимости от его нагрузки;
- ❒ авторизованный доступ в Интернет;
- ❒ возможность обмена данными кэша с соседними proxy-серверами.

Однако не все данные могут быть корректно получены через proxy-серверы. Это касается, прежде всего, динамически формируемой информации. Однако большинство современных proxy-серверов имеют большое количество настроек и обладают множеством интеллектуальных алгоритмов, позволяющих в большинстве случаев корректно получать самую свежую информацию.

Наиболее распространенным proxy-сервером, доступным под лицензией GNU, является Squid.

Squid

Squid это высокопроизводительный кэширующий proxy-сервер, поддерживающий протоколы FTP, gopher и HTTP. Squid сохраняет часто запрашиваемые данные в оперативной памяти компьютера, что позволяет резко увеличить производительность proxy-сервера, кэширует DNS-запросы (это свойство интересно тем, кто не имеет своего DNS-сервера). Помимо вышеперечисленных возможностей, поддерживает SSL, расширенный контроль доступа и полную регистрацию запросов.

Одной из ключевых возможностей пакета Squid является использование протокола Internet Cache Protocol (ICP, Протокол интернет-кэширования), что позволяет создать иерархию proxy-серверов Squid для дополнительной экономии пропускной способности канала.

Поддерживаемые функции Squid:

- proxy и кэширование HTTP, FTP;
- proxy для SSL;
- иерархия кэшей;
- ICP, HTCP, CARP, Cache digests;
- прозрачный proxy;
- WCCP;
- гибкий контроль доступа;
- HTTP-серверное ускорение;
- SNMP;
- кэширование DNS-запросов;
- возможность ограничения трафика.

Рассмотрим некоторые из этих функций подробнее.

Протокол ICP

Протокол ICP используется в иерархии кэшей для поиска объектов в дереве кэшей Squid-серверов. Если ваш Squid не находит нужного документа, то

посылает ICP-запрос другим Squid-серверам, входящим в вашу иерархию proxy-серверов. Эти серверы отвечают ICP ответами `HIT` (попадание) или `MISS` (промах). После получения ответов ваш сервер решает, при помощи какого кэша proxy-сервера получить необходимые ему данные.

Cache digest

Компактная форма представления списка содержимого кэша proxy-сервера. Proxy-серверы могут обмениваться этой информацией с соседями для избежания необходимости делать ICP-запросы (экономия трафика). В качестве ключей объектов используется протокол шифрования MD5.

Иерархия кэшей

Иерархия кэшей — это структура кэширующих proxy-серверов, расположенных логически как родительский/дочерний и братский узлы таким образом, что кэши, ближайшие к интернет-каналу, являются родителями тем proxy-серверам, которые находятся дальше от точки доступа к Интернету. В случае, когда кэш запрашивает объект от родителя, и у того в кэше необходимый объект отсутствует, родительский proxy-сервер получает объект из Интернета, кэширует его и передает дочернему. Таким образом, при помощи иерархии достигается максимальная разгрузка канала.

Кроме родительских/дочерних отношений, Squid поддерживает понятие братских кэшей — находящихся на одном уровне иерархии. Каждый proxy-сервер в иерархии независимо ни от кого решает, откуда получать необходимый объект — напрямую из Интернета, от родительского или братского кэша.

Алгоритм получения запрошенного объекта пакетом Squid

Алгоритм таков:

1. Разослать ICP-запросы всем братским кэшам.
2. Дождаться всех ответов, пришедших в течение заданного времени:
   - получив первый ответ `HIT` (попадание), получить объект;
   - или взять объект от первого родительского кэша, ответившего `MISS` (зависит от настройки);
   - или получить объект из Интернета.

Конфигурирование пакета Squid

Основное место конфигурирования пакета Squid — файл /etc/Squid.conf. Размер этого файла достаточно велик, поскольку он содержит множество

конфигурируемых параметров, начиная с номера порта для ICP-запросов и заканчивая правилами доступа к информации. Далее приведены параметры конфигурации Squid-сервера, разбитые на типы. Однако приведенный список не является полным и исчерпывающим, поскольку он содержит только наиболее интересные (с нашей точки зрения) параметры конфигурации.

Сетевые параметры

Сетевые параметры proxy-сервера имеют следующие настройки.

- Порт для запросов клиентов proxy-сервера:

```
http_port 3128
```

- Порт для ICP-запросов. В том случае, если не предполагается использовать иерархию proxy-серверов — необходимо указать нулевой порт:

```
icp_port 3130
```

- Порт для общения с соседями ICP — через TCP-протокол:

```
htcp_port 4827
```

- К каким multicast-группам (соседи-серверы squid) подсоединяться для получения ICP, если используется multicast:

```
mcast_groups 239.128.16.128 224.0.1.20
```

- По умолчанию режим пассивного FTP включен, но если Squid находится за брандмауэром, то необходимо выключить:

```
passive_ftp on | off
```

Соседи

Как уже упоминалось ранее, Squid может обмениваться информацией с другими squid-серверами, которых принято называть соседями.

- Каждый сосед описывается отдельной строкой:

```
cache_peer hostname type proxy-port icp-port options
```

  - параметр `type` имеет следующие значения:
    - `parent` — старший в иерархии;
    - `sibling` — одного уровня.
  - параметр `options` имеет следующие значения:
    - `proxy-only` — объекты, взятые с указанного узла, не хранить у себя в кэше;
    - `weight=число` — указывает приоритет хоста, чем значение больше, тем больше приоритет;

- ◊ `ttl=число` — время жизни пакета используется при настройке multicast;
- ◊ `no-query` — не посылать ICP-запросы;
- ◊ `default` — самый старший в иерархии;
- ◊ `round-robin` — определяет родительские кэши, используемые по очереди;
- ◊ `multicast-responder` — данный сосед является членом multicast-группы;
- ◊ `no-digest` — не запрашивать от этого соседа cashe digest;
- ◊ `login=user:password` — определение имени и пароля для случая, если старший в иерархии proxy-сервер требует аутентификации;
- ◊ `connect-timeout=число` — время ожидания ответа от соседей;

- ❒ `cache_peer_domain host domain [domain...]` — ограничить запросы к данному соседу данным списком доменов;
- ❒ `icp_query_timeout milisec` — время ожидания ответа в миллисекундах;
- ❒ `mcast_icp_query_timeout milisec` — ожидание ответа на регулярные multicast-опросы;
- ❒ `dead_peer_timeout seconds` — время ожидания ответа от соседа, по истечении которого считается, что сосед отсутствует в сети;
- ❒ `hierarchy_stoplist` — список строк (через пробел), при встрече которых в URL, запрос не будет кэшироваться; по умолчанию cgi-bin;
- ❒ `no_cache deny имя-ACL` — определяет список объектов, которые не будут кэшироваться.

Размер кэша

Раздел предназначен для определения параметров кэша — размера, использования, времени хранения информации и т. п.

- ❒ `cache_mem 8 MB` — объем оперативной памяти, используемой для хранения обрабатываемых объектов;
- ❒ `cache_swap_high 95` — при достижении данного уровня заполнения кэша (в процентах) начинается ускоренный процесс очистки кэша от устаревших объектов;
- ❒ `cache_swap_low 90` — процесс удаления старых объектов заканчивается, если достигнут данный уровень (в процентах);
- ❒ `maximum_object_size 4096 KB` — максимальный размер кэшируемого объекта;
- ❒ `minimum_object_size 0 KB` — минимальный размер кэшируемого объекта; файлы меньшего размера не сохраняются;

- `ipcache_size 1024` — размер кэша для IP-адресов;
- `ipcache_high 95` — верхний уровень заполнения IP-кэша для алгоритма удаления старых объектов;
- `ipcache_low 90` — нижний уровень заполнения IP-кэша для алгоритма удаления старых объектов.

Имена и размеры файлов

В этом разделе определяются имена и размеры используемых файлов:

- `cache_dir тип Directory-Name Mbytes Level-1 Level2` — определяет имя, размер и количество подкаталогов на первом и втором уровне кэша на диске — каждый кэшируемый объект кладется в отдельный файл, файлы хранятся в двухуровневой иерархии каталогов;
- `cache_access_log /usr/local/squid/logs/access.log` — место хранения журнала обращений к кэшу;
- `cache_log /usr/local/squid/logs/cache.log` — место хранения журнала запусков процессов;
- `cache_store_log /usr/local/squid/logs/store.log` — место хранения журнала записи объектов в дисковый кэш;
- `emulate_httpd_log on|off` — производить ли эмуляцию формата журнала HTTPD;
- `mime_table /usr/local/squid/etc/mime.conf` — таблица типов MIME;
- `log_mime_hdrs off` — в журнал access записываются полученные HTTP-заголовки;
- `useragent_log имя-файла` — в этот файл будут записываться строки User-agent из HTTP-заголовков;
- `debug_options раздел, уровень` — уровень отладки; `ALL` — для всех разделов; по умолчанию `ALL,1`;
- `log_fqdn off` — позволяет определять и записывать в журнал полные доменные имена источника запроса.

Параметры внешних программ

Как и большинство серьезных программ, Squid позволяет воспользоваться внешними программами для выполнения некоторых действий. К примеру — сбор статистики или обработка трафика.

- `ftp_user email-адрес` — будет подставляться вместо пароля при анонимном доступе к FTP-серверам; по умолчанию — `Squid@`, вызывает проблемы с серверами, которые проверяют синтаксис адреса;

- `cache_dns_program /usr/local/squid/bin/dnsserver` — местоположение программы, кэширующей DNS-запросы;
- `dns_children 5` — число процессов, которые делают DNS lookup (получение по IP-адресу доменного имени и наоборот);
- `dns_nameservers список-IP-адресов` — используется вместо списка DNS-серверов, определенного в /etc/resolv.conf;
- `redirect_program none` — позволяет подключить программу преобразования URL при каждом запросе;
- `redirect_children 5` — параметр определяет, сколько процессов преобразования URL запускать параллельно;
- `redirect_rewrites_host_header on` — разрешает или запрещает изменение поля `Host:` в заголовке запроса; по умолчанию Squid переписывает поле `Host:` в заголовках преобразованных запросов;
- `redirector_access acl` — какие запросы направлять через редиректор; по умолчанию — все;
- `authenticate_program none` — позволяет производить аутентификацию клиентов, делающих запросы; программа должна в цикле читать строку "имя пароль" выдавать `OK` или `ERR`; должен быть определен параметр `ACL proxy_auth`;
- `authenticate_children 5` — сколько параллельных процессов будут заниматься аутентификацией;
- `authenticate_ttl 3600` — сколько секунд кэшировать результаты работ программы аутентификации;
- `authenticate_ip_ttl число` — необходимо установить 0, чтобы с нескольких адресов не смогли воспользоваться одним именем.

Тонкая настройка кэша

С помощью следующих параметров можно произвести тонкую настройку параметров кэша:

- `wais_relay_host localhost` — куда перенаправлять WAIS-запросы;
- `wais_relay_port 8000` — куда перенаправлять WAIS-запросы;
- `request_header_max_size 10KB` — максимальный размер заголовка;
- `request_body_max_size 1 MB` — максимальный размер объекта;
- `refresh_pattern [-i] regex MIN_AGE percent MAX_AGE[options]` — используется для определения, не устарел ли объект в кэше.

  Имя объекта сравнивается по очереди с регулярными выражениями в строках `refresh_pattern` до первого совпадения, параметры из соответствующей строки используются в алгоритме проверки. По умолчанию регу-

лярные выражения различают прописные/строчные буквы, чтобы игнорировать это различие, используется ключ `-i`. `MIN_AGE` и `MAX_AGE` — время жизни объекта в минутах. По умолчанию:

- `refresh_pattern ^ftp: 1440 20% 10080`
- `refresh_pattern ^gopher: 1440 0% 1440`
- `refresh_pattern. 0 20% 4320`

Более подробную информацию смотрите в документации на Squid;

- `reference_age 1 month` — максимальное время хранения неиспользуемого объекта до его удаления;
- `quick_abort_min 16 KB` — если клиент оборвал запрос, а осталось докачать всего `min KB`, то Squid произведет докачку объекта;
- `quick_abort_max 16 KB` — если клиент оборвал запрос и осталось качать больше `max KB`, то Squid прекратит получение объекта;
- `quick_abort_pct число` — если клиент оборвал запрос и уже получено больше чем `число` процентов объекта, то Squid докачает объект;
- `negative_ttl 5 minutes` — время кэширования негативных ответов (например "connection refused ", "404 not found") — число задает их время жизни в кэше;
- `positive_dns_ttl 6 hours` — время кэширования положительных DNS-ответов — число задает их время жизни в кэше;
- `negative_dns_ttl 5 minutes` — время кэширования негативных DNS-ответов — число задает их время жизни в кэше;
- `range_offset_limit 0 KB` — если клиент делает запрос с середины объекта, то:
  - `1` — вынуждает Squid загрузить весь объект в кэш до того, как начать передачу клиенту;
  - `0` — означает, что Squid никогда не будет грузить больше, чем клиент запросил;
  - число, отличное от `1` — начало запроса меньше этого числа — Squid будет грузить весь объект.

Время ожидания

В этом разделе задаются различные временные параметры Squid:

- `connect_timeout 120 seconds` — время ожидния соединения с сервером;
- `siteselect_timeout 4 seconds` — максимальное время на выбор URL;
- `read_timeout 15 minutes` — сколько времени разрешается ждать следующего байта от сервера;

- `request_timeout 30 seconds` — сколько разрешается ждать запроса после установления соединения;
- `client_lifetime 1 day` — сколько времени разрешать клиенту быть присоединенным к Squid; соединение обрывается, даже если происходит передача данных;
- `half_closed_clients on` — разрешать наполовину закрытые соединения — например чтение есть, а запись уже закрыта;
- `shutdown_lifetime 30 seconds` — сколько времени продолжать обслуживание после получения сигнала `SIGTERM` или `SIGHUP`.

ACL — Access Control List

Этот раздел определяет правила доступа пользователей к группам файлов и хостов. С помощью ACL можно очень гибко настроить доступ к различным сайтам.

`acl имя тип строка` — определение списка доступа (`имя` — имя правила, `тип` — тип объекта, `строка` — регулярное выражение (шаблон для сравнения), по умолчанию чувствительное к регистру букв.

Параметр `тип` может принимать следующие значения:

- IP-адреса клиентов:

  ```
  src ip-address/netmask...
  ```

- диапазон адресов:

  ```
  src addr1-addr2/netmask...
  ```

- получение IP-адреса по URL:

  ```
  srcdomain foo.com...
  ```

- если в URL использовался IP, то делается попытка определить имя домена, если не удалась, то подставляется слово `none`:

  ```
  dstdomaiвn foo.com...
  ```

- получение IP-адреса клиента по URL с использованием регулярных выражений:

  ```
  srcdom_regex [-i] строка...
  ```

- если в URL использовался IP, то делается попытка определить имя домена используя регулярные выражения:

  ```
  dstdom_regex [-i] строка...
  ```

- регулярное выражение для всего URL:

  ```
  url_regex [-i] строка
  ```

- регулярное выражение для path-части URL:

```
urlpath_regex [-i] строка
```

- определяются безопасные порты:

```
port порт...
```

- сопоставляется заголовок User-Agent:

```
browser [-i] regexp
```

- ограничивает число соединеной с одного и того же IP:

```
maxconn число
```

Права доступа

Права доступа определяются следующими строками:

- `http_access allow|deny [!]aclname...` — кому разрешать доступ к proxy по HTTP;
- `icp_access allow|deny [!]aclname...` — кому разрешать доступ к proxy по ICP;
- `miss_access allow|deny [!]aclname...` — кому разрешить получать ответ MISS;
- `cache_peer_access cache-host allow|deny [!]aclname...` — ограничить запросы к данному соседу;
- `proxy_auth_realm Squid proxy-caching web server` — строка текста, которая будет выдана на экран клиента при запросе имени/пароля доступа к кэшу.

Параметры администрирования

Параметры администрирования определяются следующими строками:

- `cache_mgr email` — почтовый адрес, на который будет послано письмо, если у Squid возникнут проблемы;
- `cache_effective_user nobody` — если запускается Squid от имени root, то заменить UID на указанный;
- `cache_effective_group nogroup` — если запускается Squid от группы root, то заменить GID на указанный;
- `visible_hostname имя-хоста` — это имя будет упоминаться в сообщениях об ошибках;
- `unique_hostname уникальное-имя` — если нескольким кэшам дали одно и то же `visible_hostname`, необходимо определить каждому из них уникальное имя;
- `hostname_aliases имя...` — список синонимов для имени хоста.

Параметры для работы в режиме ускорителя HTTP-сервера

Параметры для работы в режиме ускорителя HTTP-сервера определяются следующими строками:

- `httpd_accel_host hostname` — если нужна поддержка виртуальных хостов, в частности для transparent proxy (прозрачное кэширование), то вместо имени указать `virtual`;
- `httpd_accel_port port` — порт для HTTP-сервера;
- `httpd_accel_with_proxy on|off` — кэширование для ускоряемого сервера;
- `httpd_accel_uses_host_header on|off` — для работы в прозрачном режиме требуется включить, иначе виртуальные серверы не будут правильно кэшироваться.

Разное

Здесь содержатся параметры Squid, не вошедшие в предыдущие разделы:

- `dns_testnames netscape.com internic.net microsoft.com` — список имен хостов, на примере которых проверяется работоспособность DNS;
- `logfile_rotate 10` — данный параметр задает количество старых копий при ротации;
- `append_domain.vasya.ru` — добавляется к имени хоста, если в нем нет ни одной точки;
- `tcp_recv_bufsize 0 bytes` — `0` означает, что надо использовть размер буфера по умолчанию;
- `err_html_text строка` — подставляется в шаблоны текстов сообщений об ошибках;
- `deny_info err_page_name acl` — запросы, не прошедшие проверку в http_access, проверяются на соответствие ACL, выдается соответствующее сообщение об ошибке из файла page_name;
- `memory_pools on|off` — эта переменная определяет политику использования захваченной памяти:
  - `on` — однажды захваченная, но не используемая память не отдается обратно в систему;
  - `off` — позволяет освобождать захваченную память;
- `memory_pools_limit байт` — максимальное количество неиспользуемой памяти, которое Squid будет удерживать, если `0` — то удерживать все, что было захвачено;

- `forwarded_for on|off` — если включено, то Squid будет вставлять IP-адрес или имя в заголовки перенаправляемых HTTP-запросов: `X-Forwarded-For: 192.1.2.3`; если выключено, то `X-Forwarded-For: unknown`;
- `log_icp_queries on|off` — записываются ли в журнал ICP-запросы;
- `icp_hit_stale on|off` — возвращать ли ответ `ICP_HIT` для устаревших объектов;
- `cachemgr_passwd password action action...` — задание пароля для действий по администрированию Squid; чтобы запретить действие — поставьте пароль `disable`; чтоб разрешить действие без проверки пароля — поставьте пароль `none`, кроме действий config и shutdown; полную информацию смотрите в документации на Squid;
- `store_avg_object_size 13 KB` — предполагаемый средний размер объекта, используемый для расчетов;
- `store_objects_per_bucket 20` — число объектов на хэш-корзину;
- `client_db on|off` — сбор статистики о клиентах;
- `netdb_low 900` — нижняя граница для базы данных измерения ICMP;
- `netdb_high 1000` — верхняя граница для базы данных измерения ICMP;
- `netdb_ping_period 5 minutes` — минимальное время между посылок ping-пакетов в одну и ту же сеть;
- `query_icmp on|off` — должны ли соседи в ICP-ответы включать ICMP-данные;
- `test_reachability on|off` — если включить, то ответ `ICP_MISS` будет заменяться на `ICP_MISS_NOFETCH`, если сервер отсутствует в базе данных ICMP или RTT равен нулю;
- `buffered_logs on|off` — при включении запись в журнал буферизуется;
- `always_direct allow|deny [!]aclname...` — запросы, удовлетворяющие данным ACL, не кэшировать, а всегда направлять к первоисточнику;
- `never_direct allow|deny [!]aclname...` — запросы, удовлетворяющие данным ACL, всегда кэшировать;
- `anonymize_headers allow|deny header_name...` — перечень заголовков, которые нуждаются в анонимизации;
- `fake_user_agent none` — если заголовок `User-Agent` фильтруется с помощью анонимизатора, то подставляется эта строка;
- `minimum_retry_timeout 5 seconds` — если сервер имеет несколько IP-адресов, то тайм-аут соединения делится на их количество;
- `maximum_single_addr_tries 3` — сколько раз пытаться соединиться с сервером, имеющим один IP-адрес; если сервер имеет несколько IP-адресов, то каждый из них будет опробован один раз;

- `snmp_port 3401` — порт, который слушает Squid для SNMP-запросов;
- `snmp_access allow|deny [!]aclname...` — определяет, кто будет допущен к SNMP-порту;
- `offline_mode on|off` — если включить, то Squid будет брать объекты только из кэша и не будет пытаться обращаться к первоисточникам;
- `uri_whitespace strip` — что делать с запросами, имеющими пробелы в URI; возможные варианты:
  - `strip` — удалять пробелы;
  - `deny` — сообщать Invalid Request (ошибочный запрос);
  - `allow` — передавать как есть;
  - `encode` — кодировать в соответствии с RFC1738, передавать дальше;
  - `chop` — остаток после первого же пробела отбрасывать;
- `mcast_miss_addr адрес` — по этому multicast-адресу посылается сообщение при каждом "непопадании" в кэш;
- `mcast_miss_port порт` — этот порт используется для посылки сообщения;
- `strip_query_terms on` — удалять параметры запроса перед записью в журнал;
- `ignore_unknown_nameservers on` — игнорировать сообщения от DNS-серверов, с которыми Squid не работает.

Пример конфигурации Squid

Как вы уже заметили, опций для конфигурации Squid очень много. Для быстрой настройки proxy-сервера можно воспользоваться приведенными ниже параметрами. Конечно они не являются идеальными, наверняка тонкая настройка поможет вам оптимизировать сервер как с точки зрения увеличения производительности, так и с точки зрения безопасности.

Возьмем стандартный файл Squid.conf и отредактируем только нижеприведенные строки:

- `http_port 3128` — номер порта, на котором Squid будет слушать команды от клиентов;
- `hierarchy_stoplist cgi-bin, chat` — слова в URL, при обнаружении которых proxy-сервер будет не кэшировать объекты, а напрямую перенаправлять запрос серверу;
- `cache_mem 16 MB` — сколько оперативной памяти Squid может забрать под свои нужды. Чем больше выделить памяти — тем быстрее будут обрабатываться запросы. Весьма зависит от количества клиентов;

- `maximum_object_size 16384 KB` — максимальный размер объектов, которые будут сохранены в кэше. Размер специфичен для ваших задач и объема жесткого диска;
- `cache_dir /usr/local/Squid/cache 2048 16 256` — указывает proxy-серверу, где сохранять кэшируемые файлы. Под кэш выделяется два гигабайта и создается 16 и 256 каталогов 1-го и 2-го уровня;
- `ftp_user anonymous@vasya.ru` — задает proxy-серверу, под каким паролем регистрироваться на анонимных FTP-серверах;
- `negative_ttl 1 minutes` — время жизни страничек с ошибкой;
- `positive_dns_ttl 6 hours` — время жизни удачного преобразования DNS-имен в IP-адреса;
- `negative_dns_ttl 5 minutes` — время жизни соответственно удачного и неудачного преобразования DNS-имен в IP-адреса.

Дальнейшие наши действия касаются разграничения прав пользователей.

Сначала необходимо определить ACL (Access Control List, список управления доступом). Сначала закомментируем все строчки в файле Squid.conf, начинающиеся на `acl`. Затем пишем свои правила. К примеру:

- `acl users proxy_auth vasya tolik petya nina` — этой строчкой мы указываем proxy-серверу правило, по которому разрешаем пускать вышеперечисленных пользователей с использованием авторизующей программы через Squid;
- `acl BANNER url_regex banner reklama linkexch banpics us\.yimg\.com [\./]ad[s]?[\./]` — это правило определяет адреса, содержащие рекламу. Интересна для тех, кто хочет отказаться от получения разнообразных баннеров. Позволяет экономить сетевой трафик;
- `http_access deny !users` — эта строка запрещает доступ всем пользователям, кроме тех, которые перечислены в группе `users`;
- `http_access deny BANNER` — запрещаем доступ к URL, удовлетворяющим правилу `BANNER` (убираем рекламу);
- `proxy_auth_realm Vasy Pupkina proxy-caching web server` — строка, которая выводится в окно с логином/паролем;
- `cache_mgr vasya@pupkin.ru` — если у клиента возникает проблема — выводится HTML-страница с сообщением и адресом электронной почты администратора, в нашем случае **vasya@pupkin.ru**;
- `cache_effective_user nobody` — с правами какого пользователя выполняется proxy-сервер;
- `cache_effective_group nogroup` — с правами какой группы выполняется proxy-сервер;
- `client_db on` — параметр разрешает собирать статистику по клиентам.

Поскольку стандартной настройки в такой сфере, как использование канала, места на винчестере, оперативной памяти просто не может быть, более тонкие настройки и ограничения вы должны обдумать и настроить сами.

Создание иерархии proxy-серверов

Чтобы разместить кэш в иерархии, нужно воспользоваться директивой `cache_host`.

Приведенной ниже частью конфигурационного файла Squid.conf сервер pupkin.ru сконфигурирован так, что его кэш получает данные с одного родительского и с двух братских кэшей:

```
cache_host petya.com parent 3128 3130
cache_host monya.ru sibling 3128 3130
cache_host gesha.ru sibling 3128 3130
```

Директива `cache_host_domain` позволяет задавать для каждого определенного домена или группы доменов как братский, так и родительский кэш. Приведенный ниже пример показывает что kesha.ru получает данные из доменов .ru, .au, .aq, .fj, .nz, а gesha.ru — из доменов .uk, .de, .fr, .no, .se, .it.

```
cache_host kesha.ru parent 3128 3130
cache_host gesha.ru parent 3128 3130
cache_host uc.cache.nlanr.net sibling 3128 3130
cache_host bo.cache.nlanr.net sibling 3128 3130
cache_host_domain kesha.ru.ru.au.aq.fj.nz
cache_host_domain gesha.ru.uk.de.fr.no.se.it
```

Transparent proxy

Transparent proxy — это таким образом настроенный proxy-сервер, что его использование прозрачно для пользователей. То есть пользователям не придется что либо настраивать в своих браузерах. Для этого необходимо решить следующие задачи:

1. Добиться чтобы все HTTP-запросы пользователей попали на компьютер, где работает ваш HTTP proxy-сервер.
2. Добиться чтобы эти запросы попадали собственно к proxy-серверу.
3. Добиться того чтобы ваш proxy-сервер их правильно обработал.

Выполнить первый пункт можно разными способами. Самый простой путь — поставить proxy-сервер и маршрутизатор на один сервер, через который проходит весь трафик.

Чтобы HTTP-запросы пользователей попали к HTTP proxy-серверу, необходимо таким образом настроить маршрутизатор (брандмауэр), чтобы тран-

зитные пакеты, предназначенные для порта 80, попадали на вход proxy-сервера. Если proxy-сервер должным образом настроен, он правильно обработает полученные запросы. В Squid.conf добавляются следующие строчки:

```
httpd_accel www.your.domain 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Ключи запуска Squid

Помимо конфигурационного файла, поведением программы Squid можно управлять с помощью ключей командной сторки. Далее приведены некоторые из них с поянениями:

- `-a` — указывает порт для входных HTTP-запросов;
- `-d` — выводит отладочную информацию на устройство stderr (обычно — текущая консоль);
- `-f имя_файла_конфигурации` — позволяет использовать альтернативный конфигурационный файл (удобно для отладки сервера);
- `-h` — выводит краткую справку по программе Squid;
- `-k` — этот ключ позволяет посылать Squid следующие управляющие сигналы:
  - `reconfigure` — посылка сигнала `HUP`. Используется для прочтения измененного конфигурационного файла;
  - `rotate` — позволяет произвести ротацию журналов (сигнал `USR1`);
  - `shutdown` — прервать выполнение программы с корректным завершением (сигнал `TERM`);
  - `interrupt` — немедленно завершить работу программы (сигнал `INT`);
  - `kill` — "убить" приложение (`KILL`);
  - `debug` — начать/закончить полную трассировку (сигнал `USR2`);
  - `check` — проверка (сигнал `ZERO`);
- `-u` — задает порт для входных ICP-запросов;
- `-v` — выводит версию программы;
- `-z` — создает дисковый кэш при первом запуске (Важно!);
- `-D` — предписывает не производить DNS-тест при запуске;
- `-F` — восстанавливает после сбоя не в фоновом режиме (ускорение восстановления);
- `-N` — предписывает не становиться фоновым процессом;
- `-V` — включает поддержку виртуальных хостов для режима акселерации;

- `-X` — включает отладку при разборе конфигурационного файла;
- `-Y` — включает быстрое восстановление после сбоев.

Первый раз Squid нужно запускать с ключом `-z`:

```
Squid -z
```

При этом программа создаст дерево кэшей. Этой же командой следует воспользоваться с том случае, если вам необходимо очистить кэш proxy-сервера.

Для закрыатия текущих файлов журналов и создания новых (чистых) файлов используется команда:

```
Squid – k rotate
```

Файлы журналов Squid

Файл access.log

Файл access.log используется для хранения информации о всех подключениях к proxy-серверу. Запись добовляется, когда клиент закрывает соединение. Для сервера с большим трафиком файл может за день увеличиться на десятки мегабайт. К примеру, при трафике в 10 тыс. запросов в сутки объем журнала увеличивается примерно на 2 Мбайт.

Еденицей информации о соединении является строка. Строка состоит из десяти полей. Ниже приведено описание полей с пояснениями:

- `timestamp` — время в UNIX-формате (время с 1 января 1970 года в миллисекундах);
- `elapsed` — затраченное время в миллисекундах;
- `client IP address` — IP-адрес клиента, пославшего запрос;
- `type/HTTP` — результат запроса, где `type`:
  - `TCP_HIT` — верная копия объекта нашлась в кэше;
  - `TCP_MISS` — запрашиваемый объект не был в кэше;
  - `TCP_EXPIRED` — объект есть в кэше, но он устарел;
  - `TCP_CLIENT_REFRESH` — клиент запросил принудительное обновление объекта;
  - `TCP_REFRESH_HIT` — объект в кэше был старым, был сделан запрос к источнику и источник ответил "объект не изменился";
  - `TCP_REFRESH_MISS` — объект в кэше был старым, был сделан запрос к источнику и тот вернул обновленное содержание;
  - `TCP_IMS_HIT` — клиент выдал запрос, объект оказался в кэше и свежим;
  - `TCP_IMS_MISS` — клиент выдал запрос для просроченного объекта;

- TCP_REF_FAIL_HIT — объект в кэше устарел, но запросить новую копию не удалось;
- TCP_SWAPFAIL — объект должен находиться в кэше, но его не смогли извлечь;
- TCP_DENIED — отказ;

- size — количество байтов, переданных клиенту;
- method — метод передачи информации; GET, HEAD, POST для TCP-запросов или ICP_QUERY для UDP-запросов;
- URL — адрес запрашиваемого объекта;
- ident "-", если недоступен;
- hierarhy data/Hostname — результат запросов к братским/родительским кэшам:
  - PARENT_HIT — UDP-запрос к родительскому кэшу (parent) вернулся с подтверждением;
  - PARENT_UDP_HIT_OBJECT — объект оказался в родительском кэше parent и поместился в UDP-ответе;
  - DIRECT — объект был запрошен с оригинального сервера;
- тип содержимого (MIME-тип/подтип).

Файл store.log

Файл store.log используется для хранения информации о всех кэшируемых объектах proxy-сервера. Единицей информации о соединении является строка. Строка состоит из одиннадцати полей. Ниже приведены поля с пояснениями:

- Time — время в UNIX-формате (время с 1 января 1970 года в миллисекундах);
- action — действие:
  - RELEASE — удален из кэша;
  - SWAPOUT — сохранен на диск;
  - SWAPIN — был на диске, загружен в память;
- HTTP reply code — код ответа HTTP-сервера;
- HTTP Date — дата создания объекта;
- HTTP Last-Modified — время последней модификации объекта;
- HTTP Expires — срок жизни объекта;
- HTTP Content-Type — тип объекта;

- `HTTP Content-Length` — размер объекта;
- реально полученное число байтов. В том случае, если не совпадает с предыдущим полем, объект не сохраняется;
- `HTTP metod` — метод передачи информации (`GET`, `HEAD`, `POST`).
- `Access key` — ключ доступа (обычно URL).

Файл useragent.log

Предназначен для хранения информации о том, с какими пользовательскими агентами (Web-браузерами) работают клиенты. Малоинтересен в практическом плане. Разве что для получения статистики по частоте использования тех или иных Web-браузеров.

Нестандартные применения

Функциональность программы Squid не ограничивается только функцией proxy-сервера. У нее есть достаточно много других интересных применений. В этом разделе мы рассмотрим только некоторые из них.

Борьба с баннерами

Наверняка вам встречались Web-страницы, на которых нужной информации было от силы на килобайт, а рекламных баннеров (зачастую анимированных) — пять-шесть. Хорошо когда канал большой и бесплатный. Когда же пользуешься обычным коммутируемым соединением да еще платишь за соединение из своего кармана, каждый килобайт начинаешь считать. В этом случае можно настроить локальный сервер Squid таким образом, чтобы не происходила закачка ненужных баннеров. Этого можно добиться несколькими способами.

Вариант 1

Простой. На месте баннеров показываются разорванные картинки или перекрещенные прямоугольники (неполученные файлы).

1. Определяем сайты банерных сетей и создаем для них регулярные выражения.
2. Создаем в каталоге /usr/local/Squid/etc следующие файлы:
   - banners_path_regex — содержит по одному регулярному выражению на строку;
   - banners_regex — содержит по одному регулярному выражению на строку;

- banners_exclusion — это строки, трактуемые в предыдущих файлах как баннеры, но изменять которые не рекомендуется.

3. В Squid.conf добавляем следуюшие правила:

```
acl banners_path_regex urlpath_regex
"/usr/local/Squid/etc/banners_path_regex"
acl banners_regex url_regex "/usr/local/Squid/etc/banners_regex"
acl banners_exclusion url_regex
"/usr/local/Squid/etc/banners_exclusion"
http_access deny banners_path_regex !banners_exclusion
http_access deny banners_regex !banners_exclusion
```

Вариант 2

Замена рекламных баннеров на свою картинку, которая находится на локальном для proxy-сервера компьютере.

1. Определяем сайты банерных сетей и создаем для них регулярные выражения.
2. На своем сервере создаем "заменитель" рекламных картинок — файл mybanner.gif.
3. Настраиваем перенаправление (редиректор) в файле Squid.conf — redirect_program /usr/local/Squid/bin/banner.pl.
4. Создаем прстой скрипт на Perl — banner.pl:

```
#!/usr/bin/perl
$|=1;
while (<>)
{
 s@регулярное-выражение@http://www.myhost.org/mybanner.gif@;
 print;
}
```

Конечно, можно сделать более элегагантно, однако данный метод работает, а настроить proxy-сервер можно за пару минут.

Разделение внешнего канала

Часто бывает так, что у вас есть внешний канал — скажем, 128 Кбит, и есть несколько групп пользователей с определенным приоритетом. И требуется, чтобы группа 1 имела одну фиксированную ширину наружного канала (скажем, 64 Кбит), а группа 2 и 3 — ширину наружного канала по 32 Кбит. Для решения этой непростой задачи мы также можем воспользоваться Squid.

Немного терминологии:

- пул — набор групп "емкостей" определенного класса;
- группа "емкостей" — часть пула, привязанная к хосту, сети или общая для всех;
- "емкость" ограниченного объема — та, в которую с определенной скоростью вливается внешний трафик, и из которой он раздается клиенту.

Определены три класса пулов:

- одна "емкость" на всех из этого класса;
- одна общая "емкость" и 255 отдельных для каждого хоста из сети класса C;
- 255 "емкостей" для каждой сетки класса B и отдельная "емкость" для каждого хоста.

Пример конфигурации Squid для трех классов пулов:

```
delay_pools 3  # 3 poolы
delay_class 1 1 # 1 pool 1 класса
delay_class 2 1 # 2 pool 1 класса
delay_class 3 3 # 3 pool 3 класса
delay_access 1 allow staff
delay_access 1 deny all
delay_access 2 allow students
delay_access 2 deny all
delay_access 3 allow college
delay_access 3 deny all
delay_parameters 1 640000/640000
delay_parameters 2 64000/64000
delay_parameters 3 64000/64000 32000/64000 6400/32000
```

Строка, определяющая максимальную ширину виртуального канала, имеет следующий вид:

```
delay_parameters pool total_rest/total_max net_rest/net_max
ind_rest/ind_max
```

где:

- `pool` — номер пула, для которого определяются каналы;
- `total` — ширина канала на всех;
- `net` — ширина канала на подсеть;
- `ind` — ширина канала на отдельный адрес;
- `rest` — скорость заполнения (байт/с);
- `max` — объем "емкости" (байт).

Обработка статистики

В стандартную поставку пакета Squid входят скрипты, написанные на Perl, позволяющие создавать отчеты о работе программы Squid:

- access-extract.pl — скрипт получает на стандартный ввод журнал access.log и выдает на стандартный вывод промежуточный результат;
- access-summary.pl — скрипт получает на вход результат работы access-extract.pl и делает из него красивый отчет.

Программа Squid Cache and Web Utilities (SARG)

Программа, обрабатывающая журналы Squid и составляющая на их базе отчеты.

С помощью этой програмы можно получить следующую информацию:

- количество работавших пользователей;
- время их работы;
- трафик по каждому пользователю;
- использование кэша каждым пользователем;
- список Web-серверов, посещаемых пользователем;
- итоговые цифры по трафику и времени.

Существуют также дополнительные отчеты: Top sites и Useragents.

Отчет генерируется за период, интервал которого берется из log-файла Squid. В отчете, кроме зарегистрированных пользователей Squid, отражается информация о попытках незаконного вхождения в сеть и неправильных наборах пароля.

Если не производится ротация журналов Squid, то SARG генерирует отчеты нарастающим итогом. Отчеты хранятся в стандартном формате HTML-страниц, поэтому их можно просматривать через браузер, копировать и распечатывать. Также отчеты можно генерировать в определенный каталог или получать по почте.

Программа MRTG

Еще одна программа, позволяющая получать при соответствующей настройке отчеты о работе Squid. Вывод осуществляется в виде HTML-страниц.

Ссылки

- **http://www.Squid-cache.org** — официальный сайт Squid.
- **http://karjagin.narod.ru/solaris/Squid-faq-rus.html** — русский перевод Squid-faq.
- **http://www.nlanr.net/Cache/ICP/ICP-id.txt** — протокол Internet Cache Protocol.
- **http://Squid.org.ua** — зона особого внимания: сайт, полностью посвященный программе Squid.
- **http://linux.webclub.ru/security/proxy/Squid.html** — Иван Паскаль. Настройка Squid.
- **http://www.bog.pp.ru/work/Squid.html** — Bog BOS: Squid (кэширующий proxy для HTTP): установка, настройка и использование.
- **http://www.nitek.ru/~igor/Squid** — борьба с баннерами с помощью Squid.

Глава 23

Синхронизация времени через сеть, настройка временной зоны

Для комфортной работы с компьютером иногда может не хватать такой малости, как нормально настроенное системное время. Плохо, когда приходится ежедневно его подправлять или вручную производить переход на летнее/зимнее время.

Особенно это неприятно, когда компьютеров несколько десятков, и время у всех должно быть синхронизировано. Для синхронизации системного времени создателями Интернета был предусмотрен специальный сервис — сетевой протокол времени (Network time protocol, NTP).

Сетевой протокол времени

Сетевой протокол времени предназначен для синхронизации клиента или сервера точного времени с другим сервером точного времени или эталонным источником времени (радио, атомные часы и тому подобные устройства). Для локальной сети NTP способен обеспечить точность до миллисекунды, а для распределенной сети (в частности Интернета) достижима точность синхронизации порядка нескольких десятков миллисекунд. Последний стандарт этого протокола предусматривает криптографическую защиту передаваемых данных, одновременное подключение к нескольким серверам точного времени для достижения более точной синхронизации времени и повышения отказоустойчивости системы и многое другое.

Структура сети серверов точного времени многоуровневая. Главные серверы точного времени, напрямую подключенные к источнику эталонного времени, образуют первый уровень, серверы точного времени, присоединенные непосредственно к главным серверам, образуют второй уровень, и т. д.

В качестве сетевого протокола используется протокол UDP, порт 123. Для увеличения надежности и точности получаемых данных применяется фильт-

рация, селекция и комбинация пакетов на принципах максимальной вероятности, а также несколько резервных серверов и путей передачи.

Для передачи и хранения времени используется беззнаковое 64-битовое число с фиксированной точкой, которое хранит число секунд в формате UTC. Старшие 32 бита — число секунд, младшие 32 бита — дробная часть секунд. Достижимая точность — 232 пикосекунды. 0 означает неопределенное время.

Классы обслуживания

Служба точного времени имеет несколько классов обслуживания клиентов:

- multicast — предназначен для использования в быстрой локальной сети со множеством клиентов, где отсутствует необходимость в высокой точности. Принцип действия — один или более NTP-серверов рассылают широковещательное сообщение, клиенты определяют время, предполагая, что задержка составляет несколько миллисекунд. Сервер не принимает ответных NTP-сообщений;
- procedure-call — предназначен для получения высокоточного времени. NTP-клиент посылает запрос на сервер точного времени, который обрабатывает запрос и немедленно посылает ответ. Сервер не синхронизируется с клиентом;
- symmetric — предназначен для использования серверами точного времени. Представляет собой динамически реконфигурируемую иерархию серверов точного времени. Каждый сервер точного времени синхронизирует своих соседей и синхронизируется своими соседями в соответствии с правилами выбора соседей. Активный режим используется серверами точного времени низшего уровня с заранее определенными адресами соседей, пассивный режим используется серверами точного времени, близкими к первому уровню, и взаимодействующими с соседями с заранее неизвестными адресами.

Обеспечение достоверности данных

Алгоритм функционирования сервера точного времени подразумевает несколько способов для обеспечения достоверности данных.

- Если в течение восьми последовательных интервалов опроса от соседнего сервера точного времени не было сообщений, то этот сервер считается недостижимым.
- Осуществляется проверка времени:
  - если время передачи совпадает с временем предыдущего сообщения — дублированный пакет;

- если время отправки сообщения не совпадает с временем, содержащимся в пакете, сервер считает, что он получил фальшивый пакет.

- Имеется алгоритм защиты от очень старых сообщений.
- Аутентификатор состоит из ключа и шифрованной контрольной суммы, которая создается с использованием алгоритма шифрования DES.

Формат NTP-пакета

Пакет NTP включает следующие поля:

- LI (leap indicator) — в конце суток должна быть вставлена секунда для синхронизации атомных и астрономических часов;
- VN — номер версии протокола;
- mode — режим работы сервера точного времени;
- stratum — уровень сервера;
- precision — точность часов сервера;
- poll interval — интервал запросов. Используется наименьший интервал из своего сервера и сервера, отвечающего на запросы;
- synchronization distance — полный цикл обмена сообщениями до первичного источника;
- synchronization dispersion — дисперсия задержек синхронизации;
- reference clock identifier — тип источника времени;
- reference timestamp — время последнего изменения источника времени;
- originate timestamp — время соседа, когда было отправлено последнее NTP-сообщение;
- receive timestamp — местное время получения последнего NTP-сообщения;
- transmit timestamp — местное время отправки текущего сообщения;
- authenticator (96 bit) — ключ и шифрованная контрольная сумма сообщения.

Рекомендуемая конфигурация

Рекомендуемая конфигурация подразумевает наличие трех местных серверов точного времени, соединенных между собой, каждый из которых подключен к двум различным внешним серверам. Клиенты службы точного времени подключаются к каждому местному серверу точного времени.

Стандарты

Используемые для протокола NTP стандарты приведены в табл. 23.1.

Таблица 23.1. *Стандарты протокола NTP*

Стандарт	Название	Примечание
RFC 1128	Measured performance of the Network Time Protocol in the Internet system (Измерение производительности сетевого протокола времени в Интернете)	
RFC 1129	Internet time synchronization: The Network Protocol (Синхронизация времени через Интернет. Сетевой протокол времени)	Описывает процесс синхронизации времени
RFC 1165	Network Time Protocol (NTP) over the OSI Remote Operations Service (Сетевой протокол времени и взаимодействие с моделью OSI)	
RFC 1305	Network Time Protocol (v3) (Сетевой протокол времени, версия три)	Отменил стандарты RFC 1119, RFC 1059, RFC 958

Сервер xntpd

Для UNIX-платформы, в том числе и Linux, существует сервер точного времени, носящий название xntpd. Этот сервер полностью реализует стандарт RFC 1305 и имеет расширенные возможности, которые планируется включить в следующую версию стандарта. Входит в стандартную поставку большинства дистрибутивов Linux. Установка тривиальна. Файл конфигурации — /etc/ntp.conf.

Конфигурация сервера

Поскольку варианты конфигурирования сервера зависят от класса обслуживания, сервер имеет достаточно много настроек, которые в основном содержатся в конфигурационном файле /etc/ntp.conf.

Класс *symmetric*

Этот класс предназначен для конфигурирования сервера точного времени в режиме symmetric.

```
peer address [key key] [version version] [prefer] [minpoll minpoll]
[maxpoll maxpoll]
```

Здесь:

- *address* — адрес симметричного сервера;
- *key* — 32-битный ключ для поля аутентификации (по умолчанию отсутствует);
- prefer — предпочитать данный сервер при прочих равных условиях;
- *minpoll* — минимальный интервал запросов (секунды, 2 в степени minpoll в диапазоне от 4 (16 с) до 14 (16 384 с), по умолчанию 6 (64 с));
- *maxpoll* — максимальный интервал запросов (секунды, 2 в степени maxpoll, по умолчанию 10—1024 с).

Класс *procedure-call*

Этот класс предназначен для конфигурирования сервера точного времени в режиме procedure-call.

- server *address* [key *key*] [version *version*] [prefer] [mode *mode*]
- *address* — адрес сервера;
- *key* — 32-битный ключ для поля аутентификации (по умолчанию отсутствует);
- *mode* — режим.

Класс *multicast*

Предназначен для настройки multicast-режима. Обычно используется в локальных сетях.

- broadcast *address* [key *key*] [version *version*] [ttl *ttl*]
  - *address* — адрес симметричного сервера;
  - *key* — 32-битный ключ для поля аутентификации (по умолчанию отсутствует);
  - *version* — версия протокола;
  - *ttl* — время жизни пакета.
- broadcastclient [*address*] *address* — адрес клиента, получающего информацию.
- broadcastdelay секунд — позволяет самостоятельно указать задержку в распространении пакета.

Общие параметры

Здесь описываются общие параметры настройки сервера xntpd/:

- driftfile *driftfile* — определяет файл, в котором хранится и извлекается при запуске сдвиг частоты местных часов;

- `enable/disable` *`auth/monitor/pll/pps`*`/stats` — включить/выключить режим работы:
  - *`auth`* — с неупомянутыми соседями общаться только в режиме аутентификации;
  - *`monitor`* — разрешить мониторинг запросов;
  - *`pll`* — разрешать настраивать частоту местных часов по NTP;
  - *`stats`* — разрешить сбор статистики;
- `statistics` *`loopstats`* — при каждой модификации локальных часов записывает строчку в файл loopstats;

  Формат файла loopstats:
  - номер модифицированного дня по юлианскому календарю;
  - секунды с полуночи (UTC);
  - смещение в секундах;
  - смещение частоты в миллионных долях;
  - временная константа алгоритма дисциплинирования часов;
- `statistics peerstats` — каждое общение с соседом записывается в журнал, хранящийся в файле peerstats;

  Формат файла peerstats:
  - номер модифицированного дня по юлианскому календарю;
  - секунды с полуночи (UTC);
  - IP-адрес соседа;
  - статус соседа, шестнадцатеричное число;
  - смещение, с;
  - задержка, с;
  - дисперсия, с;
- `statistics clockstats` — каждое сообщение от драйвера локальных часов записывается в журнал, хранящийся в файле clockstats;
- `statsdir имя-каталога-со-статистикой` — задает имя каталога, в котором будут находиться файлы со статистикой сервера;
- `filegen [file filename] [type typename] [flag flagval] [link | nolink] [enable | disable]` — определяет алгоритм генерации имен файлов.

  Имена файлов состоят из следующих элеметов:
  - префикс — постоянная часть имени файла, задается либо при компиляции, либо специальными командами конфигурации;

- имя файла — добавляется к префиксу без косой черты, две точки запрещены, может быть изменена ключом `file`;
- суффикс — генерируется в зависимости от *`typename`*:
  - ◊ `none` — обычный файл;
  - ◊ `pid` — при каждом запуске xntpd создается новый файл (к префиксу и имени файла добавляются точка и номер процесса);
  - ◊ `day` — каждый день создается новый файл (к префиксу и имени файла добавляются .yyyymmdd);
  - ◊ `week` — каждую неделю создается новый файл (к префиксу и имени файла добавляются .yyyywww);
  - ◊ `month` — каждый месяц создается новый файл (к префиксу и имени файла добавляются .yyyymm);
  - ◊ `year` — каждый год создается новый файл (к префиксу и имени файла добавляются .yyyy);
  - ◊ `age` — новый файл создается каждые 24 часа (к префиксу и имени файла добавляются .a и 8-значное количество секунд на момент создания файла от момента запуска xntpd);
  - ◊ `link/nolink` — по умолчанию создается жесткая ссылка от файла без суффикса к текущему элементу набора (это позволяет обратиться к текущему файлу из набора используя постоянное имя);
  - ◊ `enable/disable` — разрешают/запрещают запись в соответствующий набор файлов;

❒ `restrict numeric-address [ mask` *`numeric-mask`* `] [flag] ...` — задает ограничение доступа: пакеты сортируются по адресам и маскам, берется исходный адрес и последовательно сравнивается, от последнего удачного сравнения берется флаг доступа:

- нет флагов — дать доступ;
- `ignore` — игнорировать все пакеты;
- `noquery` — игнорировать пакеты NTP 6 и 7 (запрос и модификация состояния);
- `nomodify` — игнорировать пакеты NTP 6 и 7 (модификация состояния);
- `notrap` — отказать в обеспечении mode 6 trap сервиса (удаленная журнализация событий);
- `lowpriotrap` — обслуживать ловушки, но прекращать обслуживание, если более приоритетный клиент потребует этого;

- `noserve` — обслуживать только запросы mode 6 и 7;
- `nopeer` — обслуживать хост, но не синхронизироваться с ним;
- `notrust` — не рассматривать как источник синхронизации;
- `limited` — обслуживать только ограниченное количество клиентов из данной сети;
- `ntpport/non-ntpport` — модификатор алгоритма сравнения адресов (сравнение успешно если исходный порт равен/неравен 123), алгоритм сортировки ставит эту строку в конец списка.

- `clientlimit limit` — для флага `limited` определяет максимальное количество обслуживаемых клиентов (по умолчанию 3);
- `clientperiod секунд` — сколько секунд считать клиента активным и учитывать при определении количества обслуживаемых клиентов;
- `trap host-address [port port-number] [interface interface-address]` — задать хост и порт, которые будут вести журнал;
- `setvar variable` — установка дополнительных переменных;
- `logfile имя-файла` — использовать файл имя-файла для ведения журнала вместо syslog;
- `logconfig keyword` — управление количеством сообщений, сбрасываемых в журнал. Ключевое слово может быть предварено символами равно (установка маски), минус (удаление класса сообщений), плюс (добавление); ключевое слово образуется слиянием класса сообщений (`clock`, `peer`, `sys`, `sync`) и класса событий (`info`, `event`, `statistics`, `status`); в качестве суффикса или префикса может использоваться слово `all`.

Обеспечение безопасности сервера

Если сервер точного времени не предназначен для широкой общественности, а используется только внутренней локальной сетью, желательно закрыть 123 порт для доступа извне, чтобы избежать возможной атаки типа denial of service (отказ в обслуживании), поскольку это грозит неправильным функционированием сервера. Также, если возможно, необходимо использовать шифрование.

Вот список правил для организации брандмауэра, закрывающего доступ к вашему серверу снаружи *(см. гл. 29)*:

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 123
ipchains -A input -p udp -j ACCEPT -s some.trusted.host -d 0.0.0.0/0 123
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 123
```

Программы и утилиты, относящиеся к службе точного времени

ntpdate

Эта утилита позволяет установить время на компьютере, используя список NTP-серверов.

Используемые ключи:

- `-B` — только плавный сдвиг, даже если смещение больше 128 мс;
- `-b` — всегда использовать settimeofday;
- `-d` — отладка;
- `-p число` — число запросов к каждому серверу (от 1 до 8, по умолчанию 4);
- `-q` — только запрос времени;
- `-s` — использовать syslog вместо stdout;
- `-t timeout` — время ожидания ответа (по умолчанию 1 с);
- `-u` — использовать непривилегированный порт.

ntpq

Утилита для получения состояния NTP-сервера и его изменения (использует NTP mode 6).

ntptrace

Утилита для поиска серверов первого уровня.

Используемые ключи:

- `-r число` — количество запросов (по умолчанию 5);
- `-t секунд` — время ожидания ответа (по умолчанию 2).

xntpd

Собственно демон точного времени. Используемые параметры при запуске:

```
xntpd [-aAbdm ] [-c config-file] [-f drift-file] [-k key-file]
      [-l log-file] [-p pid-file] [-r broadcast-delay] [-s stats-dir]
      [-t key] [-v variable] [-V variable]
```

Здесь:

- `-a` — разрешить аутентификацию;
- `-A` — запретить аутентификацию;

- `-b` — широковещательные сообщения;
- `-c config-file` — конфигурационный файл (по умолчанию /etc/ntp.conf);
- `-d` — отладка;
- `-f drift-file` — файл, хранящий смещение часов (по умолчанию /etc/ntp.drift);
- `-k key-file` — файл ключей (по умолчанию /etc/ntp.keys);
- `-l log-file` — файл протокола (по умолчанию syslog).

xntpdc

Утилита для запроса состояния NTP-сервера и его изменения. Применяется только для xntpd-серверов. Использует NTP mode 7.

Публичные NTP-серверы

Список публичных серверов точного времени можно найти в Интернете. В любом случае вам придется этот список публичных серверов протестировать, чтобы определить задержки и качество соединения. Попробуйте сначала получить список серверов точного времени вашего провайдера (провайдеров). В списке литературы приведена ссылка на список серверов точного уровня первого и второго уровней, можно попробовать синхронизироваться от них.

Клиентские программы для синхронизации времени

Сам по себе сервер точного времени не нужен, если у пользователей нет возможности синхронизировать время. В настоящее время практически для всех операционных систем есть программы для получения времени с серверов NTP.

UNIX/Linux

Для этих операционных систем можно на компьютере установить сервер xntpd и настроить его для получения точного времени. У этого решения есть как достоинства, так и недостатки. Положительным моментом является то, что мы можем максимально точно синхронизировать время и построить отказоустойчивую конфигурацию. Отрицательный момент — достаточно сложное конфигурирование сервера и относительно большой объем занимаемой оперативной памяти компьютера.

Более простой вариант — воспользоваться утилитой ntpdate. Маленькая, простая в конфигурировании, позволяющая получить достаточно точное время — расхождение порядка 100 миллисекунд. Для синхронизации времени следует выполнить следующую команду:

```
ntpdate -B ntp ntp2 ntp3
```

где `ntp`, `ntp2`, `ntp3` — адреса серверов точного времени. Рекомендуется добавить эту строчку в таблицу заданий crontab (если у вас, конечно, постоянное подключение к Интернету).

Apple

Для компьютеров фирмы Apple есть клиент NTP, называющийся macntp.

Windows

Для операционной системы Windows существует несколько клиентов службы точного времени. В частности программа AboutTime, которую можно получить по адресу **ftp.psn.ru/pub/abouttime_nomsie.zip**. Или программа AnalogX Atomic TimeSync, получить которую можно по адресу **www.analogx.com/contents/download/network/ats.htm**.

Настройка временной зоны

Обычно при установке операционной системы вы корректно выбираете свой часовой пояс. В качестве ориентира часового пояса в малых странах указывается столица государства, к примеру для Беларуси — Минск, для Украины — Киев. Для больших стран ориентируются на крупнейший город нужной часовой зоны.

Однако иногда можно ошибиться и выбрать зону неправильно или автоматически нажать ввод и получить временную зону, которую предлагает программа инсталляции дистрибутива по умолчанию. В результате — неправильное время.

Для корректной установки временной зоны необходимы всего два файла: /etc/localtime и /etc/sysconfig/clock. Первый файл отвечает за временную зону, а второй — за способ хранения и представления времени в системе. Рассмотрим поподробнее эти файлы.

/etc/localtime

Файл /etc/localtime представляет собой описание временной зоны, в которой определяется смещение часового пояса относительно Гринвича, даты перехода на летнее/зимнее время и некоторую дополнительную информа-

цию. Формат файла — бинарный. Обычно разработчики дистрибутива поставляют файлы для всех временных зон в различных представлениях и форматах, из которых можно выбрать подходящую временную зону. В дистрибутиве Red Hat Linux 7.2 скомпилированные файлы временных зон находятся в каталоге /usr/share/zoneinfo/. Найдя подходящий для вашей временной зоны файл, скопируйте его в каталог /etc и переименуйте в localtime. В том невероятном случае, если вам не подходит файл временной зоны — сделайте его самостоятельно. Для этого необходимо создать текстовый файл, содержащий описание вашей временной зоны и скомпилировать его утилитой zic. Ниже приведен пример текстового файла, описывающего временную зону с переключением на летнее/зимнее время для Украины:

```
Rule dst 1982 maximum — Apr lastSun 2:00 1 "EET DST"
Rule dst 1982 maximum — Oct lastSun 2:00 0  "EET DST"
Zone EET 2:00 dst %s
```

Подробное описание утилиты zic смотрите в соответствующей справочной странице man.

/etc/sysconfig/clock

Этот файл содержит описание часовой зоны и некоторых других параметров.

Пример:

```
ZONE="Europe/Kiev"
UTC=false
ARC=false
```

Как видно из примера, системные часы не используют универсальное представление времени, а система находится в Киевском часовом поясе (Гринвич + 2 часа).

Ссылки

- **www.bog.pp.ru/work/xntpd.html** — Сергей Богомолов. Bog BOS: xntpd (UNIX-сервер NTP — network time protocol).
- **www.bog.pp.ru/work/ntp.html** — Сергей Богомолов. Bog BOS: Network time protocol.
- **www.tomsknet.ru/ftp/docs/rfc/rfc1305.txt** — Network Time Protocol (Version 3) Specification, Implementation and Analysis (RFC 1305).
- **cisco.opennet.ru/docs/RUS/lasg/time.html** — сетевые сервисы: NTP.
- **www.psn.ru/net/servis/ntp.shtml** — как пользоваться службой NTP.

- **www.eecis.udel.edu/~ntp** — страница, посвященная xntp.
- **www.eecis.udel.edu/~ntp/ntp_spool/html/index.htm** — руководство по инсталляции и настройке xntp.
- **www.eecis.udel.edu/~ntp/database/faq.html** — часто задаваемые вопросы по протоколу NTP.
- **www.eecis.udel.edu/~mills/ntp/clock1.html** — официальный список NTP-серверов первого уровня.
- **www.eecis.udel.edu/~mills/ntp/clock2.html** — официальный список NTP-серверов второго уровня.
- **www.eecis.udel.edu/~mills/** — официальный сайт разработчика стандарта NTP.

Глава 24

Сервер Samba — для клиентов Windows

Подобно службам NFS и Mars, которые позволяют сетевым пользователям подключаться к удаленным дискам и каталогам (первая служба для UNIX-систем, вторая для Novell), в операционной системе Linux существует пакет Samba, предназначенный для клиентов сети Microsoft Windows.

Этот пакет позволяет Linux-системе выступать в качестве файл- и принт-сервера в сети Microsoft Windows. Существует также и Samba-клиент для операционной системы Linux, позволяющий Linux-клиенту подключаться к ресурсам, предоставляемым серверами сети Microsoft Windows.

Такая объединенная схема дает ряд преимуществ:

- поскольку в целом операционная система Linux устойчивее Windows 9*x*, повышается надежность функционирования системы;
- отпадает необходимость приобретать лицензионную Windows;
- если у вас уже есть Linux-сервер, представляется рациональным нагрузить его дополнительной работой;
- сервер Samba имеет возможность мониторинга и удаленного управления как через SSH, так и через Web-интерфейс, предоставляемый пакетом SWAT (Samba Web-based Administrative Tool).

Установка сервера Samba проблем не вызывает — достаточно при инсталляции Red Hat Linux отметить соответствующий пакет RPM. Если вы не установили Samba при инсталляции дистрибутива — не беда, командой `rpm -i sambaXXX.rpm` сервер будет установлен на вашем компьютере.

В том случае, если вы хотите установить самую свежую версию пакета, и она досталась вам в виде tgz-архива, содержащего исходный текст, процесс установки несколько растянется.

1. Сначала необходимо распаковать архив, содержащий исходные коды Samba. Для этого надо выполнить следующую команду:

   ```
   tar zxvf samba-X.X.X.tar.gz
   ```

   где `X.X.X` — версия пакета.

2. После этого следует перейти в каталог /source, где находятся исходные коды. Находится там и файл Readme, в котором подробно рассказано, как сконфигурировать и произвести компиляцию и установить пакет Samba.
3. Набрать в командной строке и выполнить следующую команду:

   ```
   configure --with-smbmount --prefix=/opt/samba --with-msdfs
   ```

   Эта команда производит конфигурирование файла Makefile.

   Замечание

   Команда указывает компилировать утилиту smbmount, которая служит для монтирования SMB-ресурсов в файловую структуру Linux, включает поддержку Microsoft DFS и указывает устанавливаться после компиляции в каталог /opt/samba. Конечно, есть еще много параметров, которые можно назначить. Подробную информацию о них следует смотреть в документации к пакету Samba или вызывать командой `configure --help`.

4. Следующим действием необходимо набрать в командной строке `make` и нажать <Enter>. Этой командой запускается процесс компиляции программного пакета.
5. Если в ходе работы программы make не появились сообщения об ошибках, то надо выполнить команду `make install`. Эта команда установит пакет Samba в ее родной каталог (если действовать в точности по инструкции, то файлы попадут в каталог /opt/samba).

Теперь на очереди конфигурирование сервера Samba.

Файл конфигурации smb.conf

Самое трудное, с чем можно столкнуться при настройке сервера Samba, — это создание (или редактирование) файла конфигурации. Все файлы конфигурации Samba находятся в каталоге /etc/samba. Вот список файлов, которые обычно содержатся в этом каталоге:

- lmhosts — содержит список хостов и соответствующих им адресов;
- smbpasswd — содержит пароли пользователей сервера Samba;
- smbusers — файл, предназначенный для хранения списка пользователей, которым разрешен доступ к ресурсам Samba;
- smb.conf — главный конфигурационный файл сервера.

Примеры конфигурационных файлов, поставляемых с пакетом, находятся в каталоге /examples. В большинстве случаев их можно использовать в качестве базы.

Далее приведен файл smb.conf сервера Samba, который успешно функционирует на одном из серверов.

Листинг 24.1. Пример файла smb.conf

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps too
# many!) most of which are not shown in this example
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentry and a ; for parts of the config file that you
# may wish to enable
# NOTE: Whenever you modify this file you should run the command
"testparm"
# to check that you have not many any basic syntactic errors.
#
#======================= Global Settings ==============================
[global]

# workgroup = NT-Domain-Name or Workgroup-Name
   workgroup = Kontora

# server string is the equivalent of the NT Description field
   server string = Kontora Samba Server

# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
hosts allow = 192.168.10. 193.166.17.
# if you want to automatically load your printer list rather
# than setting them up individually then you'll need this
#   printcap name = /etc/printcap
#   load printers = yes

# It should not be necessary to spell out the print system type unless
# yours is non-standard. Currently supported print systems include:
```

```
# bsd, sysv, plp, lprng, aix, hpux, qnx
   printing = lprng

# Uncomment this if you want a guest account, you must add this to
/etc/passwd
# otherwise the user "nobody" is used
;  guest account = pcguest

# this tells Samba to use a separate log file for each machine
# that connects
   log file = /var/log/samba/%m.log

# Put a capping on the size of the log files (in Kb).
   max log size = 0

# Security mode. Most people will want user level security. See
# security_level.txt for details.
   security = user
# Use password server option only with security = server or
# security = domain
;   password server = <NT-Server-Name>

# Password Level allows matching of _n_ characters of the password for
# all combinations of upper and lower case.
;  password level = 8
;  username level = 8

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# Do not enable this option unless you have read those documents
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd

# The following are needed to allow password changing from Windows to
# update the Linux sytsem password also.
# NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
# NOTE2: You do NOT need these to allow workstations to change only
#        the encrypted SMB passwords. They allow the Unix password
#        to be kept in sync with the SMB password.
```

```
;  unix password sync = Yes
;  passwd program = /usr/bin/passwd %u

# Unix users can map to different SMB User names
;  username map = /etc/samba/smbusers

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
;   include = /etc/samba/smb.conf.%m

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
   socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

# Configure Samba to use multiple interfaces
# If you have multiple network interfaces then you must list them
# here. See the man page for details.
interfaces = 192.168.10.0/24

# Configure remote browse list synchronisation here
#  request announcement to, or browse list sync from:
#  a specific host or from / to a whole subnet (see below)
;   remote browse sync = 192.168.3.25 192.168.5.255
# Cause this host to announce itself to local subnets here
;   remote announce = 192.168.1.255 192.168.2.44

# Browser Control Options:
# set local master to no if you don't want Samba to become a master
# browser on your network. Otherwise the normal election rules apply
;   local master = no

# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
;   os level = 33

# Domain Master specifies Samba to be the Domain Master Browser. This
# allows Samba to collate browse lists between subnets. Don't use this
```

```
# if you already have a Windows NT domain controller doing this job
;   domain master = yes

# Preferred Master causes Samba to force a local browser election on
#startup and gives it a slightly higher chance of winning the election
;   preferred master = yes

# Enable this if you want Samba to be a domain logon server for
# Windows95 workstations.
;   domain logons = yes

# if you enable domain logons then you may want a per-machine or
# per user logon script
# run a specific logon batch file per workstation (machine)
;   logon script = %m.bat
# run a specific logon batch file per username
;   logon script = %U.bat

# All NetBIOS names must be resolved to IP Addresses
# 'Name Resolve Order' allows the named resolution mechanism to be
# specified the default order is "host lmhosts wins bcast".
name resolve order = wins lmhosts bcast

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable it's WINS
Server
wins support = yes

# WINS Server - Tells the NMBD components of Samba to be a WINS Client
#  Note: Samba can be either a WINS Server, or a WINS Client, but NOT
both
;   wins server = w.x.y.z

# WINS Proxy - Tells Samba to answer name resolution queries on
# behalf of a non WINS capable client, for this to work there must be
# at least one  WINS Server on the network. The default is NO.
;   wins proxy = yes

# DNS Proxy - tells Samba whether or not to try to resolve NetBIOS names
# via DNS nslookups. The built-in default for versions 1.9.17 is yes,
```

```
# this has been changed in version 1.9.18 to no.
   dns proxy = no

# Case Preservation can be handy - system default is _no_
# NOTE: These can be set on a per share basis
;  preserve case = no
;  short preserve case = no
# Default case is normally upper case for all DOS files
default case = lower
# Be very careful with case sensitivity - it can break things!
case sensitive = no

client code page = 866
character set = koi8-r
printer driver file=/home/samba/hplj1200/printers.def
#============================ Share Definitions ========================
[homes]
   comment = Home Directories
   browseable = no
   writable = yes
   valid users = yura tol katya slava vova lena alst

[comm]
        comment = Common place
        path = /home/samba/comm
        valid users = root slava tol yura katya vova lena alst
        public = no
        writable = yes
        printable = no
        create mask = 0775
        directory mask= 0775
        force group = office
[hp]
   comment = HP LaserJet 1200 Series PCL6
   path = /var/spool/samba
   printer = lp
   public = no
   printable = yes
```

```
   printer driver=HP LaserJet 1200 Series PCL6
   printer driver location=\\%h\printer$

[printer$]
  path=/home/samba/hplj1200
  public=yes
  browseable=yes

# This one is useful for people to share files
[tmp]
   comment = Temporary file space
   path = /tmp
   read only = no
   public = yes
```

Как видно из примера, конфигурационный файл разбит на разделы. Каждый раздел начинается с заголовка раздела, такого как `[global]`, `[homes]` и т. д. Структурой конфигурационный файл сильно напоминает ini-файлы операционной системы Windows. Символы `#` и `;` используются в качестве признаков комментария.

Секция *[global]*

Секция `[global]` определяет переменные, которые Samba будет использовать для определения доступа ко всем ресурсам. Рассмотрим переменные секции `[global]`.

- `workgroup = Kontora`

  Переменная `workgroup` содержит имя NT-домена или имя рабочей группы, к которой будет принадлежать сервер Samba.

- `netbios name = bw`

  Переменная `netbios name` задает имя сервера для отклика по протоколу NetBIOS. Не делайте его таким же, как и имя рабочей группы.

- `server string = Kontora Samba Server`

  Переменная `server string` содержит описание сервера (комментарий).

- `hosts allow = 192.168.10. 197.64.17.`

  Переменная `hosts allow` содержит список IP-адресов компьютеров и сетей, разделенных пробелом, которые имеют право подключаться к ресурсам вашего сервера Samba.

- `printing = lprng`

  Переменная `printing` определяет тип системы печати; поддерживается `bsd`, `sysv`, `plp`, `lprng`, `aix`, `hpux`, `qnx`.

- `guest account = pcguest`

  Переменная используется, если вы хотите разрешить гостевой вход на Samba-сервер. Соответствующего пользователя так же придется завести в Linux-системе. Однако по соображениям безопасности не рекомендуется разрешать гостевой вход.

- `log file = /var/log/samba/%m.log`

  Переменная `log file` указывает серверу создавать log-файлы отдельно для каждого пользователя; заодно указывает каталог, где будут создаваться файлы.

- `max log size = 0`

  Переменная `max log size` определяет максимальный размер log-файла.

- `security = user`

  Переменная `security` используется для задания уровня безопасности системы; обычно используется уровень `user`, так же используют уровни `share`, `server` и уровень `domain`.

- `password server = <NT-Server-Name>`

  Переменная `password server` используется только совместно с параметрами `security = server` или `security = domain`; задает имя сервера паролей.

- `password level` и `username level`

  Переменные `password level` и `username level` позволяют задать количество символов пароля и имени пользователя.

- `encrypt passwords = yes`

  Переменная `encrypt passwords` позволяет использовать пересылку паролей пользователей в зашифрованном виде; если задать `encrypt passwords = no`, то пароли пользователей будут пересылаться в незашифрованном виде, что очень плохо с точки зрения безопасности.

- `smb passwd file = /etc/samba/smbpasswd`

  Переменная `smb passwd file` задает путь и имя файла, содержащего пароли пользователей; поскольку принципы хранения пароля в Linux не позволяют его расшифровать, приходится создавать отдельный файл паролей для пользователей Samba.

- `local master = yes`

  Переменная `local master` позволяет серверу Samba стать мастер-браузером.

- `preferred master = yes`

  Переменная `preferred master` позволяет серверу Samba сразу же при запуске устроить перевыборы master с наибольшим шансом для себя.

- `dns proxy = yes`

  Разрешает серверу сопоставлять NetBIOS-имена с IP-адресом при помощи DNS.

  Замечание

  Протокол NetBIOS в принципе предназначен для одноранговой локальной сети, т. е. такой сети, где все компьютеры равноправны. Тем не менее, в NetBIOS предусмотрен специальный компьютер, называемый master, который ведет список компьютеров, подключенных к сети, их разделяемые ресурсы и вновь подключаемые компьютеры. Именно от master вновь подключающиеся компьютеры получают список компьютеров в сети и их доступные ресурсы.

- `username map = /etc/samba/smbusers`

  Переменная `username map` позволяет задать файл пользователей Samba, в котором ставится в соответствие имя Linux-пользователя имени Samba-пользователя; обычно в качестве имени пользователя Samba используется имя Linux-пользователя.

- `socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192`

  Переменная `socket options` используется для тонкой настройки сетевых параметров, позволяющих несколько улучшить производительность сервера.

- `interfaces = 192.168.10.0/24`

  Переменная `interfaces` указывает серверу, с какой сетевой картой (сетью) он имеет дело; используется в том случае, если на сервере установлено несколько сетевых карт из разных локальных сетей.

- `name resolve order = wins lmhosts bcast`

  Переменная `name resolve order` определяет порядок получения имен.

- `wins support = yes`

  Переменная `wins support` указывает, что сервер Samba выступает в роли WINS-сервера.

- `wins server = w.x.y.z`

  Переменная `wins server` определяет IP-адрес WINS-сервера; если установлено `wins support = yes`, то использование переменной `wins server` запрещено.

- `default case = lower`

  Переменная `default case` определяет регистр имен файлов, создаваемых на ресурсах Samba.

- `case sensitive = no`

  Переменная `case sensitive` определяет чувствительность к регистру символов.

- `client code page = 866`

  Переменная `client code page` задает кодовую страницу клиента; для DOS клиента — 866.

- `character set = koi8-r`

  Переменная `character set` задает набор символов, используемых сервером.

- `printer driver file=/home/samba/hplj1200/printers.def`

  Переменная `printer driver file` определяет имя драйвера принтера.

- `time server = true`

  Эта переменная предписывает серверу показывать клиентам Windows, что он выступает для них в роли сервера точного времени.

Секция *[homes]*

Секция `[homes]` позволяет удаленным пользователям получить доступ к своим домашним каталогам на Linux-машине. Для этого пользователь должен быть зарегистрирован в Linux-системе. Рассмотрим переменные секции `[homes]`.

- `comment = Home Directories`

  Эта переменная — просто комментирует содержимое данной секции.

- `browseable = no`

  Переменная запрещает просматривать каталог посторонним пользователям.

- `writable = yes`

  Переменная разрешает записывать в домашний каталог.

- `valid users = yura katya vova alst`

  Переменная `valid users` задает список пользователей, для которых разрешен доступ к своим домашним каталогам; в принципе параметр не обязательный.

Секция *[comm]*

Секция `[comm]` отвечает за каталог, доступный всем пользователям Samba. Это своего рода аналог FTP, куда могут записывать и откуда читать пользователи. Подробнее разберем эту секцию.

- `comment = Common place`

  Эта переменная — просто комментирует содержимое данной секции.

- `path = /home/samba/comm`

  Переменная определяет каталог, который используется для совместного доступа.

- `valid users = root yura katya vova alst`

  Переменная содержит список пользователей, которым разрешен доступ к общему ресурсу.

- `public = no`

  Запрещает остальным пользователям получать доступ к данному ресурсу.

- `writable = yes`

  Разрешает запись в общий ресурс.

- `printable = no`

  Указывает, что разделяемый ресурс не является печатающим устройством.

- `create mask = 0775`

  Маска для создания файлов на разделяемом ресурсе.

- `directory mask= 0775`

  Маска для создания каталогов на разделяемом ресурсе.

- `force group = office`

  Переменная определяет, что файлу, создаваемому или копируемому на общий ресурс, принудительно задается принадлежность к группе office, для того чтобы любой пользователь мог изменить или удалить файл.

Секция *[tmp]*

Секция `[tmp]` предназначена для создания разделяемого ресурса, в который могли бы записывать все пользователи. Как видно из нижеприведенного описания, от секции `[comm]` отличается отсутствием списка пользователей и значением переменной `public`.

```
comment = Temporary file space
path = /tmp
read only = no
public = yes
```

Пароли пользователей

Сервер Samba подразумевает использование нескольких типов безопасности. В частности переменная `encrypt password` определяет, какой механизм авторизации будет использован. Если переменной `encrypt password` присвоено значение `no`, то авторизация пользователей производится, исходя из учетных

записей Linux, хранящихся в файлах /etc/passwd и /etc/shadow. При таком типе авторизации пользователя пароли передаются по сети в незашифрованном виде. Это несколько упрощает настройку, но резко снижает безопасность системы. В дополнение к этому, такой тип авторизации требует в Windows 95, Windows 98, Windows NT изменений в системном реестре. Ниже приведены изменения, которые необходимо внести в системный реестр.

- Windows 95

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"EnablePlainTextPassword"=dword:00000001
```

- Windows 98

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"EnablePlainTextPassword"=dword:00000001
```

- Windows NT

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdr\Parameters]
"EnablePlainTextPassword"=dword:00000001
```

- Windows 2000

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkStatio
n\Parameters]
"EnablePlainTextPassword"=Data: 0x01
```

В том случае, если переменной `encrypt password` присвоено значение `yes`, авторизация пользователя происходит с использованием файла /etc/samba/smbpasswd, и передача паролей происходит в зашифрованном виде.

Почему для использования шифрованных паролей необходимо создавать отдельную базу паролей пользователей Samba? Все дело в методе хранения пароля. Windows-системы хранят зашифрованный пароль, и при аутентификации пользователя производят сверку паролей. Linux *не хранит* пароль как таковой. В файле shadow хранится так называемый хэш пароля, а в последних версиях Linux — контрольная сумма пароля, рассчитанная по алгоритму MD5. И при аутентификации пользователя сравниваются хэши паролей. Особенность хэша — он необратим, т. е. зная хэш, невозможно по нему восстановить пароль. Поэтому приходится отдельно для Samba заводить базу паролей пользователей. Для администратора системы это представляет некоторое неудобство — еще один повод забыть прописать пользователя, а с другой стороны — за все надо платить.

Добавление пользователей Samba

Для добавления пользователей в файл /etc/samba/smbpasswd необходимо наличие самого файла /etc/samba/smbpasswd. Также должна существовать и

учетная запись пользователя в Linux-системе. Если эти условия соблюдены, следует:

1. Воспользоваться программой smbpasswd для создания учетной записи

   ```
   smbpasswd -a user_name
   ```

2. Активировать учетную запись

   ```
   smbpasswd -e user_name
   ```

3. Эту операцию придется произвести с каждым пользователем. Существуют скрипты, позволяющие перебросить пользователей из файла passwd в файл smbpasswd. Но они только перебрасывают пользователей, а пароли для них все равно придется заводить вручную. Еще один недостаток этих скриптов — после них придется удалять пользователей типа nobody, root, news и т. п.

Для монтирования ресурсов, предоставляемых сервером Samba, используются команды `smbclient` и `smbmount`. Обо всех возможностях этих команд можно узнать из соответствующих man-страниц, краткие сведения о команде `smbclient` вы получите в этой главе.

Принтеры

Принтер, установленный в системе с сервером Samba, предоставить в общее пользование Samba-клиентам очень просто. Все принтеры, которые определены в файле /etc/printcap, становятся доступными после того, как вы добавите следующую секцию в конфигурационный файл smb.conf:

```
[printers]
path = /var/spool/lpd
writeable = no
guest ok = no
printable = yes
```

Использование ресурсов Samba

Хотя сервер Samba позиционируется как средство доступа Windows-клиентов к ресурсам Linux-систем, тем не менее, в пакете есть средства для того, чтобы Linux-компьютеры могли также просматривать и монтировать SMB-ресурсы. И что особенно приятно, доступ к ресурсам Windows-сети можно получить и в том случае, когда сервером является машина с Windows!

Программа клиента SMB для Linux включена в дистрибутив Samba и называется smbclient. Она обеспечивает FTP-подобный интерфейс командной строки. Также существует пакет smbfs, который позволяет монтировать и размонтировать SMB-ресурсы.

Для того чтобы увидеть доступные SMB-ресурсы, выполните команду:

```
/usr/bin/smbclient -L host
```

где `host` — это имя машины, ресурсы которой вас интересуют. Эта команда вернет список имен доступных сервисов.

Пример команды `smbclient`:

```
smbclient -L ziga
Server time is Sat Aug 19 19:58:27 1999
Timezone is UTC+2.0
Password:
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]

Server=[ZIGA] User=[] Workgroup=[WORKGROUP] Domain=[]

        Sharename      Type      Comment
        ---------      ----      -------
        ADMIN$         Disk      Remote Admin
        public         Disk      Public
        C$             Disk      Default share
        HP             Printer   HP6L


This machine has a browse list:

        Server               Comment
        ---------            -------
        HOP                  Samba 1.9.15p8
        ZIGA
```

Для использования сервиса выполните следующую команду:

```
/usr/bin/smbclient service <password>
```

где `service` — имя хоста и сервиса. Например, если вы пытаетесь обратиться к каталогу, который доступен под именем `public` на машине, названной `ziga`, то имя сервиса должно представлять собой `\\ziga\public`. Поскольку в языке C обратный слэш является спецсимволом, то практически необходимо ввести такую строку:

```
/usr/bin/smbclient \\\\ziga\\public mypasswd
```

где *`mypasswd`* — ваш пароль.

В результате вы должны получить приглашение smbclient:

```
smb: \>
```

Для получения справки необходимо ввести `h` и нажать <Enter>:

```
smb: \> h
ls              dir             lcd             cd              pwd
get             mget            put             mput            rename
more            mask            del             rm              mkdir
md              rmdir           rd              prompt          recurse
translate       lowercase       print           printmode       queue
cancel          stat            quit            q               exit
newer           archive         tar             blocksize       tarmode
setmode         help            ?               !
smb: \>
```

Как видите, практически все команды дублируют команды FTP-клиента.

Утилита smbclient многое позволяет, однако она утомительна для использования. Если от Windows-сети нужен только доступ к дисковым ресурсам — рекомендуется воспользоваться пакетом Smbfs.

В пакет Smbfs входят утилиты smbmount и smbumount, которые работают подобно mount и umount.

Так же есть графическая утилита gnomba — подобная утилите Сеть Windows.

Утилиты

Как и у других подобных проектов, для пакета Samba существует достаточно много сторонних утилит, позволяющих упростить конфигурирование и доступ к ресурсам.

Вот список утилит и программ, в той или иной мере относящихся к пакету Samba:

- `smbstatus` — утилита для мониторинга Samba;
- `SWAT` — инструмент для конфигурирования Samba через Web-интерфейс;
- `smbpasswd` — управление паролями Samba;
- `testparm` — проверка конфигурационного файла;
- `testprns` — проверка конфигурации принтера;
- `smbtar` — SMB-утилита резервного копирования;
- `smbclient` — клиент командной строки;
- `Ksamba` — KDE-программа, предназначенная для конфигурации;

- `Smbedit` — Win32-приложение для правки конфигурационного файла Samba;
- `Webmin` — универсальная программа конфигурации через Web-интерфейс, в том числе и Samba;
- `GSMB` — графический интерфейс для утилиты smbpasswd;
- `SambaSentinel` — графический интерфейс для утилиты smbstatus.

SWAT

SWAT (Samba Web Administration Tool) — одна из наиболее известных утилит для работы с сервером Samba через Web-интерфейс. Для доступа к SWAT в браузере необходимо набрать `localhost:901`. Далее, после ввода логина и пароля вы получаете доступ к программе SWAT, которая охватывает практически все настройки Samba, доступные через Web-интерфейс.

Webmin

Программа с Web-интерфейсом, позволяющая конфигурировать множество служб и сервисов через Web (рис. 24.1, 24.2). В частности есть возможность настраивать сервер Samba.

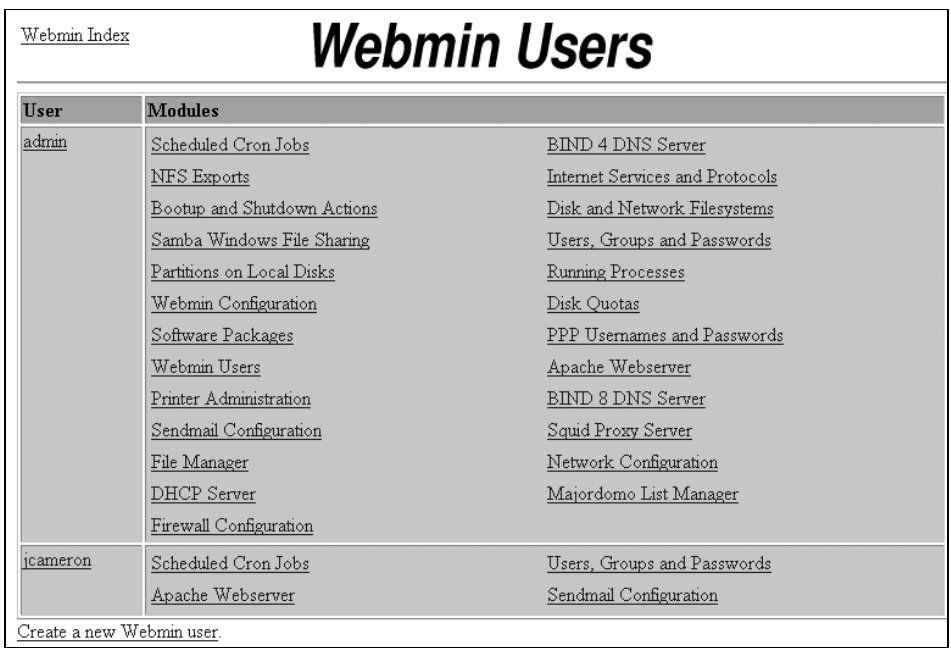

Рис 24.1. Webmin — список сервисов

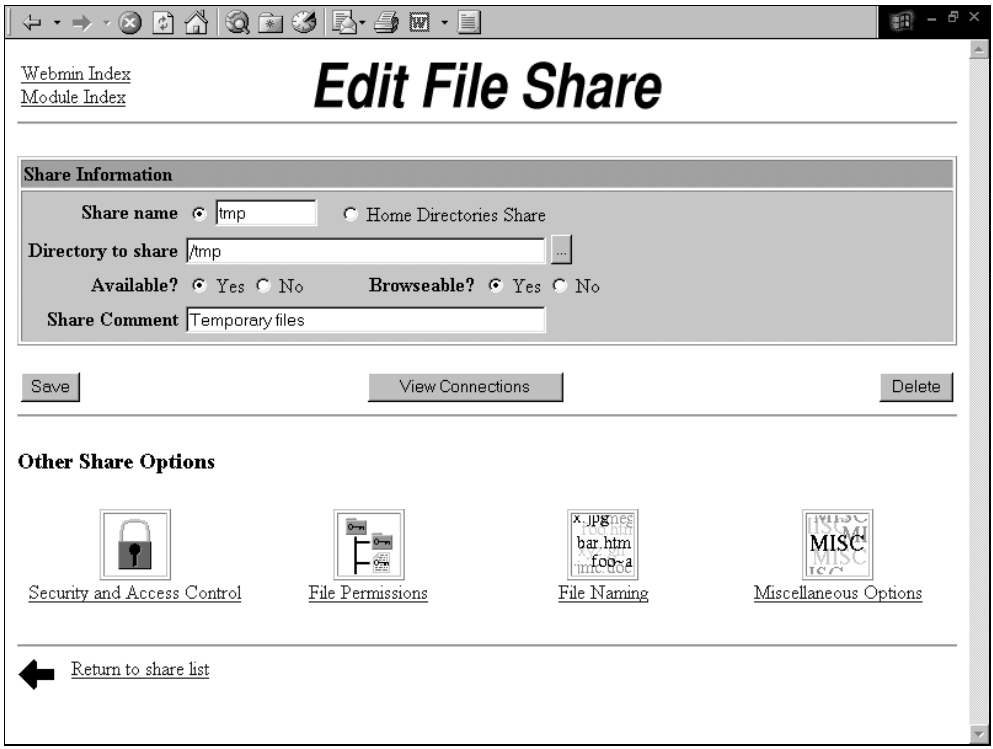

Рис 24.2. Webmin — управление Samba

Ksamba

Программа для KDE-оболочки, предназначенная для конфигурации Samba (рис. 24.3). Достаточно удобная и понятная.

GSMB

Графический интерфейс к утилите smbpasswd — намного приятней работать (рис. 24.4).

SambaSentinel

Графический интерфейс к утилите smbstatus (рис. 24.5). Позволяет производить мониторинг, удалять зависшие задачи.

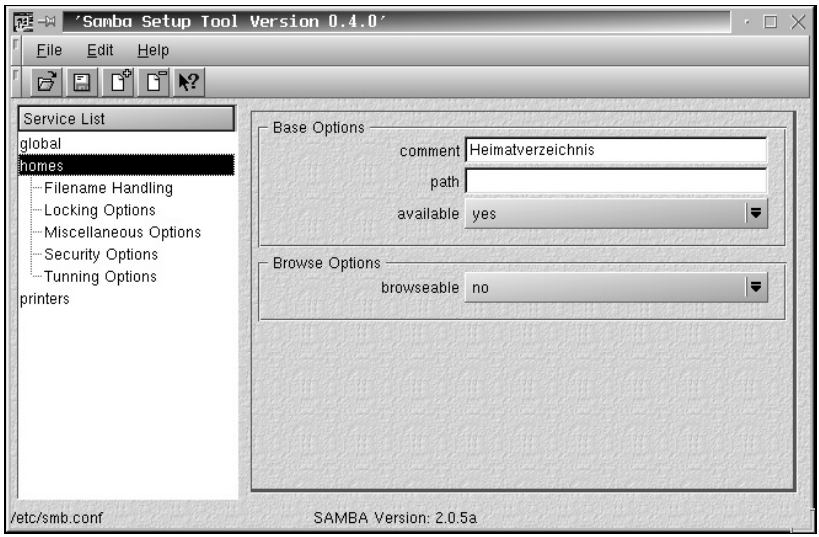

Рис 24.3. Ksamba

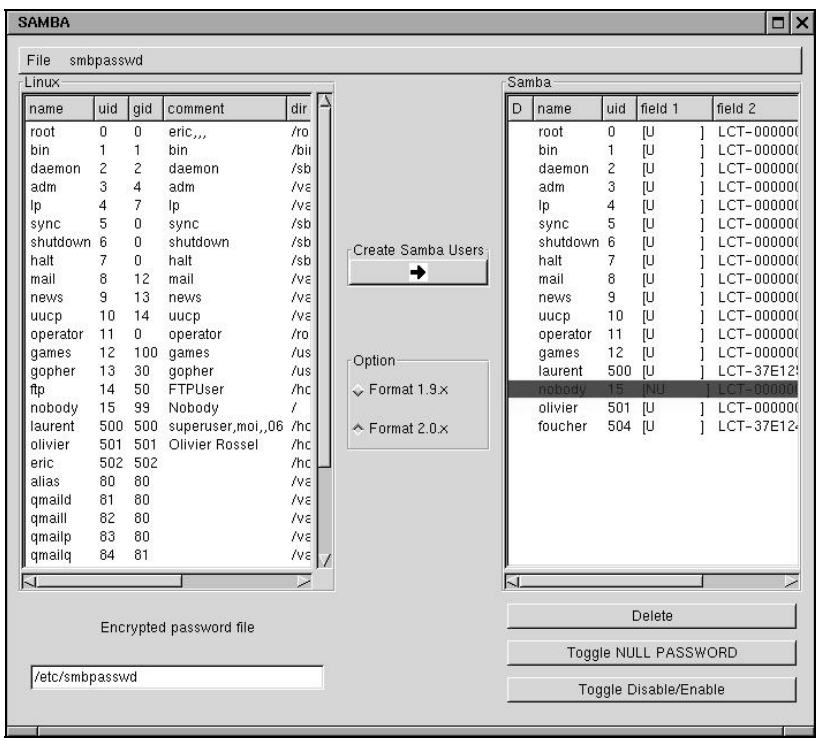

Рис 24.4. GSMB

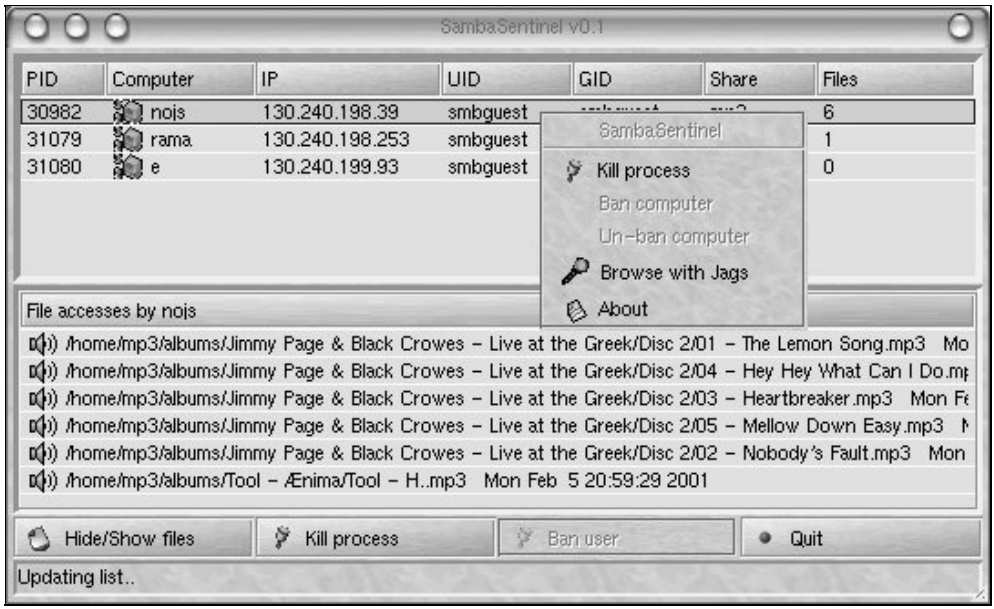

Рис 24.5. SambaSentinel

Ссылки

- **www.linuxoid.ru/how_to/samba5.html** — Илья Басин. Samba за пять минут.
- **www.samba.org** — официальный сайт проекта Samba.
- **www.webmin.com** — официальный сайт проекта Webmin.
- **www.culte.org/projets/developpement/gsmb/** — официальный сайт проекта GSMB.
- **boombox.campus.luth.se/sambasentinel.php** — сайт проекта SambaSentinel.
- **www.linux.org.ru/books/HOWTO/SMB-HOWTO.html** — SMB-HOWTO (русский перевод).

Глава 25

Linux — для клиентов Novell

В последние несколько лет протокол TCP/IP стал стандартом de-facto при построении сетей, однако еще лет пять назад в корпоративной среде в качестве сетевого протокола повсеместно использовался протокол IPX, разработанный фирмой Novell.

Фирма Novell стояла у истоков локальной сети как таковой. До сих пор на большинстве сетевых карт написано "Ready for Novell" или "NE compatible" — что означает совместимость с сетевой картой производства фирмы Novell (которые не производятся уже лет 10—15). Флагманский продукт фирмы Novell — программное обеспечение для сервера, носящее название "Novell Netware". Это программное обеспечение на просторах СНГ долгое время занимало монопольное положение при построении локальных сетей на базе недорогих компьютеров *x*86.

Наиболее популярным и чаще устанавливаемым программным обеспечением была Novell Netware версии 3.*x*. Конечно, были выпущены Novell Netware версий 4 и 5, однако из-за просчетов компании и других факторов (в частности увеличения популярности Windows NT и Linux) рыночная доля Novell Netware стремительно сократилась. Тем не менее, во многих крупных учреждениях сохранились серверы, использующие Novell Netware. Поэтому мы должны уметь с ними взаимодействовать.

Термины, используемые в тексте

Для лучшего понимания текста приведем расшифровку некоторых используемых терминов. Поскольку идеология IPX несколько отличается от TCP/IP, некоторые термины могут оказаться незнакомыми для широкой аудитории.

- 802.2 — это протокол I.E.E.E., определяющий набор процедур управления логическими связями (Logical Link Control). Он обеспечивает упрощенный способ сосуществования различных протоколов, однако является ограниченным.

- 802.3 — это протокол I.E.E.E., определяющий механизм множественного доступа к среде переноса с определением коллизий (Carrier Sense Multiple Access with Collision Detection, CSMA/CD). Базируется на оригинальном стандарте DIX Ethernet с некоторыми дополнениями. Важнейшее изменение — в кадре поле типа "идентификатор протокола" используется в качестве поля длины. IEEE 802.3 был спроектирован для того, чтобы переносить *только* фреймы IEEE 802.2, однако существуют реализации, которые используют этот тип для прямого переноса фреймов IPX.
- Bindery — специализированная база данных, сохраняющая сетевую конфигурационную информацию на файловом сервере Novell. Используется для получения информации о доступных серверах, маршрутизации и пользователях. Отдаленно напоминает сервер DNS.
- Ethernet_II — это упрощенная версия оригинального стандарта DIX Ethernet. Протокол часто используется в среде Novell Netware.
- Hardware address — адрес устройства. Число, уникально идентифицирующее хост в физической сети на уровне доступа к среде передачи данных.
- IPX (Internet Packet eXchange, Межсетевой обмен пакетами) — протокол, используемый корпорацией Novell для обеспечения межсетевой поддержки NetWare. Похож по функциональности на IP-протокол.
- Сеть IPX — это набор оборудования, подключенного к одному и тому же сегменту локальной сети, использующий один и тот же тип фрейма. Различные типы фреймов в одном и том же сегменте локальной сети считаются отдельными сетями.

  Каждой сети выделяется адрес, который должен быть уникальным во всей локальной сети. Клиентам IPX этот адрес выдается сервером при запуске.
- Сетевой адрес IPX — уникальное число, которое идентифицирует частную сеть IPX.
- Внутренняя сеть IPX — это виртуальная сеть IPX. Используется для обеспечения уникальной идентификации хоста IPX. Применяется для хостов IPX, которые существуют больше чем в одной физической сети IPX (файловые серверы, маршрутизаторы).
- NCP (NetWare Core Protocol, базовый протокол NetWare) — протокол сетевой файловой системы Novell NetWare. NCP по функциональности похож на NFS.
- RIP (Routing Information Protocol, протокол маршрутной информации) — протокол, используемый для автоматического распространения сетевых маршрутов в сетях IPX.
- Route (маршрут) — путь прохождения пакета для достижения хоста назначения.

- SAP (Service Advertisement Protocol, Протокол объявления сервисов) — протокол, который используется для объявления сетевых сервисов в Novell NetWare.
- SNAP (Sub Network Access Protocol, Протокол доступа к подсетям) — спроектирован для использования поверх протоколов 802.3 и 802.2.

Linux и IPX

В этом разделе мы рассмотрим три варианта настройки Linux-системы:

- IPX-клиент;
- IPX-сервер;
- IPX-маршрутизатор.

Прежде чем приступить к настройке системы, необходимо убедиться, что ядро вашей операционной системы Linux скомпилировано с поддержкой протокола IPX.

Файлы в /proc, относящиеся к IPX

Существует несколько файлов, тем или иным образом касающихся поддержки IPX в Linux, которые располагаются в каталоге /proc:

- /proc/net/ipx_interface — этот файл содержит информацию о существующих интерфейсах IPX в вашей системе;
- /proc/net/ipx_route — этот файл содержит список маршрутов, существующих в таблице маршрутов IPX;
- /proc/net/ipx — этот файл содержит список сокетов IPX, которые открыты для использования на вашем компьютере.

Linux-утилиты IPX

Помимо пакета Mars_nwe, есть несколько утилит, позволяющих сконфигурировать поддержку IPX-протокола в операционной системе Linux.

- `ipx_interface` — эта команда используется для добавления, удаления или проверки IPX на существующем сетевом устройстве. Обычно сетевым устройством является устройство Ethernet. Например:

```
ipx_interface add -p eth0 802.2 x39ab0222
```

- `ipx_configure` — эта команда разрешает или запрещает автоматическую установку конфигурации интерфейсов и первичного интерфейса. Например:

```
ipx_configure --auto_interface=on --auto_primary=on
```

- `ipx_internal_net` — эта команда позволяет настраивать адрес внутренней сети. Например:

```
ipx_internal_net add 0xab000000 1
```

- `ipx_route` — эта команда позволяет вручную модифицировать таблицу маршрутизации IPX. Например:

```
ipx_route add 0x39ab0222 0x39ab0108 0x00608CC33C0F
```

IPX-клиент

Существует пакет ncpfs, который позволяет Linux эмулировать обычную рабочую станцию Novell для файловых сервисов. В этот пакет также входит утилита печати, которая позволяет использовать принт-сервер Novell. Пакет ncpfs предназначен для работы с файловыми серверами Novell версий 3.*x*. Для использования ncpfs с файловыми серверами Novell 4.*x* файловый сервер должен работать в режиме эмуляции bindery.

Настройка сетевого программного обеспечения IPX

Существует два способа настройки сетевого программного обеспечения IPX:

- вы можете вручную настроить всю информацию о вашей сети IPX;
- можно также позволить программному обеспечению определить для себя некие разумные установки с помощью команды:

```
ipx_configure --auto_interface=on --auto_primary=on
```

Проверка конфигурации

После того как ваша сеть IPX настроена, воспользуйтесь командой `slist`, для того чтобы увидеть список всех файловых серверов Novell в вашей сети.

Монтирование сервера или тома Novell

Для того чтобы смонтировать том файлового сервера Novell в файловую систему Linux, существует команда `ncpmount`. Для демонтирования смонтированных файловых систем Nowell используется команда `ncpumount`.

Посылка сообщения пользователю Nowell

Для посылки сообщений пользователям Novell можно воспользоваться утилитой nsend. Пример:

```
nsend rod hello
```

посылает сообщение "hello" пользователю, вошедшему под именем "rod".

IPX-сервер

Существует, по меньшей мере, два пакета, которые позволяют операционной системе Linux выступать в качестве файлового сервера Novell — это Mars_nwe и Lwared. Эти пакеты позволяют осуществлять доступ к файлам на Linux-системе для пользователей, использующих клиентское программное обеспечение Novell NetWare.

Пакет mars_nwe

Разработан Martin Stover для обеспечения в операционной системе Linux работы файловых сервисов и сервисов печати для клиентов NetWare. Mars_nwe реализует подмножество полного Novell NCP для файловых сервисов, основанного на bindery и сервисах печати.

Настройка сервера

Необходимо отредактировать файл /etc/nwserv.conf. Файл состоит из текстовых строк. Каждая строка разделена пробелами и начинается с числа, которое обозначает содержимое этой строки. Все символы, следующие за символом '#', считаются комментарием и игнорируются. В комплекте mars_new есть пример настроечного файла.

Пример конфигурационного файла:

```
# ТОМА (максимум 5)
# Только том SYS является необходимым. Каталог, содержащий том SYS,
# должен содержать каталоги: LOGIN, PUBLIC, SYSTEM, MAIL.
# Опция 'i' регистр букв.
# Опция 'k' преобразует все имена в запросе NCP в нижний регистр
# Опция 'm' обозначает том как сменный
# Опция 'r' устанавливает том только для чтения
# Опция 'o' показывает, что том является единой файловой системой
# Опция 'P' разрешает командам использоваться как файлы
# Опция 'O' позволяет использовать пространство имен OS/2
# Опция 'N' разрешает использование пространства имен NFS
# По умолчанию в верхнем регистре.
# Синтаксис:
#    1 <Имя тома>  <Путь к тому>        <Опции>

1   SYS          /home/netware/SYS/               # SYS
1   DATA         /home/netware/DATA/      k       # DATA
1   CDROM        /cdrom                   kmr     # CDROM
```

```
# ИМЯ СЕРВЕРА
# Если не установлено, тогда имя машины linux hostname будет
# конвертировано в верхний регистр. Этот пункт
# является опциональным, если не настроено — будет использовано имя
# машины.
# Синтаксис:
#    2 <Имя сервера>

2   LINUX_FS01

# АДРЕС ВНУТРЕННЕЙ СЕТИ
# Адрес внутренней сети IPX — это свойство, которое упрощает
# маршрутизацию IPX для многосетевых машин
# Синтаксис:
#    3 <Адрес внутренней сети> [<Номер узла>]
# или:
#    3 auto
#
# Если вы используете 'auto', тогда будет использован IP-адрес
# вашей машины.
3   0x49a01010  1



# СЕТЕВОЕ УСТРОЙСТВО(А)
# Этот раздел настраивает вашу сеть IPX. Если она у вас уже
# настроена, вам этот пункт не нужен. Это тоже самое, что и
# использование утилит ipx_configure/ipx_interface до запуска
# сервера.
# Синтаксис:
#    4 <Номер сети IPX> <имя устройства> <тип фрейма> [<ticks>]
#                          Frame types: ethernet_ii, 802.2, 802.3, SNAP

4  0x39a01010  eth0  802.3  1

# СОХРАНЯТЬ МАРШРУТЫ IPX ПОСЛЕ ОКОНЧАНИЯ РАБОТЫ СЕРВЕРА
# Синтаксис:
#    5 <флаг>
#       0 = не сохранять маршруты, 1 = сохранять маршруты
```

```
5 0

# ВЕРСИЯ NETWARE
# Синтаксис:
#    6 <версия>
#      0 = 2.15, 1 = 3.11

6 1

# ОБРАБОТКА ПАРОЛЯ
# Настоящие клиенты Novell для DOS поддерживают процедуру, которая
# шифрует пароли при их изменении. Вы можете выбрать, хотите ли вы,
# чтобы ваш сервер поддерживал эту процедуру или нет.
# Синтаксис:
#    7 <флаг>
#    <флаг> может быть:
#        0 force password encryption. (Клиенты не могут сменить пароль)
#        1 force password encryption, разрешить изменение нешифрованного
#        пароля
#        7 разрешаются нешифрованные пароли, но не пустые
#        8 разрешаются нешифрованные пароли, включая пустые
#        9 полностью нешифрованные пароли (не работает с OS/2)

7 1

# МИНИМАЛЬНЫЕ ПРАВА GID UID
# разрешения, используемые для подсоединения без входа. Эти разрешения
# будут использоваться для файлов на присоединении к вашему
# основному серверу.
# Синтаксис:
#    10 <gid>
#    11 <uid>
#    <gid> <uid> из /etc/passwd, /etc/groups

10  200
11  201

# ПАРОЛЬ АДМИНИСТРАТОРА (SUPERVISOR)
# Может быть убран после первого запуска сервера. Сервер зашифрует
```

```
# эту информацию в файл bindery после запуска. Вы должны избегать
# использования пользователя 'root' и вместо этого использовать
# другой идентификатор для администрирования файлового сервера mars.
#
# Эта запись читается и шифруется в файлы bindery сервера, так что она
# необходима только при первом запуске сервера, чтобы обеспечить
# безопасность пароля.
#
# Синтаксис:
#    12 <Идентификатор администратора> <имя пользователя Unix> [<пароль>]

12  SUPERVISOR  terry  secret

# ЗАПИСИ ПОЛЬЗОВАТЕЛЕЙ
# Этот раздел ассоциирует идентификаторы NetWare с идентификаторами
# пользователей UNIX. Наличие пароля является опциональным.
# Синтаксис:
     13 <Идентификатор пользователя> <имя пользователя в Unix> [<пароль>]

13  MARTIN martin
13  TERRY  terry

# НАСТРОЙКА СИСТЕМЫ "ЛЕНИВОГО" АДМИНИСТРИРОВАНИЯ
# Если у вас большое количество пользователей, и вы не хотите
# беспокоиться использованием индивидуального сопоставления
# пользовательских имен  между Linux_системой
#  и Mars_nwe, то вы можете
# автоматически  сопоставить идентификаторы mars_nwe в имена
# пользователей Linux. Но в настоящее время нет способа использовать
# пароли linux, так что все пользователи, настроенные таким способом,
# будут пользоваться единственным паролем, указанным здесь.
# Рекомендуется не использовать это до тех пор, пока вас перестанет
# беспокоить безопасность.
# Синтаксис:
#    15 <флаг> <общий пароль>
#    <флаг>:    0  — не делать автоматическое мапирование пользователей
#               1  — автоматически мапировать пользователей, не указанных
#               выше
#               99 — автоматически мапировать всех пользователей этим
#               способом
```

```
15  0  duzzenmatta

# ПРОВЕРКА РАБОТОСПОСОБНОСТИ
# mars_nwe будет автоматически убеждаться, что определенные
# каталоги существуют, если установлен этот флаг
# Синтаксис:
#    16 <флаг>
#    <флаг> — 0 для нет, не делать, или 1 для да, делать проверку

16  0

# ОЧЕРЕДИ ПЕЧАТИ
# Этот раздел ассоциирует принтер NetWare с принтерами UNIX.
# Каталоги очередей должны быть созданы вручную до попытки печати.
# Каталоги очередей НЕ являются очередями lpd.
# Синтаксис:
#    21 <имя очереди> <каталог очереди> <команда печати unix>

21  EPSON  SYS:/PRINT/EPSON lpr -h
21  LASER  SYS:/PRINT/LASER lpr -Plaser

# ФЛАГИ ОТЛАДКИ
# Обычно они не нужны, но могут быть полезными, если вы ищете решение
# проблемы.
# Синтаксис:
#    <тема отладки> <флаг отладки>
#
#    100 = IPX KERNEL
#    101 = NWSERV
#    102 = NCPSERV
#    103 = NWCONN
#    104 = start NWCLIENT
#    105 = NWBIND
#    106 = NWROUTED
#                0 = запрещает отладку, 1 = разрешает отладку

100 0
101 0
102 0
```

```
103 0
104 0
105 0
106 0

# ЗАПУСК NWSERV В ФОНОВОМ РЕЖИМЕ И ИСПОЛЬЗОВАНИЕ ФАЙЛА ПРОТОКОЛА
# Синтаксис:
#    200 <флаг>
#        0 = запуск NWSERV в нормальном режиме и не использовать файл
#        протокола
#        1 = запуск NWSERV в фоновом режиме и использовать файл протокола

200  1

# ИМЯ ФАЙЛА ПРОТОКОЛА
# Синтаксис:
#    201 <файл протокола>

201  /tmp/nw.log

# ДОПОЛНЯТЬ ПРОТОКОЛ ИЛИ ПЕРЕЗАПИСЫВАТЬ
# Синтаксис:
#    202 <флаг>
#        0 = добавлять к существующему файлу протокола
#        1 = переписывать существующий файл протокола

202  1

# ВРЕМЯ ВЫКЛЮЧЕНИЯ СЕРВЕРА
# Этот раздел устанавливает время между выдачей команды SERVER DOWN и
# действительным выключением сервера.
# Синтаксис:
#    210 <время>
#        в секундах. (по умолчанию 10)

210  10

# ИНТЕРВАЛ МЕЖДУ  ШИРОКОВЕЩАТЕЛЬНЫМИ ПЕРЕДАЧАМИ МАРШРУТОВ
# Время в секундах между широковещательными передачами сервера.
```

```
# Синтаксис:
#    211 <время>
#        в секундах. (по умолчанию 60)

211  60

# ИНТЕРВАЛ ПРОТОКОЛИРОВАНИЯ МАРШРУТИЗАЦИИ
# Устанавливает сколько широковещательных передач произойдет до
# протоколирования маршрутизационной информации.
# Синтаксис:
#    300  <число>

300  5

# ФАЙЛ ПРОТОКОЛА МАРШРУТИЗАЦИИ
# Устанавливает имя файла протокола маршрутизации.
# Синтаксис:
#    301 <имя файла>

301  /tmp/nw.routes

# ДОБАВЛЕНИЕ/ПЕРЕЗАПИСЬ МАРШРУТНОЙ ИНФОРМАЦИИ
# Устанавливает, хотите ли вы добавлять информацию к существующему
# файлу протокола или перезаписывать его.
# Синтаксис:
#    302 <флаг>
#        <flag> — 0 для дополнения, 1 для создания/перезаписи

302  1

# WATCHDOG TIMING
# Устанавливает хронометраж для наблюдательных сообщений, чтобы
# убедиться, что сеть функционирует.
# Синтаксис:
#    310 <значение>
#        <значение> = 0 — всегда посылать наблюдательные сообщения
#                   < 0 — (-ve) для запрета наблюдений
#                   > 0 — посылать наблюдательные сообщения при
#                         падении трафика ниже 'n' ticks
```

```
310  7

# ФАЙЛ СТАНЦИЙ
# Устанавливает имя для файла станций, который определяет, для каких
# машин этот файловый сервер будет выступать как первичный файловый
# сервер. Синтаксис этого файла описан в каталоге 'examples' исходного
# кода пакета.
# Синтаксис:
#    400 <имя файла>

400  /etc/nwserv.stations

# ОБРАБОТКА 'GET NEAREST FILESERVER'
# Устанавливает, как будет обрабатываться запрос SAP 'Get Nearest
# Fileserver' (получить ближайший файловый сервер).
# Синтаксис:
#    401 <флаг>
#        <флаг>: 0 — запретить запросы 'Get Nearest Fileserver'.
#                1 — файл 'stations' перечисляет исключаемые станции
#                2 — файл 'stations' перечисляет включаемые станции

401  2
```

Для запуска сервера достаточно выполнить команду `nwserv`.

Пакет lwared

Разработан Ales Dryak. Этот пакет позволяет системе Linux функционировать в качестве файлового сервера Nowell.

Сервер lwared обеспечивает подмножество всех функций Novell NCP. Он включает функции сообщений, но не обеспечивает возможности печати. Сервер lwared полагается на внешние программы для выполнения функций построения и обновления таблиц маршрутизации IPX и таблиц SAP.

Настройка и использование lwared

Сначала необходимо настроить интерфейсы Ethernet для поддержки сетей IPX, которые будет использовать ваш сервер. Для того чтобы сделать это, необходимо знать сетевые адреса IPX для каждого из сегментов локальной вычислительной сети (ЛВС), какие устройства Ethernet находятся в системе, какой тип фреймов (802.3, EtherII) использует каждый сегмент ЛВС и какой адрес внутренней сети должен использовать ваш сервер. Настройка для сер-

вера, который находится в двух непохожих сегментах с сетевыми адресами IPX, равными 23a9c300 и 23a9c301, и адресом внутренней сети bdefaced, может выглядеть так:

```
ipx_internal_net add BDEFACED 1
ipx_interface add eth0 802.3 23a9c300
ipx_interface add eth1 etherii 23a9c301
```

Для управления таблицей маршрутизации используются два демона, входящие в комплект lwared:

- `ipxripd` — управляет маршрутизационной информацией IPX;
- `ipxsapd` — управляет информацией SAP.

Для конфигурирования сервера lwared необходимо сконфигурировать следующие файлы:

- /etc/lwpasswd — в этом файле хранится информация о пользователях сервера lwared. Для работы с записями в этом файле используется программа lwpasswd.

  Файл /etc/lwpasswd содержит текстовые строки, каждая из них идентифицирует пользователя и его пароль в зашифрованном виде. Отсутствие шифрованного пароля разрешает вход без пароля. Пользователи lwared должны быть также зарегистрированы в операционной системе Linux;
- /etc/lwvtab — этот файл содержит таблицу томов lwared и хранит информацию о доступных для сетевых клиентов каталогов сервера.

  Формат файла очень прост — после имени тома через пробел следует экспортируемый каталог Linux. Вы должны иметь по крайней мере запись для тома SYS, чтобы запустить сервер.

Для запуска сервера lwared достаточно выполнить команду `lwared`.

IPX-маршрутизатор

Маршрутизатор используется для того, чтобы пересылать информацию из одной локальной сети в другую. Для сети на базе Novell Netware есть два вида информации, которые необходимо распространять по сети для ее нормального функционирования. Это информация о сетевых маршрутах, использующая Novell RIP, и информация о сервисах, использующая Novell SAP. Поэтому маршрутизатор должен поддерживать оба этих протокола.

Для нормального функционирования IPX-маршрутизатора Linux необходимы программы, реализующие Novell RIP и SAP, обеспечивающие правильность построения таблицы маршрутизации IPX и ее периодическое обновление для отражения изменений в сети.

Существует по крайней мере два способа создать IPX-маршрутизатор:

- можно использовать демон маршрутизации ipxripd;
- в состав пакета mars_nwe входит свой демон маршрутизации.

Для настройки системы в качестве IPX-маршрутизатора необходимо выполнить следующие условия:

- ядро должно быть скомпилировано с поддержкой IPX и Ethernet;
- необходимо установить программу ipxd;
- включить протокол IPX на каждом сетевом интерфейсе, используя команду `ipx_interface`;
- запустить программу демона ipxd.

Пример:

```
# ipx_interface add eth0 802.2 0x0100000000
# ipx_interface add eth1 802.2 0x0200000000
# ipx_interface add eth2 etherii 0x0300000000
# ipxd
```

Для проверки работоспособности маршрутизации проверьте файл /proc/net/ipx_route. В этом файле вы должны увидеть маршруты IPX, относящиеся к вашей конфигурации.

Настройка Linux как клиента печати сервера Novell

Пакет ncpfs содержит две программы, которые позволяют производить печать из Linux-системы на принтер, подключенный к серверу печати Novell. Команда `nprint` позволяет печатать файл в очередь печати NetWare. Команда `pqlist` позволяет выводить список доступных очередей печати на сервере NetWare.

Обе команды требуют указать имя пользователя и пароль.

Пример:

```
pqlist -S ACCT_FS01 -U guest -n
nprint -S ACCT_FS01 -q LASER -U guest -n filename.txt
```

Синтаксис команд похож на синтаксис команды `ncpmount`.

Настройка Linux как сервера печати Novell

Программа pserver, которая позволяет Linux выступать в качестве сервера печати в сети Netware, входит в пакет ncpfs. Альтернативная поддержка включена в пакет mars_nwe.

Когда у вас на сервере настроены принтеры и установлена утилита pserver, необходимо добавить команды ее запуска в rc-файл.

Простейший вариант приведен ниже:

```
pserver -S ACCT_01 -U LASER -P secret -q LASERJET
```

Эта команда предписывает утилите pserver войти на файловый сервер ACCT_01 с именем пользователя LASER и паролем `secret` и брать задания из очереди печати `LASERJET`. Когда входящее задание печати будет переслано, то начнет действовать команда печати по умолчанию `lpr` для переноса задания печати на демон печати Linux. Очередь печати должна быть уже определена на файловом сервере, и пользователь должен иметь привилегии оператора для этой очереди.

Команды пользователя и администрирования ncpfs

В пакет ncpfs входит набор пользовательских и административных команд.

Команды пользователя

В качестве пользовательских используются следующие команды:

- `ncopy` (Network Copy) — позволяет копировать файлы, используя функцию копирования Netware вместо копирования по сети;
- `nprint` (Network Print) — позволяет печатать файл в очередь печати на сервере Netware;
- `nsend` (Network Send) — позволяет послать сообщение другим пользователям на сервере Netware;
- `nwbols` (List Bindery Objects) — позволяет вам увидеть содержимое bindery на сервере Netware;
- `Nwboprops` (List Properties of a Bindery Object) — позволяет просмотреть свойства объекта bindery Netware;
- `nwbpset` (Set Bindery Property) — позволяет установить свойства объекта bindery Netware;
- `nwbpvalues` (Print Netware Bindery Objects Property Contents) — позволяет напечатать содержимое свойства bindery Netware;
- `nwfsinfo` (Fileserver Information) — печатает общую информацию о сервере Netware;
- `nwpasswd` (Netware Password) — позволяет сменить пароль пользователя Netware;

- `nwrights` (Netware Rights) — показывает список прав, ассоциированных с отдельным файлом или каталогом;
- `nwuserlist` (Userlist) — перечисляет пользователей, подключенных к файловому серверу Netware;
- `pqlist` (Print Queue List) — показывает содержимое очереди печати Netware;
- `Slist` (Server List) — показывает список известных серверов Netware.

Утилиты администрирования

В качестве утилит администрирования используются следующие команды:

- `nwbocreate` (Create a Bindery Object) — позволяет создать объект bindery Netware;
- `nwborm` (Remove Bindery Object) — позволяет удалить объект bindery Netware;
- `nwbpadd` (Add Bindery Property) — позволяет установить значение существующего свойства объекта bindery Netware;
- `nwbpcreate` (Create Bindery Property) — позволяет создать новое свойство для существующего объекта bindery Netware;
- `nwbprm` (Remove Bindery Property) — позволяет удалить свойство из объекта bindery Netware;
- `nwgrant` (Grant Trustee Rights) — позволяет установить попечительские права на каталог на файловом сервере Netware;
- `nwrevoke` (Revoke Trustee Rights) — позволяет удалить попечительские права с каталога на файловом сервере Netware.

Тунелирование IPX через IP

В том случае, если у вас две локальных сети Novell, между которыми есть только IP-сеть, и вам необходимо каким-либо образом соединить две эти сети — воспользуйтесь пакетом ipxtunnel.

Пакет ipxtunnel позволяет пакетам IPX быть включенными в пакеты TCP/IP так, что они могут без потерь информации переноситься TCP/IP-соединением. Для нормального функционирования необходимо сконфигурировать и запустить пакет ipxtunnel на обоих концах туннеля.

Настройка

Настроить ipxtunnel не составляет труда. Пусть один конец туннеля (компьютер) называется q.odessa.ua, а второй компьютер — w.odessa.ua. Для конфигурации ipxtunnel используется файл /etc/ipxtunnel.conf. Этот файл по-

зволяет указать порт UDP по умолчанию для использования в соединении TCP/IP, куда посылать инкапсулированные данные, на каком локальном интерфейсе должен слушать ipxtunnel и на который отправлять пакеты IPX.

Пример конфигурационного файла:

```
#
# /etc/ipxtunnel.conf для q.odessa.ua
#
# Порт UDP для использования:       (по умолчанию 7666)
port 7777
#
# Удаленная машина, на которую отправлять пакеты IPX
remote w.odessa.ua
#
# Локальные интерфейсы, на которых искать пакеты IPX: (по умолчанию eth0)
interface eth0
interface eth1
```

Другой компьютер должен иметь похожий конфигурационный файл.

Ссылки

- **www.compu-art.de/mars_nwe/** — домашняя страница mars_new.
- **www.osp.ru/pcworld/1998/05/44.htm** — А. Суханов, В. Хименко. Linux и Windows 95: эффективность совместной работы. Мир ПК №5/98.
- Соответствующие HOWTO *(см. гл. 13)*:
  - IPX HOWTO — настройка IPX-протокола;
  - Ethernet HOWTO — все, что касается Ethernet-устройств.

Глава 26

Управление процессами

Данная глава посвящена процессам операционной системы Linux. Поскольку администрирование операционной системы в конечном счете сводится к управлению процессами, вполне логично выделить отдельную главу на описание столь важной темы.

Каждый раз, когда вы запускаете на выполнение программу, вы начинаете то, что в литературе именуется как *процесс*. Или другими словами — процессом называется выполняемая в данный момент программа или ее потомки. Каждый процесс запускается от имени какого-то пользователя. Процессы, которые стартовали при загрузке, обычно выполняются от имени пользователей root или nobody.

Каждый пользователь может управлять поведением процессов, им запущенных. При этом пользователь root может управлять всеми процессами — как запущенными от его имени, так и процессами, порожденными другими пользователями операционной системы. Управление процессами осуществляется с помощью утилит, а также при помощи некоторых команд командной оболочки (shell).

Каждый процесс в системе имеет уникальный номер — идентификационный номер процесса (Process Identification, PID). Этот номер используется ядром операционной системы, а также некоторыми утилитами для управления процессами.

Выполнение процесса на переднем плане и в фоновом режиме

Процессы могут выполняться на *переднем плане* (foreground) — режим по умолчанию и в *фоновом режиме* (background). На переднем плане в каждый момент для текущего терминала может выполняться только один процесс. Однако пользователь может перейти в другой виртуальный терминал и запустить на выполнение еще один процесс, а на другом терминале еще один и т. д. Процесс переднего плана — это процесс, с которым вы взаимодейст-

вуете, он получает информацию с клавиатуры (стандартный ввод) и посылает результаты на ваш экран (стандартный вывод).

Фоновый процесс после своего запуска благодаря использованию специальной команды командной оболочки отключается от клавиатуры и экрана (то есть не ожидает ввода данных со стандартного ввода и не выводит информацию на стандартный вывод), а командная оболочка не ожидает окончания запущенного процесса, что позволяет пользователю немедленно запустить еще один процесс.

Обычно фоновые процессы требуют очень большого времени для своего завершения и не требуют вмешательства пользователя во время существования процесса. К примеру, компиляция программ или архивирование большого объема информации — кандидаты номер один для перевода процесса в фоновый режим.

Процессы так же могут быть *отложенными*. Отложенный процесс — это процесс, который в данный момент не выполняется и временно остановлен. После того как вы остановили процесс, в дальнейшем вы можете его продолжить как на переднем плане, так и в фоновом режиме. Возобновление приостановленного процесса не изменит его состояния — при возобновлении он начнется с того места, на котором был приостановлен.

Для выполнения программы в режиме переднего плана достаточно просто набрать имя программы в командной строке и запустить ее на выполнение. После этого вы можете работать с программой.

Для запуска программы в качестве фонового процесса достаточно набрать в командной строке имя программы и в конце добавить знак амперсанта (`&`), отделенный пробелом от имени программы и ее параметров командной строки, если таковые имеются. Затем программа запускается на выполнение. В отличие от запуска программы в режиме переднего плана мы получим приблизительно следующее сообщение:

```
/home/vasya# yes > /dev/null &;
[1] 123
/home/vasya#
```

Оно состоит из двух чисел и приглашения командной строки. Таким образом, мы запустили программу выполняться в фоновом режиме и получили возможность запустить с той же самой консоли на выполнение еще какую-то программу.

Число `[1]` означает номер запущенного нами фонового процесса. Как вы узнаете несколько позже, с его помощью можно будет производить манипуляции с нашим фоновым процессом. Значение `123` показывает идентификационный номер (PID) нашего процесса. Отличия этих двух чисел достаточно существенные. Номер фонового процесса уникален *только* для пользователя, запускающего данный фоновый процесс. То есть если у нас в системе

три пользователя решили запустить фоновый процесс (первый для текущего сеанса) — в результате у каждого пользователя появится фоновый процесс с номером [1]. Напротив, идентификационный номер процесса (PID) уникален для всей операционной системы и однозначно идентифицирует в ней каждый процесс. Спрашивается, для чего тогда вводить нумерацию фонового процесса для пользователя? Для удобства. Номер фонового процесса хранится в переменных командной оболочки пользователя и позволяет не забивать голову цифрами типа 2693 или 1294, а использовать переменные вида %1, %2. Однако допускается пользоваться и идентификационным номером процесса.

Для проверки состояния фоновых процессов можно воспользоваться командой командной оболочки — jobs.

```
/home/vasya# jobs
[1]+  Running                 yes >/dev/null  &
/home/vasya#
```

Из вышеприведенного листинга видно, что у пользователя в данный момент запущен один фоновый процесс, и он выполняется.

Остановка и возобновление процесса

Помимо прямого указания выполнять программу в фоновом режиме, существует еще один способ перевести процесс в фоновый режим. Для этого мы должны выполнить следующие действия:

1. Запустить процесс выполняться на переднем плане.
2. Остановить выполнение процесса.
3. Продолжить процесс в фоновом режиме.

Для выполнения программы введем ее имя в командной строке и запустим на выполнение. Для остановки выполнения программы необходимо нажать на клавиатуре следующую комбинацию клавиш — <Ctrl>+<Z>. После этого вы увидите на экране следующее:

```
/home/vasya# yes > /dev/null
ctrl+Z
[1]+  Stopped                 yes >/dev/null
/home/vasya#
```

Мы получили приглашение командной строки. Для того чтобы перевести выполнение процесса в фоновый режим, необходимо выполнить следующую команду:

```
bg %1
```

Причем необязательно делать это сразу после остановки процесса, главное правильно указать номер остановленного процесса.

Для того чтобы вернуть процесс из в фонового режима выполнения на передний план, достаточно выполнить следующую команду:

```
fg %1
```

В том случае, если вы хотите перевести программу в фоновый или, наоборот, на передний план выполнения сразу после остановки процесса, можно выполнить соответствующую программу *без* указания номера остановленного процесса.

Существует большая разница между фоновым и остановленным процессом. Остановленный процесс не выполняется и не потребляет ресурсы процесса, однако занимает оперативную память или пространство свопинга. В фоновом же режиме процесс продолжает выполняться.

Как остановить выполнение фонового процесса? Использование комбинации клавиш <Ctrl>+<Z> не поможет, поскольку процесс находится в фоновом режиме и не реагирует на ввод данных с консоли. Для решения этой проблемы следует переместить процесс на передний план, а затем остановить.

Завершение работы процесса

Ну вот, вы научились запускать и останавливать выполнение процессов, а также переводить исполняемый процесс в фоновый режим и в режим переднего плана. Однако вы, возможно, не умеете завершать работу процесса.

- Вариант первый. Если процесс интерактивный, как правило, в документации или прямо на экране написано, как корректно завершить программу.
- Вариант второй. В том случае, если вы не знаете, как завершить текущий процесс (не фоновый), можно воспользоваться клавиатурной комбинацией <Ctrl>+<C>. Попробуйте также комбинацию клавиш <Ctrl>+<Break>. А для остановки фонового процесса можно перевести его на передний план, а затем уже воспользоваться вышеприведенными клавиатурными комбинациями.
- Вариант третий и самый действенный. В том случае, если вам не удалось прекратить выполнение процесса вышеприведенными способами — например программа зависла или "слетел" терминал — для завершения процесса можно воспользоваться следующими командами: `kill`, `killall`.

Команда `kill` может получать в качестве аргумента как номер процесса, так и идентификационный номер (PID) процесса. Таким образом, команда

```
/home/vasya# kill 123
```

эквивалентна команде

```
/home/vasya# kill %1
```

Можно видеть, что не надо использовать "%", когда вы обращаетесь к работе по идентификационному номеру (PID) процесса.

С помощью команды `killall` можно прекратить выполнение нескольких процессов сразу, имеющих одно и то же имя. Например, команда `killall mc` прекратит работу всех программ mc, запущенных от имени данного пользователя.

Для того чтобы завершить работу процесса, вам надо быть его владельцем. Это сделано в целях безопасности. Если бы одни пользователи могли завершать процессы других пользователей, открылась бы возможность исполнения в системе множества злонамеренных действий. Пользователь root может завершить работу любого процесса в операционной системе.

Программы, используемые для управления процессами

Существует достаточно большое количество утилит, используемых для управления тем или иным способом процессами, исполняемыми в операционной системе. Здесь мы рассмотрим только основные такие утилиты. В табл. 26.1 приведен список основных программ, тем или иным образом предназначенных для управления процессами.

Таблица 26.1. *Программы управления процессами*

Программа	Описание
at	Выполняет команды в определенное время
batch	Выполняет команды тогда, когда это позволяет загрузка системы
cron	Выполняет команды по заранее заданному расписанию
crontab	Позволяет работать с файлами crontab отдельных пользователей
kill	Прекращает выполнение процесса
nice	Изменяет приоритет процесса перед его запуском
nohup	Позволяет работать процессу после выхода пользователя из системы
ps	Выводит информацию о процессах
renice	Изменяет приоритет работающего процесса
w	Показывает, кто в настоящий момент работает в системе и с какими программами

nohup

Эта утилита позволяет организовать фоновый процесс, продолжающий свою работу даже тогда, когда пользователь отключился от терминала, в отличие от команды &, которая этого не позволяет. Для организации фонового процесса необходимо выполнить следующую команду:

```
nohup выполняемая_фоновая_команда &
```

ps

Программа ps предназначена для получения информации о существующих в операционной системе процессах. У этой команды есть множество различных опций, но мы остановимся на самых часто используемых. Для получения подробной информации смотрите man-страницу этой программы.

Простой запуск ps без параметров выдаст список программ, выполняемых на терминале. Обычно этот список очень мал:

```
PID TTY      TIME CMD
885 tty1     00:00:00 login
893 tty1     00:00:00 bash
955 tty1     00:00:00 ps
```

Что означает полученная информация?

- Первый столбец — PID (идентификационный номер процесса). Как уже упоминалось, каждый выполняющийся процесс в системе получает уникальный идентификатор, с помощью которого производится управление процессом. Каждому вновь запускаемому на выполнение процессу присваивается следующий свободный PID. Когда процесс завершается, его номер освобождается. Когда достигнут максимальный PID, следующий свободный номер будет взят из наименьшего освобожденного.
- Следующий столбец — TTY — показывает, на каком терминале процесс выполняется. Запуск команды без параметров ps покажет процессы, выполняемые на текущем терминале.
- Столбец TIME показывает, сколько процессорного времени выполняется процесс. Оно не является фактическим временем с момента запуска процесса, поскольку Linux — это многозадачная операционная система. Информация, указанная в столбце TIME, показывает время, реально потраченное процессором на выполнение процесса.
- Столбец CMD показывает, что же это за программа. Отображается только имя программы, опции командной строки не выводится.

Для получения расширенного списка процессов, выполняемых в системе, используется следующая команда:

```
ps -ax
  PID TTY      STAT   TIME COMMAND
```

```
    1 ?        S      0:04 init
    2 ?        SW     0:00 [keventd]
    3 ?        SW     0:00 [kapm-idled]
    4 ?        SWN    0:00 [ksoftirqd_CPU0]
    5 ?        SW     0:00 [kswapd]
    6 ?        SW     0:00 [kreclaimd]
    7 ?        SW     0:00 [bdflush]
    8 ?        SW     0:00 [kupdated]
    9 ?        SW<    0:00 [mdrecoveryd]
   13 ?        SW     0:00 [kjournald]
  437 ?        S      0:00 syslogd -m 0
  442 ?        S      0:00 klogd -2
  462 ?        S      0:00 portmap
  490 ?        S      0:00 rpc.statd
  647 ?        S      0:00 /usr/sbin/sshd
  704 ?        S      0:00 lpd Waiting
  732 ?        S      0:00 sendmail: accepting connections
  751 ?        S      0:00 gpm -t ps/2 -m /dev/mouse
  769 ?        S      0:00 crond
  835 ?        S      0:00 xfs -droppriv -daemon
  853 ?        S      0:00 anacron
  871 ?        S      0:00 /usr/sbin/atd
  885 tty1     S      0:00 login -- root
  886 tty2     S      0:00 /sbin/mingetty tty2
  887 tty3     S      0:00 /sbin/mingetty tty3
  888 tty4     S      0:00 /sbin/mingetty tty4
  889 tty5     S      0:00 /sbin/mingetty tty5
  890 tty6     S      0:00 /sbin/mingetty tty6
  893 tty1     S      0:00 -bash
 1037 tty1     R      0:00 /usr/bin/mc -P
 1038 ?        S      0:00 cons.saver /dev/tty1
 1039 pts/0    S      0:00 bash -rcfile .bashrc
 1067 pts/0    R      0:00 ps -ax
```

Как можно видеть, список запущенных процессов в системе велик и достаточно сильно зависит от конфигурации операционной системы. Опции, заданные программе в этом примере, заставляют ее выводить не только имена программ, но и список опций, с которыми были запущены программы.

Появился новый столбец — `STAT`. В этом столбце отображается состояние (status) процесса. Полный список состояний вы можете прочитать в описании программы ps, а пока — самые важные состояния:

- буква `R` обозначает запущенный процесс, исполняющийся в данный момент времени;
- буква `S` обозначает спящий (sleeping) процесс — процесс ожидает какое-то событие, необходимое для его активизации;
- буква `Z` используется для обозначения "зомбированных" процессов (zombied) — это процессы, родительский процесс которых прекратил свое существование, оставив дочерние процессы рабочими.

Помимо этого позвольте обратить ваше внимание на колонку `TTY`. Как вы, наверное, заметили, многие процессы, расположенные в верхней части таблицы, в этой колонке содержат знак "`?`" вместо терминала. Так обозначаются процессы, запущенные с более не активного терминала. Как правило, это всякие системные сервисы.

Если вы хотите увидеть еще больше информации о выполняемых процессах, попробуйте выполнить команду:

```
ps -aux
USER     PID %CPU %MEM    VSZ   RSS TTY    STAT START TIME COMMAND
root       1  1.2  0.2   1412   520 ?      S    14:51 0:04 init
root       2  0.0  0.0      0     0 ?      SW   14:51 0:00 [keventd]
root       3  0.0  0.0      0     0 ?      SW   14:51 0:00 [kapm-idled]
root       4  0.0  0.0      0     0 ?      SWN  14:51 0:00 [ksoftirqd_CPU0]
root       5  0.0  0.0      0     0 ?      SW   14:51 0:00 [kswapd]
root       6  0.0  0.0      0     0 ?      SW   14:51 0:00 [kreclaimd]
root       7  0.0  0.0      0     0 ?      SW   14:51 0:00 [bdflush]
root       8  0.0  0.0      0     0 ?      SW   14:51 0:00 [kupdated]
root       9  0.0  0.0      0     0 ?      SW<  14:51 0:00 [mdrecoveryd]
root      13  0.0  0.0      0     0 ?      SW   14:51 0:00 [kjournald]
root     437  0.0  0.2   1472   592 ?      S    14:52 0:00 syslogd -m 0
root     442  0.0  0.4   1928  1040 ?      S    14:52 0:00 klogd -2
rpc      462  0.0  0.2   1552   588 ?      S    14:52 0:00 portmap
rpcuser  490  0.0  0.2   1596   756 ?      S    14:52 0:00 rpc.statd
root     590  0.0  0.2   1396   524 ?      S    14:52 0:00 /usr/sbin/apmd -p
root     647  0.0  0.4   2676  1268 ?      S    14:52 0:00 /usr/sbin/sshd
root     680  0.0  0.3   2264   992 ?      S    14:52 0:00 xinetd -stayalive
lp       704  0.0  0.3   2600  1020 ?      S    14:52 0:00 lpd Waiting
root     732  0.0  0.7   5296  1984 ?      S    14:52 0:00 sendmail: accepti
root     751  0.0  0.1   1440   492 ?      S    14:52 0:00 gpm -t ps/2 -m /d
```

```
root     769  0.0  0.2  1584  660 ?     S    14:52 0:00 crond
xfs      835  0.0  1.4  4988 3612 ?     S    14:52 0:00 xfs -droppriv -da
root     853  0.0  0.2  1416  600 ?     S    14:52 0:00 anacron
daemon   871  0.0  0.2  1444  568 ?     S    14:52 0:00 /usr/sbin/atd
root     885  0.0  0.4  2320 1076 tty1  S    14:52 0:00 login -- root
root     886  0.0  0.1  1384  448 tty2  S    14:52 0:00 /sbin/mingetty tt
root     887  0.0  0.1  1384  448 tty3  S    14:52 0:00 /sbin/mingetty tt
root     893  0.0  0.5  2464 1312 tty1  S    14:52 0:00 -bash
root    1037  0.0  0.7  3284 1804 tty1  R    14:56 0:00 /usr/bin/mc -P
root    1038  0.0  0.1  1380  348 ?     S    14:56 0:00 cons.saver /dev/t
root    1039  0.0  0.5  2552 1392 pts/0 S    14:56 0:00 bash -rcfile .bas
root    1068  0.0  0.3  2780  824 pts/0 R    14:57 0:00 ps -aux
```

Как вы видите — информации прибавилось. Появились еще следующие столбцы:

- `USER` — показывает, от имени какого пользователя был запущен данный процесс;
- `%CPU`, `%MEM` — показывают, сколько данный процесс занимает соответственно процессорного времени и объем используемой оперативной памяти;
- `TIME` — время запуска программы.

В табл. 26.2 приведены некоторые параметры командной строки программы ps.

Таблица 26.2. *Параметры командной строки программы ps*

Ключ	Описание
a	Показать процессы всех пользователей
c	Имя команды из переменной среды
e	Показать окружение
f	Показать процессы и подпроцессы
h	Вывод без заголовка
j	Формат заданий
l	"Длинный" формат вывода
m	Вывод информации о памяти
n	Числовой вывод информации
r	Только работающие процессы
s	Формат сигналов

Таблица 26.2 *(окончание)*

Ключ	Описание
S	Добавить время использования процессора порожденными процессами
txx	Только процессы, связанные с терминалом xx
u	Формат вывода с указанием пользователя
v	Формат виртуальной памяти
w	Вывод без обрезки информации для размещения в одной строке
x	Показать процессы без контролирующего терминала

Программа ps обладает достаточно большим списком возможностей, ключей запуска и выводимой информацией, однако для обычной работы будет достаточно и вышеприведенной информации.

top

Еще одна утилита, с помощью которой можно получать информацию о запущенных в операционной системе процессах. Для использования достаточно просто запустить команду top на выполнение. Эта утилита выводит на экран список процессов в системе, отсортированных в порядке убывания значений используемых ресурсов.

```
2:55pm  up 3 min,  1 user,  load average: 0,06, 0,09, 0,03
32 processes: 31 sleeping, 1 running, 0 zombie, 0 stopped
CPU states:  1,1% user,  2,9% system,  0,0% nice, 95,8% idle
Mem:   255532K av,   42856K used,  212676K free,    0K shrd,   8560K buff
Swap:  257000K av,       0K used,  257000K free             19920K cached

  PID USER   PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM  TIME COMMAND
    1 root     8   0   520  520   452 S     0,0  0,2  0:04 init
    2 root     9   0     0    0     0 SW    0,0  0,0  0:00 keventd
    3 root     9   0     0    0     0 SW    0,0  0,0  0:00 kapm-idled
    4 root    19  19     0    0     0 SWN   0,0  0,0  0:00 ksoftirqd_CPU0
    5 root     9   0     0    0     0 SW    0,0  0,0  0:00 kswapd
    6 root     9   0     0    0     0 SW    0,0  0,0  0:00 kreclaimd
    7 root     9   0     0    0     0 SW    0,0  0,0  0:00 bdflush
    8 root     9   0     0    0     0 SW    0,0  0,0  0:00 kupdated
    9 root    -1 -20     0    0     0 SW<   0,0  0,0  0:00 mdrecoveryd
   13 root     9   0     0    0     0 SW    0,0  0,0  0:00 kjournald
```

```
437 root     9   0   592  592   496 S     0,0  0,2  0:00 syslogd
442 root     9   0  1040 1040   448 S     0,0  0,4  0:00 klogd
462 rpc      9   0   588  588   504 S     0,0  0,2  0:00 portmap
490 rpcuser  9   0   756  756   660 S     0,0  0,2  0:00 rpc.statd
590 root     8   0   524  524   464 S     0,0  0,2  0:00 apmd
647 root     9   0  1268 1268  1076 S     0,0  0,4  0:00 sshd
680 root     9   0  1008  992   816 S     0,0  0,3  0:00 xinetd
704 lp       9   0  1020 1020   872 S     0,0  0,3  0:00 lpd
```

Сначала идет общесистемная информация — из нее можно узнать время запуска операционной системы, время работы операционной системы от момента последнего перезапуска системы, количество зарегистрированных в данный момент в операционной системе пользователей, а также минимальную, максимальную и среднюю загрузку операционной системы. Помимо этого, отображается общее количество процессов и их состояние, сколько процентов ресурсов системы используют пользовательские процессы и системные процессы, использование оперативной памяти и свопа.

Далее идет таблица, во многом напоминающая вывод программы ps. Идентификационный номер процесса, имя пользователя — владельца процесса, приоритет процесса, размер процесса, его состояние, используемые процессом оперативная память и ресурс центрального процесса, время выполнения и наконец — имя процесса.

Утилита top после запуска периодически обновляет информацию о состоянии процессов в операционной системе, что позволяет нам динамически получать информацию о загрузке системы.

kill

Программа kill (в переводе с английского — убить) предназначена для посылки соответствующих сигналов указанному нами процессу. Как правило, это бывает тогда, когда некоторые процессы начинают вести себя неадекватно. Наиболее часто программа применяется, чтобы прекратить выполнение процессов.

Для того чтобы прекратить работу процесса, необходимо знать PID процесса либо его имя. Например, чтобы "убить" процесс 123, достаточно выполнить следующую команду:

```
kill 123
```

Как обычно, чтобы прекратить работу процесса, вам необходимо быть его владельцем. Само собой, пользователь root может прекратить работу любого процесса в системе.

Иногда обычное выполнение программы kill не справляется с поставленной задачей. Обычно это объясняется тем, что данный процесс завис либо выполняет операцию, которую с его точки зрения нельзя прервать немедленно. Для прерывания этого процесса можно воспользоваться следующей командой:

```
kill -9 123
```

Что это означает? Вообще-то программа kill предназначена для посылки процессам управляющих сигналов, одним из которых является сигнал SIGTERM (terminate, завершиться). Этот сигнал посылается процессу при выполнении программы kill по умолчанию. Процесс, получивший данный сигнал, должен корректно завершить свою работу (закрыть используемые файлы, сбросить буферы ввода/вывода и т. п.). Ключ -9 указывает программе kill посылать процессу другой тип сигнала — SIGKILL. Это приводит к тому, что процесс не производит корректного завершения, а немедленно прекращает свою жизнедеятельность. Помимо этих сигналов, в вашем распоряжении целый набор различных сигналов. Полный список сигналов можно получить, выполнив следующую команду:

```
kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR      31) SIGSYS      32) SIGRTMIN    33) SIGRTMIN+1
34) SIGRTMIN+2  35) SIGRTMIN+3  36) SIGRTMIN+4  37) SIGRTMIN+5
38) SIGRTMIN+6  39) SIGRTMIN+7  40) SIGRTMIN+8  41) SIGRTMIN+9
42) SIGRTMIN+10 43) SIGRTMIN+11 44) SIGRTMIN+12 45) SIGRTMIN+13
46) SIGRTMIN+14 47) SIGRTMIN+15 48) SIGRTMAX-15 49) SIGRTMAX-14
50) SIGRTMAX-13 51) SIGRTMAX-12 52) SIGRTMAX-11 53) SIGRTMAX-10
54) SIGRTMAX-9  55) SIGRTMAX-8  56) SIGRTMAX-7  57) SIGRTMAX-6
58) SIGRTMAX-5  59) SIGRTMAX-4  60) SIGRTMAX-3  61) SIGRTMAX-2
62) SIGRTMAX-1  63) SIGRTMAX
```

Как видите, список внушительный. Подробную информацию о сигналах вы найдете в документации на программу kill.

killall

Еще один вариант программы kill. Используется для того, чтобы завершить работу процессов, носящих одно и то же имя. К примеру, в нашей системе

запущено несколько программ mc. Для того чтобы одновременно завершить работу этих программ, достаточно всего лишь выполнить следующую команду:

```
killall mc
```

Конечно, этим не ограничивается использование данной команды. С ее помощью можно отсылать сигналы группе одноименных процессов. Для получения более подробной информации по этой команде обращайтесь к ее man-странице.

Изменение приоритета выполнения процессов

В операционной системе Linux у каждого процесса есть свой приоритет исполнения. Это очень удобно. Поскольку операционная система многозадачна — то для выполнения каждого процесса выделяется определенное количество времени. Для некоторых задач необходимо выделить побольше, для некоторых можно поменьше. Для этого и предназначен приоритет процесса. Управление приоритетом процесса осуществляется программами nice и renice.

nice

Программа nice позволяет запустить команду с предопределенным приоритетом выполнения, который задается в командной строке. При обычном запуске все задачи имеют один и тот же приоритет, и операционная система равномерно распределяет между ними процессорное время. Однако с помощью утилиты nice можно понизить приоритет какой-либо задачи, таким образом предоставляя другим процессам больше процессорного времени. Повысить приоритет той или иной задачи имеет право только пользователь root. Синтаксис использования nice следующий:

```
nice -number command
```

Уровень приоритета процесса определяется параметром *number*, при этом большее его значение означает меньший приоритет процесса. Значение по умолчанию — 10, и *number* представляет собой число, на которое должен быть уменьшен приоритет.

К примеру, процесс top имеет приоритет, равный −5. Для того чтобы понизить приоритет выполнения процесса на десять, мы должны выполнить следующую команду:

```
nice 10 top
```

В результате процесс top имеет приоритет, равный 5.

Только пользователь root может поднять приоритет того или иного процесса, используя для этого *отрицательное* значение параметра *number*.

renice

Программа renice, в отличие от программы nice, позволяет изменить приоритет уже работающего процесса. Формат запуска программы следующий:

```
renice -number PID
```

В общем, программа renice работает точно так же, как и nice. Уровень приоритета процесса определяется параметром *number*, при этом большее его значение означает меньший приоритет процесса. Значение по умолчанию — 10, и *number* представляет собой число, на которое должен быть уменьшен приоритет процесса.

Только пользователь root может поднять приоритет того или иного процесса, используя для этого *отрицательное* значение параметра *number*.

Выполнение процессов в заданное время

Одна из основных задач автоматизации администрирования операционной системы — выполнение программ в заданное время или с заданной периодичностью. Конечно, можно запускать программы самостоятельно, но проводить 24 часа на работе или постоянно удаленно запускать программы в самое неподходящее время (часа в три ночи) — безумие. Для решения этих проблем существует несколько утилит, позволяющих запускать процессы в нужное время.

at

Для запуска одной или более команд в заранее определенное время используется команда `at`. В этой команде вы можете определить время и дату запуска той или иной команды. Команда `at` требует, по меньшей мере, двух параметров — время выполнения программы и запускаемую программу с ее параметрами запуска.

Приведенный ниже пример запустит команду на выполнение в 01:01. Для этого введите все, приведенное ниже, с терминала, завершая ввод каждой строки нажатием клавиши <Enter> и по окончании ввода всей команды — <Ctrl>+<D> для ее завершения.

```
at 1:01
ls
echo "Time is 1:01"
```

Помимо времени, в команде `at` может быть также определена и дата запуска программы на выполнение.

Пользователь root может без ограничения применять практически любые команды. Для обычных пользователей права доступа к команде `at` определяются файлами /etc/at.allow и /etc/at.deny. В файле /etc/at.allow содержится список тех, кому разрешено использовать команду `at`, а в файле /etc/at.deny находится список тех, кому ее выполнять запрещено.

batch

Команда `batch` в принципе аналогична команде `at`. Более того, `batch` представляет собой псевдоним команды `at -b`. Для чего необходима эта команда? Представьте, вы хотите запустить резервное копирование вечером. Однако в это время система очень занята, и выполнение резервирования системы практически парализует ее работу. Для этого и существует команда `batch` — ее использование позволяет операционной системе самой решить, когда наступает подходящий момент для запуска задачи в то время, когда система не сильно загружена.

Формат команды `batch` представляет собой просто список команд для выполнения, следующих в строках за командой; заканчивается список комбинацией клавиш <Ctrl>+<D>. Можно также поместить список команд в файл и перенаправить его на стандартный ввод команды `batch`.

cron

Cron — это программа, выполняющая задания по расписанию, но, в отличие от команды `at`, она позволяет выполнять задания неоднократно. Вы определяете времена и даты, когда должна запускаться та или иная программа. Времена и даты могут определяться в минутах, часах, днях месяца, месяцах года и днях недели.

Программа cron запускается один раз при загрузке системы. При запуске cron проверяет очередь заданий `at` и задания пользователей в файлах crontab. Если для запуска не было найдено заданий — следующую проверку cron произведет через минуту.

Для создания списка задач для программы cron используется команда `crontab`. Для каждого пользователя с помощью этой команды создается его собственный crontab-файл со списком заданий, имеющий то же имя, что и имя пользователя.

Каждая строка в файле crontab содержит шаблон времени и команду. Команда выполняется тогда, когда текущее время соответствует приведенному шаблону. Шаблон состоит из пяти частей, разделенных пробелами или символами табуляции, и имеет вид:

```
минуты часы день_месяца месяц день_недели задание
```

Первые пять полей представляют собой шаблон времени и обязательно должны присутствовать в файле. Для того чтобы программа cron игнорировала поле шаблона времени, поставьте в нем символ звездочки (*).

Например, шаблон `10 01 01 * *` говорит о том, что команда должна быть запущена в десять минут второго каждого первого числа любого (*) месяца, каким бы днем недели оно ни было. В табл. 26.3 приведено описание полей таблицы задания cron.

Таблица 26.3. *Параметры таблицы заданий программы cron*

Поле	Описание
`минуты`	Указывает минуты в течении часа. Значения от 0 до 59
`часы`	Указывает час запуска задания. Значения от 0 до 23, где 0 — полночь
`день_месяца`	Указывает день месяца, в который должна исполняться команда
`месяц`	Указывает месяц, в который необходимо запускать задание. Значения лежат в пределах от 1 до 12, где 1 — январь
`день_недели`	Указывает день недели — или как цифровое значение от 0 до 7 (0 и 7 означают воскресенье) или используя первые три буквы, например Mon
`задание`	Командная строка для запуска задания

Ниже приведены несколько команд, исполняемых программой cron:

- команда запускается в 1 минуту каждого часа:

```
01 * * * * /usr/bin/script
```

- команда запускается каждый день в 8:20:

```
20 8 * * *  /usr/bin/script
```

- команда запускается в 6 часов каждое воскресенье:

```
00 6 * * 0 /usr/bin/script
```

- команда запускается в 7:40 каждое первое число:

```
40 7 1 * * /usr/bin/script
```

Для создания и редактирования файла заданий для программы cron используется команда `crontab`. Прямое редактирование файла заданий не допускается.

Команда `crontab` имеет следующие параметры командной строки:

- `-e` — позволяет редактировать компоненты файла (при этом вызывается редактор, определенный в переменной `EDITOR`);

- `-r` —удаляет текущий crontab-файл из каталога;
- `-l` — используется для вывода списка текущих заданий.

Cron также имеет возможность разрешать или запрещать конкретным пользователем свое использование. Для этого существуют файлы /etc/cron.allow и /etc/cron.deny, которые аналогичны описанным ранее /etc/at.allow и /etc/at.deny.

Ссылки

www.tts.esoo.ru/~lesenka/linux/slack_book.html — Дэвид Кэнтрелл, Логэн Джонсон, Крис Люменс. Основы Slackware Linux. Официальный учебник.

Глава 27

Администрирование сети

Администратор сети просто не в состоянии надежно
контролировать всю сеть в архитектуре клиент/сервер,
что чревато несанкционированным доступом.
Непродуманные действия еще больше
усиливают эту опасность.

Синди Куллен

Пожалуй, одна из самых сложных и трудоемких задач системного администратора — администрирование сети. Эта задача настолько комплексная, что можно практически все, о чем писалось ранее, отнести к подготовке администрирования сети. Слишком много параметров, программ, настроек могут прямо или косвенно отражаться на функционировании сети и сетевых сервисов. Мы уже приводили чье-то крылатое высказывание: "Компьютер — это сеть".

В этой главе все, так или иначе, будет касаться администрирования и управления сетью, хотя некоторые вещи с первого взгляда никоим образом не относятся к сети или ее настройке.

В той части главы, где будет говориться об инструментах, предназначенных для обнаружения уязвимости системы, мы опишем несколько программных пакетов, которые с одинаковым успехом можно применить как для взлома системы, так и для ее защиты.

Расширенное управление доступом к файлам

К сожалению, стандартные средства организации прав доступа к файлам в UNIX-подобных операционных системах зачастую не удовлетворяют требованиям некоторых системных администраторов. Проблема заключается в том, что определение прав доступа к файлам сводится к установке девяти битов, с помощью которых можно задать права доступа для владельца файла, группы, к которой принадлежит владелец файла, а также для всех ос-

тальных. Часто необходимо настроить доступ к файлу достаточно сложным образом — допустим, три человека из трех разных групп имеют право делать с файлом все что угодно, десять человек из других групп могут открывать файл на чтение, а еще десять — только выполнять. Для всех других пользователей доступ к этому файлу необходимо запретить. Устроить нечто подобное стандартными средствами Linux весьма нетривиальная задача. В такой ситуации для решения данной проблемы можно воспользоваться Linux ACLs (Access Control Lists, списки контроля доступа) — версией POSIX ACLs для Linux. Linux ACLs — это набор патчей для ядра операционной системы и программ для работы с файловой системой и несколько утилит, дающих возможность устанавливать права доступа к файлам не только для пользователя-владельца и группы-владельца файла, но и для любого пользователя или группы.

Linux ACLs использует расширенные атрибуты (Extended Attributes) для хранения данных о правах доступа к файлам пользователей и групп. Расширенные атрибуты — это пара имя/значение, привязанная к определенному файлу.

Список расширенного контроля доступа существует для каждого inode и состоит из шести компонентов. Первые три являются копией стандартных прав доступа к файлу. Они содержатся в единственном экземпляре в ACL и есть у каждого файла в системе:

- ACL_USER_OBJ — режим доступа к файлу пользователя-владельца;
- ACL_GROUP_OBJ — режим доступа к файлу группы-владельца;
- ACL_OTHER — режим доступа к файлу остальных пользователей.

Следующие два компонента устанавливаются для каждого файла в отдельности и могут присутствовать в ACL в нескольких экземплярах:

- ACL_USER — содержит UID и режим доступа к файлу пользователя, которому установлены права, отличные от основных. На каждого пользователя со своими правами на данный файл хранится отдельная запись. Не может существовать более одной записи на одного и того же пользователя;
- ACL_GROUP — то же самое, что и ACL_USER, но для группы пользователей;
- ACL_MASK — маска действующих прав доступа для расширенного режима.

При установке дополнительных прав доступа присваивается значение и элементу ACL_MASK.

Каталоги также могут иметь список контроля доступа по умолчанию. В отличие от основного ACL, он действует на создаваемые внутри данного каталога файлы и каталоги. При создании файла внутри такого каталога, файл получает ACL, равный ACL по умолчанию этого каталога.

Установка Linux ACLs

Для использования Linux ACLs необходимо получить на сайте разработчиков собственно пакет Linux ACLs и патчи для ядра операционной системы Linux и некоторых утилит. Само собой, после наложения патчей придется перекомпилировать ядро операционной системы и утилиты.

При подготовке к компиляции ядра операционной системы Linux необходимо выполнить следующие действия:

1. В меню **Code Maturity Level Options** отметить пункт **Prompt for development and/or incomplete code/drivers**.
2. В меню **Filesystems** отметить пункт **Extended filesystem attributes (EXPRIMENTAL)**.
3. Затем отметить два подпункта: **Extended user attributes** и **Access Control Lists**.
4. В пункте **Second extended fs support** отметить подпункт **Extended attributes for ext2 (DANGEROUS)**.

После этого можно компилировать ядро операционной системы.

После компиляции и установки ядра операционной системы следует переходить к установке утилит управления расширенным доступом к файлам и накладыванию патчей на стандартные утилиты.

Для установки утилит необходимо скомпилировать пакет ACL, который также берется на сайте разработчиков. Процесс компиляции и установки подробно описан в документации, входящей в комплект пакета. На том же сайте берем патчи к стандартным утилитам операционной системы и применяем их по рецепту, приведенному в документации. После этого можно произвести перезагрузку операционной системы.

Установка и изменение прав доступа

Управление списками контроля доступа производится при помощи двух утилит — getfacl и setfacl.

С помощью getfacl можно просмотреть текущие параметры доступа любого файла. Например, при вызове getfacl для домашнего каталога пользователя vasya мы получим следующее:

```
getfacl  /home/vasya
file: home/vasya
owner: vasya
group: users
user::rwx
group::---
other:---
```

Как можно видеть, каталог /home/vasya принадлежит пользователю vasya, группе users и значение прав доступа к каталогу — 0700. Каталог имеет только основные параметры доступа, поскольку изначально дополнительные права не устанавливаются.

Дополнительные права доступа к файлу устанавливаются и изменяются при помощи утилиты setfacl. Для этого используется следующий формат вызова:

```
setfacl -опции ACL_структура, ACL_структура,...,ACL_структура имя_файла
имя_файла ...
```

ACL-структура представляет собой одну из следующих конструкций:

- `[d:][u:][пользователь] [:[+|^]режимы_доступа]` — определяет режим доступа к файлу или каталогу пользователя. Если пользователь не указан, определяет режим доступа пользователя-владельца;
- `[d:] g:[группа] [:[+|^]режимы_доступа]` — то же, что и предыдущая конструкция, но для группы;
- `[d:] m [:[+|^] режимы_доступа]` — определяет действующие права доступа;
- `[d:] o [:[+|^] режимы_доступа]` — определяет режим доступа для остальных пользователей.

Для установки и изменения ACL используются следующие опции:

- `-s` — заменяет полностью ACL файла на указанный в командной строке;
- `-m` — изменяет режимы доступа к файлу (каталогу);
- `-x` — убирает правила доступа из ACL.

К примеру, вот что мы получим, применив setfacl к каталогу vasya:

```
setfacl -s u::rwx,g::---,o:---,u:us1:rwx,g:usrs2:rx,u:us2:--- /home/vasya
getfacl /home/dh

file: home/vasya
owner: vasya
group: users
user::rwx
user:us1:rwx
user:us2:---
group::---
group:usrs2:r-x
mask:rwx
other:---
```

Дополнительные возможности

Кроме основных опций запуска, обе команды имеют большое количество дополнительных. Мы не будем останавливаться на этих возможностях, поскольку пакет динамично изменяется, и вполне возможно, что он уже обладает существенно бóльшими возможностями по сравнению с теми, которые присутствовали на момент написания книги. Мы не будем этого делать и потому, что пакет управления правами доступа вряд ли понадобится обычному пользователю или администратору небольшой локальной сети, а администратор большой фирмы должен быть в состоянии самостоятельно разобраться в возможностях любого программного пакета.

Шифрование трафика

Традиции — вещь очень неоднозначная. Иногда они помогают жить и успешно развиваться, иногда они просто странные или бесполезные, а иногда — весьма вредны. То же самое можно сказать и о сетевых протоколах — традициях компьютерного мира.

Бóльшая часть существующих сетевых протоколов разрабатывалась по компьютерным меркам в чуть ли не доисторическую эпоху рыцарских традиций, когда о сетевых взломах и сетевом шпионаже можно было прочитать только в научной фантастике. Как результат — подавляющее большинство данных в сети Интернет передаются в открытом виде. И как обратная сторона медали — существует большое количество утилит для прослушивания сетевого трафика. Многие из них умеют сами анализировать перехватываемые данные. С помощью таких утилит можно получить пароли пользователей для различных сетевых служб, тексты электронных писем, файлы, сообщения, переданные по ICQ, и т. д., и т. п. Защитить себя от такого прослушивания можно с помощью шифрования трафика.

Наиболее распространенным протоколом шифрования является протокол SSL, разработанный Netscape Communications. Чаше всего он используется для шифровки протокола HTTP (HTTPS), но также может применяться для создания защищенных соединений с SMTP, POP3, IMAP и другими высокоуровневыми сетевыми протоколами.

Программа, осуществляющая поддержку протокола SSL почти для любых серверных и клиентских приложений под Linux и Windows, называется Stunnel. Основное ее применение состоит в создании надежного шифрованного канала между двумя и более хостами в сетях, где существует угроза прослушивания трафика.

Stunnel

Как обычно, рекомендуется получить с сайта разработчика последнюю версию программного пакета.

Установка

Для работы Stunnel необходимо установить OpenSSL. Обычно OpenSSL устанавливается при инсталляции операционной системы Linux (по крайней мере, в дистрибутиве Red Hat Linux), поэтому проблем с установкой OpenSSL возникнуть не должно. Пакет Stunnel так же обычно входит в состав дистрибутива в виде RPM-пакета.

Организация шифрованного туннеля

Stunnel может работать в двух режимах — сервера и клиента. В качестве сервера Stunnel открывает указанный порт, дешифрует все поступившие данные и передает их либо в указанную в параметрах запуска программу, либо на указанный порт на указанном хосте. В качестве клиента Stunnel открывает указанный порт, шифрует все поступившие на него данные и передает их в определенную программу или на определенный порт на заданном хосте.

Давайте организуем защищенное telnet-соединение (хотя это и не имеет практической пользы, поскольку есть SSH) между двумя компьютерами А и Б.

На компьютере Б запускаем Stunnel в режиме сервера:

```
stunnel -d 999 -r 23
```

Опция `-d` указывает Stunnel работать в режиме отдельного демона, ждущего соединения по порту 999. Все данные, полученные в шифрованном виде на порт 999, в открытом виде передаются на порт 23 на локальной машине.

Затем на компьютере А запускаем Stunnel в режиме клиента:

```
stunnel -c -d 1055 -r B:999
```

Опция `-c` указывает на работу в режиме клиента, все данные, полученные в открытом виде на порт 1055, передаются в шифрованном виде на порт 999 на хосте Б.

После проделанных манипуляций можно устанавливать telnet-соединение с компьютером Б. Команда запуска telnet на компьютере А будет выглядеть следующим образом:

```
telnet localhost 1055
```

Несколько непривычно, зато трафик полностью шифруется. Точно по такому же принципу организовывается шифрованный туннель и для других сетевых протоколов.

Stunnel и приложения, поддерживающие SSL

Достаточно часто возникает ситуация, когда одно из приложений поддерживает протокол SSL, а приложение с другой стороны не поддерживает протокол SSL. В этом случае Stunnel можно запускать только с одной стороны — там, где приложение не способно поддерживать протокол SSL. Но в этом

случае возникает проблема — какие порты используются приложением, поддерживающим протокол SSL.

Существует официальный список SSL-портов, который приведен ниже:

```
https          443/tcp   # http protocol over TLS/SSL
smtps          465/tcp   # smtp protocol over TLS/SSL (was ssmtp)
nntps          563/tcp   # nntp protocol over TLS/SSL (was snntp)
imap4-ssl      585/tcp   # IMAP4+SSL (use 993 instead)
sshell         614/tcp   # SSLshell
ldaps          636/tcp   # ldap protocol over TLS/SSL (was sldap)
ftps-data      989/tcp   # ftp protocol, data, over TLS/SSL
ftps           990/tcp   # ftp protocol, control, over TLS/SSL
telnets        992/tcp   # telnet protocol over TLS/SSL
imaps          993/tcp   # imap4 protocol over TLS/SSL
ircs           994/tcp   # irc protocol over TLS/SSL
pop3s          995/tcp   # pop3 protocol over TLS/SSL (was spop3)
```

Сертификаты

Программа Stunnel имеет возможность проверки подлинности сертификатов тех хостов, к которым или с которых идет подключение. Для этого предназначена опция командной строки `-v`. После `-v` необходимо указать уровень проверки сертификата. Он может иметь следующие значения:

- `0` — никакой проверки наличия и подлинности сертификата не производится;
- `1` — сертификат проверяется на подлинность, если присутствует. Если сертификат не является подлинным — соединение не устанавливается;
- `2` — проверяется присутствие сертификата и его подлинность. Если сертификат отсутствует или не является подлинным — соединение не устанавливается;
- `3` — проверяется присутствие сертификата и его наличие в списке проверенных сертификатов. Если сертификат отсутствует или его нет в списке проверенных сертификатов — соединение не устанавливается.

Сертификат создается при сборке пакета и помещается вместе с секретным ключом, используемым при расшифровке входящего трафика, в файл stunnel.pem.

Более полную информацию по этому программному обеспечению смотрите в документации, идущей в комплекте с Stunnel.

Утилиты сканирования и защиты сети

Утилиты сканирования — это класс программного обеспечения, предназначенный для нахождения уязвимостей в конфигурации компьютера или сети. Они могут быть использованы и как средство для улучшения безопасности системы, и как инструмент для взлома системы.

SATAN

Одна из старейших утилит сканирования. Говорят, что автора этого пакета уволили из фирмы, где он работал, из-за того, что он выложил SATAN на свой Web-сайт.

SATAN может работать на нескольких операционных системах. Считается устаревшим, но тем не менее для проверки правильности основных сетевых настроек вполне пригоден. Работает от пользователя root, требует наличия Perl.

После запуска SATAN становится Web-сервером и запускает браузер Netscape, поскольку интерфейс у него — Web-ориентированный. Для начала сканирования необходимо указать сканируемый хост или диапазон адресов и "уровень нападения", который может быть слабым, нормальным и тяжелым. После этого кнопкой **Start the scan** запускается сканирование.

По окончании сканирования необходимо перейти в раздел **Reporting & Data Analysis**. В этом разделе можно ознакомиться с найденными проблемами, которые необходимо устранить.

Portsentry

Еще один программный продукт, предназначенный для обнаружения сканирования сетевых портов. Основные возможности программы Portsentry:

- обнаруживает практически все известные виды сканирования компьютеров;
- в реальном времени блокирует компьютер, производящий сканирование, посредством установленного на атакуемом компьютере брандмауэра, команду запуска которого можно задать в файле конфигурации;
- записывает в журнал операционной системы посредством syslogd информацию об атаке;
- может вызывать любую указанную в файле конфигурации программу, в ответ на сканирование или подключение к защищенному сетевому порту.

Установка и настройка

Процесс установки подробно описан в документации на программу и не вызывает трудностей, поэтому сразу перейдем к настройке программы.

Основной конфигурационный файл программы Portsentry называется portsentry.conf. Содержимое файла portsentry.conf представляет собой несколько строк, каждая из которых имеет вид:

```
ОПЦИЯ = "значение"
```

Ниже приведен список основных поддерживаемых опций:

- `TCP_PORTS` — в этой опции через запятую перечисляются TCP-порты, которые проверяются программой Portsentry. При обнаружении подключения к перечисленным портам Portsentry записывает информацию об этом в системный журнал и выполняет команду, заданную пользователем, а после этого блокирует хост посредством брандмауэра. TCP-порты, открытые на защищаемом компьютере другими программами, в этот список включаться не должны;
- `UDP_PORTS` — то же, что и `TCP_PORTS`, но для UDP-портов;
- `ADVANCED_PORTS_TCP` — значение этой опции определяет верхнюю границу множества TCP-портов, которые проверяются Portsentry при работе в режиме Advanced Stealth Scan Detection Mode. Нижней границей является 1, т. е. при значении `ADVANCED_PORTS_TCP`, равном 2048, проверяется подключение к любому порту в промежутке от 1 до 2048;
- `ADVANCED_PORTS_UDP` — то же, что и `ADVANCED_PORTS_TCP`, но для UDP-портов;
- `ADVANCED_EXCLUDE_TCP` — TCP-порты, которые исключаются из промежутка проверяемых портов, заданного параметром `ADVANCED_PORTS_TCP`. Здесь обязательно нужно перечислить TCP-порты, открытые работающими на защищаемом компьютере программами;
- `ADVANCED_EXCLUDE_UDP` — то же, что и `ADVANCED_EXCLUDE_TCP`, но для UDP-портов;
- `IGNORE_FILE` — имя и путь к файлу с IP-адресами хостов, которые не блокируются при подключении к портам, проверяемым программой Portsentry;
- `HISTORY_FILE` — имя и путь к файлу с историей работы программы Portsentry. В файл записывается время блокирования, имя и IP хоста, атакованный порт, протокол;
- `BLOCKED_FILE` — строка, из которой формируется имя и путь к файлам, куда записывается информация о блокированных хостах;
- `BLOCK_TCP` — эта опция в зависимости от значения задает ответную реакцию Portsentry на сканирование портов:
  - `0` — не блокировать хост, не запускать заданную пользователем команду;

- `1` — блокировать хост и запустить команду;
- `2` — только запустить заданную команду.

Команда задается при помощи опции `KILL_RUN_CMD`;

- `BLOCK_UDP` — то же, что и `BLOCK_TCP`, но для UDP;
- `KILL_ROUTE` — эта опция задает команду, которую надо выполнить для блокирования атакующего хоста. Для указания IP-адреса используется переменная `$TARGET$`. Переменная `$PORT$` используется для указания порта, к которому было подключение;
- `KILL_HOSTS_DENY` — эта опция задает строку, которая записывается в /etc/hosts.deny для блокирования доступа к сервисам, запускаемым через inetd;
- `KILL_RUN_CMD` — с помощью этой опции можно задать команду, запускаемую до блокирования хоста;
- `SCAN_TRIGGER` — данная опция задает количество разрешенных подключений к проверяемым программой Portsentry портам одного и того же хоста, прежде чем Portsentry начнет действовать. 0 определяет немедленную реакцию;
- `PORT_BANNER` — задает сообщение, которое будет выводиться при подключении к проверяемому Portsentry порту.

В файле portsentry.ignore необходимо перечислить IP-адреса компьютеров, которые не должны быть блокированы программой при подключении к проверяемому порту.

Запуск

Portsentry можно запускать в трех различных режимах. Режимы задаются в командной строке при вызове Portsentry. Одновременно можно задать только один режим работы для одного протокола:

- Classic — при работе в этом режиме Portsentry открывает порты, указанные в `TCP_PORTS` или `UDP_PORTS`, и ждет соединения. При попытке подключиться к такому порту происходит блокировка удаленного хоста. Этот режим работы задается опциями командной строки `-tcp` — для TCP-портов и `-udp` — для UDP-портов;
- Enhanced Stealth Scan Detection — этот режим используется для проверки перечисленных в `TCP_PORTS` или `UDP_PORTS` портов на предмет подключения или сканирования. Выявляет почти все виды Stealth-сканирования, а не только сканирование подключением. В отличие от режима Classic, не держит открытыми порты, поэтому сканировщик получает достоверную информацию об открытых портах. Задается опциями командной строки `-stcp` — для TCP-портов и `-sudp` — для UDP-портов;

- Advanced Stealth Scan Detection — этот режим используется для проверки всех портов в промежутке от 1 до `ADVANCED_PORT_TCP` или `ADVANCED_PORT_UDP`. Порты, открытые другими программами и перечисленные в `ADVANCED_EXLUDE_TCP` или `ADVANCED_EXCLUDE_UDP`, исключаются из проверки. Любой компьютер, попытавшийся подключиться к порту в этом промежутке, тут же блокируется. Задается опциями командной строки `-atcp` — для TCP-портов и `-audp` — для UDP-портов.

Сетевая статистика

Очень часто администратору необходимо получить развернутую информацию по сетевому трафику — кто, когда, сколько и по какому протоколу отправлял/принимал информацию. Конечно, все это можно получить из различных log-файлов, однако незачем тратить время на изготовление анализаторов log-файлов, когда уже есть готовые программные решения.

NeTraMet

Этот программный пакет позволяет подсчитывать трафик по IP-адресам в локальной сети раздельно по типам трафика: SMTP, ICMP, HTTP, FTP, UDP, TCP и т. п. Также существует возможность подробного регистрирования трафика.

Программный пакет состоит из:

- NeTraMet — программы-сборщика трафика. Собирает и хранит в оперативной памяти статистику с сетевых интерфейсов сервера;
- NeMaC — программы-менеджера сборщика NeTraMet. NeMaC собирает статистику и записывает ее в журнал;
- srl — компилятора правил для NeMaC;
- fd_filter — программы обработки журналов NeMaC;
- fd_extract — программы обработки результатов fd_filter.

Ключи запуска NeTraMet

Программа запускается со следующими ключами:

- `-i network_interface` — определяет сетевой интерфейс, трафик которого будет считать NeTraMet;
- `-l` — предписывает использовать размер пакета из заголовка, а не аппаратный размер;
- `-m 614` — определяет UDP-порт, на котором будет соединяться NeTraMet с NeMaC;

- `-r password_for_read` — устанавливает пароль на чтение;
- `-w password_for_write_and_read` — устанавливает пароль на чтение/запись;
- `-f 60000` — определяет максимальное количество сетевых потоков в NeTraMet. Чем больше клиентов, трафика и степень детализации статистики, тем больше сетевых потоков.

Ключи запуска NeMaC

Программа запускается со следующими ключами:

- `-k 120` — каждые 120 секунд NeMaC будет проверять — не перезагрузился ли NeTraMet;
- `-F /var/ntm.log/$DATER.flows` — в этот файл записывать статистику;
- `-m 614` — определяет порт для управления NeTraMet;
- `-c 900` — предписывает забирать статистику с NeTraMet каждые 15 минут;
- `-p` — предписывает после записи в файл статистики данных закрывать его. Если файл не найден, то создается новый файл;
- `-L /var/ntm.log/$DATER.nemac` — журнал работы NeMaC;
- `-r /root/ntm.sh/short.3.rules` — файл с правилами.

Протоколирование

Нет смысла тратить много времени на защиту компьютера от взлома и не обращать внимания на систему протоколирования событий. Каким образом вы сможете узнать о попытке и способе взлома, не используя инструментов для ведения log-файлов? В этом разделе мы познакомимся со стандартной системой ведения log-файлов — демоном syslogd.

Демон syslogd является частью пакета sysklogd, в который входят две программы: syslogd и klogd. Syslogd отвечает за протоколирование сообщений системы, а klogd — ядра.

Демон syslogd

Демон syslogd запускается автоматически при старте системы и обеспечивает протоколирование событий, которое используется большинством программ. Демон syslogd пишет сообщения в файлы /var/log/* в зависимости от настроек. Обычно записи в log-файле, создаваемом syslogd, содержат следующие поля: дата и время, имя компьютера, программа, сообщение.

Параметры запуска

В табл. 27.1 приведены основные параметры командной строки демона syslogd.

Таблица 27.1. *Основные параметры командной строки syslogd*

Параметр	Описание
`-d`	Включает режим отладки
`-f file`	Определяет альтернативный файл конфигурации
`-h`	По умолчанию демон не перенаправляет сообщения, которые он получает от других узлов. Этот параметр позволяет перенаправить сообщения другим хостам
`-n`	Этот параметр нужен, если syslogd запускается и контролируется программой init
`-p socket`	Позволяет задать другой сокет UNIX вместо /dev/log
`-r`	Позволяет принимать сообщения из сети
`-s socket`	Этот параметр позволяет указать дополнительный сокет, который syslog должен прослушивать
`-v`	Выводит версию syslogd

Файл конфигурации

По умолчанию используется файл конфигурации /etc/syslog.conf. Вы можете указать другой файл конфигурации с помощью опции `-f`. Типичный файл конфигурации приведен ниже.

```
# Все сообщения ядра операционной системы выводить на консоль
#kern.*                                          /dev/console

# Все сообщения уровня info или выше протоколировать в файл
# /var/log/messages
# Кроме почтовых сообщений и сообщений аутентификации
*.info;mail.none;authpriv.none;cron.none      /var/log/messages

# Протоколирование аутентификации.
# Файл протокола /var/log/secure
authpriv.*                                      /var/log/secure

# Все log-сообщения почтовой системы сохранять в файле /var/log/maillog.
mail.*                                          /var/log/maillog
```

```
# Все сообщения демона cron сохранять в файле /var/log/cron
cron.*                                          /var/log/cron

# Everybody gets emergency messages
*.emerg                                         *

# Сообщения системы новостей уровня crit и выше сохранять в файле
# /var/log/spooler
uucp,news.crit                                  /var/log/spooler

# Все загрузочные сообщения хранить в файле /var/log/boot.log
local7.*                                        /var/log/boot.log
```

Файл конфигурации состоит из двух полей: объект протоколирования и файл, в который будут записываться сообщения, порождаемые этим объектом. Для каждого объекта можно указать один из уровней протоколирования:

- `debug` — отладочная информация;
- `info` — просто информация;
- `notice` — уведомление;
- `warn` — предупреждение;
- `err` — ошибка;
- `emerg` — критический уровень.

Первые три уровня протоколирования относятся к информационным сообщениям. Уровень `warn` — это предупреждения, а `err` — ошибки. Помимо этого, существуют критические сообщения, которые выводятся прямо на консоль. Для обозначения объектов и для обозначения уровней протоколирования можно использовать символ `*`, который обозначает все объекты или все уровни.

Сетевое протоколирование

Для обеспечения повышенной защищенности сети все сообщения можно хранить не на локальном компьютере, а передавать по сети на специальный сервер, на котором будет находиться база log-файлов компьютеров, подключенных к сети.

Для передачи сообщений используется протокол UDP. Для нормального функционирования необходимо в файле /etc/service раскомментировать строку `syslog 514/udp`.

После этого необходимо внести изменения в файл конфигурации /etc/syslog.conf — вместо файлов протоколов используйте параметр `@hostname`,

где hostname — это имя компьютера, на который будут перенаправлены сообщения.

Имя узла желательно указать в файле /etc/hosts, поскольку демон syslogd обычно стартует раньше, чем сервер DNS.

Демон klogd

Демон klogd предназначен для перехвата и протоколирования сообщений ядра Linux. В табл. 27.2 приведены основные параметры командной строки демона klogd.

Таблица 27.2. *Основные параметры командной строки klogd*

Параметр	Описание
-c n	Устанавливает уровень сообщений, которые будут выводиться на экран
-d	Режим отладки
-f file	Записывать сообщения в указанный файл раньше демона syslogd
-i	Позволяет перезагрузить символьную информацию ядра о модулях
-I	Перезагружает статическую символьную информацию и информацию о модулях ядра
-k file	Использует указанный файл в качестве файла, содержащего символьную информацию ядра
-n	Не переходить в фоновый режим. Этот параметр используется, когда демон управляется программой init
-o	Демон читает и протоколирует все сообщения, которые он найдет в буферах сообщений ядра. После одного цикла чтения/протоколирования демон завершает работу
-s	Заставляет демон klogd использовать системные вызовы для обращений к буферам сообщений ядра
-v	Выводит версию klogd

По умолчанию демон klogd вызывается системным вызовом для того, чтобы препятствовать отображению всех сообщений на консоль. Это не распространяется на критические сообщения ядра (kernel panic). Эти сообщения в любом случае будут отображены на консоли.

Защита системы после взлома

Правда, несколько странное название раздела? Как это — защита системы *после* взлома? Если вы помните, вопросы обеспечения безопасности компь-

ютера в целом уже рассматривались нами *в гл. 7.* В этом же разделе мы остановимся на сетевой безопасности, а именно на том моменте, когда взлом уже произошел. После обнаружения факта взлома стандартным решением является отключение взломанного компьютера от сети и полная переустановка операционной системы с последующей установкой всех обновлений программного обеспечения, используемого на компьютере. А что делать, если нет возможности вывести из работы взломанный компьютер, а защитить его все равно необходимо? Именно этот случай мы и рассмотрим в этом разделе.

Взломы операционной системы бывают разные. Самый простой вариант — какой-то подросток начитался литературы или нашел в Интернете программку-взламыватель, например sendmail, применил свои псевдознания к вашей системе, пошалил — удалил что-то с вашего компьютера или, наоборот, оставил послание — "ваш компьютер взломан супер-хакером Васей" и ушел, причем, зачастую, не уничтожив следы своего воздействия на компьютер. На такой простой случай вам рассчитывать не стоит. Как правило, серьезный взлом подготавливается долгое время, и о нем вы узнаете, например, от администратора какого-нибудь сервера на другом конце земного шара, да и то потому, что от вас на его сервер идет очень большой трафик или еще по каким-либо косвенным признакам. Такой взлом преследует чисто прагматические цели — воспользоваться вашей системой для дальнейшего взлома других компьютеров, устроить на вашем сервере хранилище файлов или что-нибудь в подобном роде. Причем взломщик после себя всегда оставляет на вашем компьютере набор специальных утилит, называемых rootkit.

Rootkit

Rootkit (набор инструментов администратора) — это набор утилит, которые взломщик устанавливает на взломанном компьютере после получения первоначального доступа. Rootkit обычно содержит сетевой sniffer (утилиту, способную получать и обрабатывать *весь* сетевой трафик вашей локальной сети, вне зависимости от того, какому компьютеру адресованы сетевые пакеты) для прослушивания сетевого трафика, программы для модификации log-файлов, позволяющие скрыть присутствие взломщика на вашем компьютере, и специально модифицированные системные утилиты, замещающие основные утилиты системы, например ps, netstat, ifconfig, killall, login.

Основное назначение rootkit — позволить взломщику возвращаться во взломанную систему и получать доступ к ней, не будучи при этом обнаруженным системным администратором. Обычно для этого используется модифицированная версия telnetd или sshd. Модифицированный сервис будет использовать сетевой порт, отличный то того, который этот демон по умолчанию прослушивает. Большинство версий rootkit снабжены модифицированными системными программами, которые замещают существующие во взломанной

системе. Конкретный набор модифицируемых системных утилит весьма зависит от версии rootkit и нужд и квалификации взломщика, но, как правило, заменяются программы ps, w, who, netstat, ls, find, login и другие, которые могут быть использованы для контроля за работой взломанной системы.

Для усложнения обнаружения подмены системных утилит большинство rootkit, производя замену системных утилит на модифицированные версии, устанавливают точно такие же даты их создания и размеры файлов, поэтому простой список файлов с датой их создания и модификации и размером никакой пользы в обнаружении подмены системных утилит не принесет. Исходя из этого, пожалуй, лучший способ обнаружения подмены системных утилит — получить контрольную сумму файлов в системе и сохранить этот список в надежном месте — на другом компьютере или на компакт-диске.

В принципе, можно воспользоваться возможностями, предоставляемыми менеджером пакетов RPM — контрольной суммой пакета, рассчитанной по алгоритму MD5. При этом RPM использует контрольные суммы пакетов, хранящиеся в базе данных установленных RPM. Как легко заметить, данный способ не подходит для обнаружения опытных взломщиков. Причин к тому две.

- В вашей системе могут быть установлены программы не из RPM, а скомпилированные из исходных кодов — совершенно очевидно, что ваш менеджер пакетов абсолютно ничего не знает о программах, устанавливаемых без помощи RPM.
- База данных RPM находится на взломанном компьютере, и взломщику не составляет труда модифицировать ее нужным образом или вообще повредить ее.

Для решения этой проблемы обычно используются специализированные программные пакеты, например Tripwire или AIDE, о которых мы поговорим несколько позже.

Помимо вышеперечисленного, некоторые rootkit содержат сетевой анализатор пакетов и утилиты для записи нажатий клавиатурных кнопок, что позволяет взломщику с целью получения необходимой информации организовать сбор паролей и анализ сетевого трафика.

Наибольшую угрозу для безопасности вашей системы представляют rootkit, использующие загружаемые модули ядра (Loadable Kernel Module, LKM), что позволяет не подменять системные утилиты, а нарушать их правильное функционирование через ядро операционной системы.

Обнаружение rootkit

Мы нарисовали достаточно мрачную картину — получается, что после взлома системы сделать для ее излечения ничего не возможно? К счастью, не все так плохо.

Сначала необходимо определить сам факт взлома системы. Возможным последствием взлома вашего компьютера и установки на нем rootkit может стать изменение в поведении системных утилит. Например, некоторые утилиты отказываются запускаться от имени пользователя, которому было разрешено пользоваться этими утилитами. Или ваша любимая утилита top стала выглядеть несколько иначе. Другие очень настораживающие признаки — изменение показателей сетевого трафика, а также резкое уменьшение свободного места на жестком диске.

Сканирование портов

После обнаружения взлома первое, что необходимо сделать после смены паролей — лишить взломщика возможности проникновения в систему через сетевые порты. Поскольку взломанный компьютер не вызывает доверия, просканировать сетевые порты необходимо с другого компьютера.

Проще всего просканировать порты с помощью программы nmap. Для этого достаточно выполнить следующую команду:

```
nmap -p 1-65535 192.168.0.1
```

Указываем диапазон сканируемых портов — от 1 до 65535, а также адрес сканируемого компьютера. После этого на консоль будет выдан список портов, протокол, используемый для каждого порта, и сервис, который использует этот порт. Обычно всякие "специальные" программы обращаются к портам выше 1023, причем зачастую это порты с номером выше десяти тысяч.

Помимо nmap, можно воспользоваться программой lsof. Она позволяет получить список открытых на вашем компьютере сетевых портов. Для этого достаточно выполнить команду

```
lsof -i
```

Использование RPM

Хотя чуть ранее мы утверждали, что использование RPM для обнаружения rootkit — дело бесперспективное, это не совсем так. RPM можно применить для быстрой проверки. Если он не найдет ничего подозрительного — воспользуемся другими средствами, если найдет — и на том спасибо — будем знать, что́ у нас не так в системе.

RPM записывает и проверяет контрольную сумму всех файлов в пакете, включая те файлы, которые должны изменяться с течением времени. О проверке контрольных сумм пакетов RPM см. в *гл. 8*.

Сканер для rootkit

Пакет chkrootkit — набор утилит, используемых для выявления присутствия в системе уже известных rootkit. Chkrootkit удобен тем, что способен выяв-

лять большое количество rootkit с помощью единственного приложения. Вдобавок к выявлению известных rootkit, он также включает несколько тестов, помогающих обнаружить новые rootkit.

Пакет chkrootkit состоит из следующих утилит:

- chkrootkit — используется для выявления сигнатур известных rootkit;
- ifpromisc — используется для обнаружения прослушивания сетевого трафика взломанным компьютером;
- chklastlog, chkwtmp, check_wtmpx — утилиты для проверки log-файлов;
- chkproc — предназначена для обнаружения "посторонних" загружаемых модулей ядра операционной системы.

Об особенностях применения chkrootkit можно узнать в документации, идущей в комплекте с пакетом.

После обнаружения

Что делать после обнаружения rootkit? Единственно верный способ избавиться от последствий взлома — заново полностью переустановить операционную систему и установить все обновления пакетов для вашего дистрибутива. Однако не всегда есть возможность проделать такие действия сразу — квартальный отчет, непрерывное производство, болезнь администратора — да мало ли что еще.

В дистрибутивах на основе RPM-пакетов вы можете определить поврежденные пакеты. После этого необходимо переустановить их, используя следующую команду:

```
rpm -U --force rpm_package_name.rpm
```

После переустановки пакетов вы должны удалить файлы, установленные в вашу систему взломщиком. Данные, полученные chkrootkit, помогут вам определить местонахождение файлов.

После удаления всех обнаруженных "чужих" файлов запустите top и ps для выявления и уничтожения оставшихся нежелательных процессов. Помимо этого, необходимо проверить стартовые скрипты операционной системы и убедиться, что эти скрипты не используются никакими посторонними программами.

LIDS

LIDS (Linux Intrusion Detection/Defence System) — система обнаружения и защиты от вторжения. Представляет собой дополнение к ядру операционной системы Linux, добавляющее дополнительные возможности для увеличения безопасности операционной системы. LIDS позволяет запретить или огра-

ничить доступ к файлам, памяти, устройствам, сетевым интерфейсам, запущенным приложениям и т. п. пользователю root, что дает возможность надежно оградить даже взломанную операционную систему от дальнейшего вмешательства.

В отличие от других средств защиты операционной системы Linux, эту систему невозможно отключить, не зная пароля администратора LIDS, который в зашифрованном виде хранится в специальном файле, видимом только программой администрирования LIDS. Точно так же защищены и конфигурационные файлы LIDS. Даже узнав каким-то образом пароль администратора LIDS, отключить систему можно, только находясь за консолью компьютера.

LIDS позволяет распределять права доступа к файлам на уровне программ, а не на уровне пользователей, а также запретить перезапуск операционной системы, загрузку/выгрузку модулей ядра и многое другое.

Информация о всех действиях, имеющих отношение к защищаемым объектам, помимо записи в log-файлах может немедленно отправляться по электронной почте.

Помимо всего прочего, в LIDS присутствует встроенный детектор сканирования сетевых портов.

Установка

После получения пакета LIDS необходимо разархивировать его и наложить патч на исходники ядра операционной системы Linux. После этого следуйте инструкции — там все понятно — компилируем, устанавливаем.

Далее, нам необходимо перекомпилировать ядро операционной системы Linux с поддержкой LIDS. Для этого в пункте меню конфигурации ядра **Code maturity level options** необходимо включить опцию **Prompt for development and/or incomplete code/drivers**.

После этого в пункте меню **General setup** необходимо включить опцию **Sysctl support**.

Далее необходимо зайти в меню **Linux Intrusion Detection System**. Это меню полностью относится к конфигурированию LIDS. Первым идет включение поддержки LIDS в ядре:

```
[*] Linux Intrusion Detection System support (EXPERIMENTAL)
```

После включения поддержки LIDS станет доступным список опций настройки LIDS:

- **Maximum protected objects to manage** — этот пункт позволяет установить максимальное количество защищаемых объектов;
- **Maximum ACL subjects to manage** — позволяет установить максимальное количество субъектов правил доступа LIDS;

- **Maximum ACL objects to manage** — позволяет установить максимальное количество объектов правил доступа LIDS;
- **Maximum protected proceeds** — позволяет установить максимальное количество защищаемых процессов;
- **Hang up console when raising securit alert** — разрешает закрытие консоли, с которой произошло нарушение безопасности;
- **Security alert when execing unprotected programs before sealing LIDS** — разрешает вывод сообщения о нарушении безопасности при запуске незащищенных программ;
- **Do not execute unprotected programs before sealing LIDS** — включает запрет на запуск незащищенных программ до установки способностей;
- **Try not to flood logs** — при включении этой опции LIDS не будет записывать в log-файлы дублирующиеся сообщения об одном и том же нарушении защиты;
- **Autorized time between two identic logs (seconds)** — устанавливается время в секундах, в течение которых проверяется появление двух идентичных сообщений, чтобы не записывать одинаковые сообщения в log-файлы;
- **Allow switching LIDS protections** — включает возможность отключения и включения LIDS в процессе работы системы после ввода пароля. При включении данной опции появляется возможность поменять любые параметры работы без перезагрузки операционной системы;
- **Numbers of attempts to submit password** — определяет количество попыток ввода пароля, по истечении которых отключение LIDS становится невозможным на заданный далее промежуток времени;
- **Time to wait after fail (seconds)** — время в секундах, в течение которого после ввода неправильного пароля указанное количество раз, отключение LIDS становится невозможным;
- **Allow remote users to switch LIDS protections** — дает возможность удаленным пользователям отключать LIDS. С целью увеличения безопасности вашей операционной системы не включайте эту опцию;
- **Allow any program to switch LIDS protections** — позволяет любой программе отключать LIDS. Не включайте эту опцию;
- **Allow reloading config. File** — разрешает переконфигурирование LIDS без перезагрузки компьютера;
- **Port Scanner Detector in kernel** — позволяет в ядро операционной системы добавить детектор сканирования портов;
- **Send security alerts through network** — разрешает отправку электронной почты при нарушении безопасности на указанный электронный адрес с информацией о нарушении. Письмо отправляется незамедлительно при попытке совершения несанкционированных действий;

- **Hide klids network threads** — позволяет скрывать сетевые соединения LIDS;
- **Number of connection tries before giving up** — задается количество попыток соединения с SMTP-сервером;
- **Sleep time after a failed connection** — задает время в секундах между попытками соединения с почтовым сервером;
- **Message queue size** — определяет максимальное количество почтовых сообщений в очереди. При превышении данного количества самое старое неотправленное сообщение удаляется из очереди;
- **LIDS debug** — используется для включения вывода отладочных сообщений LIDS.

После конфигурирования можно компилировать и устанавливать ядро операционной системы.

Конфигурирование LIDS

После установки LIDS в каталоге /etc появляется каталог lids, содержащий следующие конфигурационные файлы:

- lids.cap — предназначен для хранения текущих значений установок способностей;
- lids.net — предназначен для настройки отправки электронных сообщений системой LIDS;
- lids.pw — в этом файле записан в зашифрованном виде пароль администратора. Изменять этот файл можно только с помощью lidsadm;
- lids.conf — файл содержит текущие установки правил доступа. Изменять этот файл можно только с помощью lidsadm.

Способности

Способности (capabilities) — определяют возможность программ совершать какие-либо действия. LIDS позволяет использовать по отношению к программам большое количество способностей. В частности, LIDS поддерживает способность перезагружать компьютер, изменять владельца файла, загружать или выгружать модули ядра и многое другое.

Текущие установки способностей хранятся в файле lids.cap в формате:

```
[+|-] Номер:Способность
```

Здесь:

- `+` — включает способность;
- `-` — отключает способность.

Редактировать файл lids.cap можно с помощью любого текстового редактора. Включение способности влияет на все программы без исключения, а выключение влияет на все программы, кроме тех, которым напрямую указана данная способность с помощью правил доступа lidsadm.

Сразу после установки LIDS файл lids.cap содержит включенными следующие способности:

- CAP_CHOWN — устанавливает способность программ изменять владельца и группу владельца файла;
- CAP_DAC_OVERRIDE — разрешает программам, запускаемым пользователем root, не принимать во внимание режимы доступа к файлам. При отключении этой способности пользователь root теряет возможность изменять любые файлы, невзирая на права доступа;
- CAP_DAC_READ_SEARCH — то же самое, что и предыдущая способность, только по отношению к каталогам;
- CAP_FOWNER — разрешает операции с файлами, когда владелец файла должен совпадать с пользователем, совершающим операцию;
- CAP_FSETID — разрешает установку SUID- или SGID-бита на файлах, не принадлежащих пользователю root;
- CAP_KILL — разрешает процессам пользователя root "убивать" чужие процессы;
- CAP_SETGID — управляет способностью программ пользователя root изменять группу, под которой работает программа;
- CAP_SETUID — управляет способностью программ пользователя root изменять пользователя, под которым работает программа;
- CAP_SETPCAP — разрешает программам менять способности;
- CAP_LINUX_IMMUTABLE — управляет способностью снимать атрибуты S_IMMUTABLE и S_APPEND с файлов;
- CAP_NET_BIND_SERVICE — разрешает программам использовать сетевой порт, меньший чем 1024;
- CAP_NET_BROADCAST — управляет способностью программ рассылать широковещательные пакеты;
- CAP_NET_ADMIN — параметр управляет большим количеством различных способностей: конфигурирование сетевых интерфейсов, изменение правил брандмауэра, изменение таблиц маршрутизации и многих других, связанных с сетевыми настройками Linux;
- CAP_NET_RAW — управляет способностью программ использовать сокеты;
- CAP_IPC_LOCK — управляет способностью процессов пользователя root блокировать сегменты разделяемой памяти;

- CAP_IPC_OWNER — управляет доступом программ пользователя root к ресурсам межпроцессорного взаимодействия процессов, не принадлежащих пользователю root;
- CAP_SYS_MODULE — управляет способностью загружать модули ядра;
- CAP_SYS_RAWIO — управляет доступом на чтение/запись к таким устройствам, как /dev/mem, /dev/kmem, /dev/port, /dev/hdXX, /dev/sdXX;
- CAP_SYS_CHROOT — управляет способностью устанавливать корневой каталог для текущей командной оболочки;
- CAP_SYS_PTRACE — этот параметр включает способность программ использовать вызов функции ptrace(), которая позволяет управлять выполнением процессов-потомков процессу-родителю;
- CAP_SYS_PACCT — управляет способностью конфигурировать учет процессов;
- CAP_SYS_ADMIN — управляет множеством способностей: управление устройством /dev/random, создание новых устройств, конфигурирование дисковых квот, настройка работы klogd, установка имени домена, установка имени хоста, сброс кэша, монтирование и размонтирование дисков, включение/отключение swap-раздела, установка параметров последовательных портов и многое другое;
- CAP_SYS_BOOT — управляет способностью перегружать систему;
- CAP_SYS_NICE — управляет способностью изменять приоритет процессов, не принадлежащих пользователю root;
- CAP_SYS_RESOURCE — управляет способностью изменять лимиты использования ресурсов системы: дисковые квоты, зарезервированное пространство на Ext2-разделах, максимальное количество консолей и т. п.;
- CAP_SYS_TIME — управляет способностью изменять системное время;
- CAP_SYS_TTY_CONFIG — управляет способностью изменять настройки tty-устройств;
- CAP_HIDDEN — управляет способностью программ делаться невидимыми в списке процессов. Не влияет на все программы;
- CAP_INIT_KILL — управляет способностью "убивать" процессы-потомки процесса init.

Как видите, впечатляющий набор возможностей. Самое время разобраться, что из этого нужно включить, а что выключить для вашей операционной системы.

Для инициализации параметров способностей в процессе загрузки используется команда

```
lidsadm -I
```

Обычно ее ставят в конце /etc/rc.d/rc.local, что позволяет произвести отключение способностей только после запуска всех необходимых для работы сервера программ.

Правила доступа

Все управление LIDS осуществляется с помощью программы — lidsadm. Lidsadm работает в двух режимах — настройки правил доступа или ввода команд администрирования. Установки правил доступа находятся в файле /etc/lids/lids.conf. Для просмотра текущих установок правил доступа необходимо выполнить следующую команду:

```
lidsadm -L
LIST
Subject ACCESS TYPE Object
-----------------------------------------------------
Any File READ /sbin
Any File READ /bin
Any File READ /boot
Any File READ /lib
Any File READ /usr
Any File DENY /etc/shadow
/bin/login READ /etc/shadow
/bin/su READ /etc/shadow
Any File APPEND /var/log
Any File WRITE /var/log/wtmp
/sbin/fsck.ext2 WRITE /etc/mtab
Any File WRITE /etc/mtab
Any File WRITE /etc
/usr/sbin/sendmail WRITE /var/log/sendmail.st
/bin/login WRITE /var/log/lastlog
/bin/cat READ /home/xhg
Any File DENY /home/httpd
/usr/sbin/httpd READ /home/httpd
Any File DENY /etc/httpd/conf
/usr/sbin/httpd READ /etc/httpd/conf
/usr/sbin/sendmail WRITE /var/log/sendmail.st
/usr/X11R6/bin/XF86_SVGA NO_INHERIT RAWIO
/usr/sbin/in.ftpd READ /etc/shadow
/usr/sbin/httpd NO_INHERIT HIDDEN
```

Правила доступа состоят из трех элементов: субъекта, объекта и цели. Объектом является любой файл или каталог, на который и должны действовать правила доступа и защита LIDS. Если в качестве объекта указывается каталог, то все файлы в нем и вложенные каталоги с их файлами автоматически становятся объектами. Субъектом является любая защищенная программа, которой дают доступ к защищаемому объекту, поэтому прежде чем использовать программу в качестве субъекта, ее саму надо защитить средствами LIDS, применив к ней правила доступа как к объекту. Если субъект не указан, то субъектом является любая программа. Целью является тип доступа:

- READ — доступ на чтение;
- WRITE — запись;
- DENY — запрет на какой-либо доступ;
- APPEND — открытие только для записи в конец файла;
- IGNORE — игнорирование защиты.

Построение прав доступа подробно описано в документации на пакет LIDS, поэтому мы на этом здесь не останавливаемся.

После настройки LIDS необходимо перезагрузить операционную систему. В том случае, если с функционированием LIDS возникли проблемы, можно загрузить Linux с выключенным LIDS, для чего при загрузке необходимо передать ядру операционной системы параметр `security=0`. Например, для LILO это будет выглядеть так:

```
LILO boot: linux security=0
```

Tripwire

Программный пакет tripwire предназначен для обнаружения изменения файлов, позволяя обнаруживать порчу данных и взломы. База данных контрольных сумм файлов шифруется, что предотвращает ее подделку взломщиками.

Непосредственно после установки операционной системы необходимо установить tripwire, которая, используя правила, определенные политикой безопасности, создает базу данных, содержащую информацию обо всех файлах в системе (список файлов может задаваться администратором) — размер, контрольная сумма, дата модификации и т. п. После создания базы данных она ежедневно сравнивается с текущим состоянием файловой системы, позволяя обнаружить добавленные, измененные и удаленные файлы. Получаемые при этом отчеты могут быть просмотрены с различной степенью детализации.

Пакет tripwire входит в состав практически всех современных дистрибутивов Linux.

AIDE

Пакет AIDE — система обнаружения вторжений, основанная на использовании мониторинга изменения контрольных сумм защищаемых файлов операционной системы. Система AIDE разработана таким образом, что полная инсталляция ее помещается на одной дискете, что позволяет избежать вмешательства взломщика в функционирование программы.

Функционально программа является аналогом tripwire, только имеет более простые конфигурационные файлы и интерфейс.

Ссылки

- **acl.bestbits.at** — официальная страница проекта Linux ACLs (Access Control Lists).
- **bog.pp.ru/work/tripwire.html** — Bog BOS: Tripwire: принципы работы, установка и настройка.
- **freshmeat.net/projects/netramet/** — страница проекта NeTraMet.
- **gazette.linux.ru.net/lg75/articles/rus-maiorano.html** — Ariel Maiorano. Инсталляция и использование AIDE. Перевод А. Куприна.
- **linuxrsp.ru/artic/portsentry.html** — Ерижоков А. А. Portsentry.
- **linuxrsp.ru/artic/posixacls.html** — Ерижоков А. А. Списки контроля доступа.
- **linuxrsp.ru/artic/stunnel.html** — Ерижоков А. А. Stunnel: Шифрование трафика.
- **linuxsecurity.com** — сайт, посвященный безопасности операционной системы Linux.
- **rootshell.com** — сайт, посвященный безопасности операционных систем.
- **stunnel.mirt.net** — официальный сайт пакета Stunnel.
- **www.chkrootkit.org** — официальный сайт chkrootkit.
- **www.cs.tut.fi/~rammer/aide.html** — страница разработчика AIDE.
- **www.false.com/security/linux/** — Secure Linux patches by Solar Designer — дополнения к ядру Linux, повышающие безопасность операционной системы.
- **www.insecure.org** — местонахождение программы nmap — сканера сетевых портов.
- **www.lids.org** — сайт проекта LIDS.
- **www.linuxrsp.ru/artic/lids.html** — Ерижоков А. А. LIDS — система обнаружения и защиты от вторжения.

- **www.monkey.org/~dugsong/dsniff** — страничка программы-снифера Dsniff.
- **www.psionic.com** — сайт Psionic Software, разработчика программы Portsentry.
- **www.softerra.ru/freeos/16901/** — Oktay Altunergil. Понятие Rootkit. Перевод Инги Захаровой.
- **www.softerra.ru/freeos/16999/** — Oktay Altunergil. Сканирование для обнаружения Rootkit. Перевод Инги Захаровой.
- **www.softerra.ru/freeos/17032/** — Денис Колисниченко. Протоколирование.
- **www.tripwire.org** — сайт разработчиков Tripwire.
- **www.opennet.ru/docs/RUS/netramet/index.html** — Денис Матыцын. Сбор статистики по TCP/IP на базе NeTraMet.
- REFERENCE MANUAL NeTraMet & NeMaC. Nevil Brownlee.

Глава 28

Доступ к удаленным компьютерам

Мы уже неоднократно упоминали о том, что любая UNIX-подобная операционная система может удаленно администрироваться и конфигурироваться. Говорили мы и о возможности удаленной работы на компьютере под управлением Linux, а также и об администрировании через Web-интерфейс. Настало время познакомиться с *полным удаленным* администрированием, которое обеспечивает протокол Telnet и одноименные программа-сервер и программа-клиент.

Telnet

Под Telnet понимают трехкомпонентную систему, состоящую из:

- Telnet-клиента;
- Telnet-сервера;
- Telnet-протокола.

В общем, ничего оригинального. Клиент-серверная система, использующая свой протокол обмена. Начнем с протокола.

Протокол Telnet

Протокол Telnet описан в стандарте RFC854. Авторы стандарта определяют назначение Telnet следующим образом:

"Назначение Telnet-протокола — дать общее описание, насколько это только возможно, двунаправленного, восьмибитового взаимодействия, главной целью которого является обеспечение стандартного метода взаимодействия терминального устройства и терминал-ориентированного процесса. При этом протокол может быть использован и для организации взаимодействий "терминал-терминал" (связь) и "процесс-процесс" (распределенные вычисления)."

Telnet является протоколом приложения и использует транспортный протокол TCP. Протокол Telnet для обеспечения функциональности основан на следующих базовых концепциях:

- сетевого виртуального терминала (Network Virtual Terminal, NVT);
- согласования параметров взаимодействия;
- симметрии связи "терминал-процесс".

Рассмотрим эти концепции подробнее.

Сетевой виртуальный терминал позволяет абстрагироваться от реалий жизни. Удаленная программа может считать, что она использует один тип терминала, а у клиента может быть совершенно другой тип терминала. Сетевой виртуальный терминал — это стандартное описание наиболее широко используемых возможностей реальных физических терминальных устройств. Сетевой виртуальный терминал позволяет описать и преобразовать в стандартную форму способы отображения и ввода информации. Telnet-клиент и Telnet-сервер преобразовывают характеристики физических устройств в спецификацию сетевого виртуального терминала, что позволяет унифицировать характеристики физических устройств и обеспечить принцип совместимости устройств с разными возможностями. Характеристики диалога диктуются устройством с меньшими возможностями.

В протоколе Telnet сетевой виртуальный терминал определен как "двунаправленное символьное устройство, состоящее из принтера и клавиатуры". Принтер предназначен для отображения приходящей по сети информации, а клавиатура — для ввода данных, передаваемых по сети. По умолчанию предполагается, что для обмена информацией используется 7-битовый код ASCII, каждый символ которого закодирован в 8-битовое поле.

Согласование параметров взаимодействия позволяет унифицировать возможности представления информации на терминальных устройствах. Благодаря этой концепции можно использовать большинство возможностей современных терминалов. Обычно для этого существует специальная таблица соответствия, которая позволяет нестандартные команды терминала заменить стандартными наборами команд. Как правило, процесс согласования форм представления информации происходит в начальный момент организации Telnet-соединения. Каждый из процессов старается установить максимальные параметры сеанса. В UNIX-системах параметры терминалов содержатся в базе данных описания терминалов termcap. При инициировании Telnet-соединения обычно именно эти параметры используются в процессе согласования формы представления данных. При этом из одной системы в другую передается значение переменной окружения `TERM`. В процессе договора останутся только те функции, которые поддерживаются на обоих концах соединения.

Симметрия взаимодействия позволяет клиенту и серверу в течение одной сессии меняться ролями.

Команды Telnet

В табл. 28.1 приведены некоторые команды протокола Telnet с кратким пояснением. Как видно из таблицы, каждая команда представлена одним байтом, и чтобы различать команды и передаваемые данные, используется специальный признак команды.

Таблица 28.1. *Команды протокола Telnet*

Команда	Десятичное значение	Описание
EOF	236	Конец файла
SUSP	237	Подавить текущий процесс
ABORT	238	Прервать процесс
EOR	239	Конец записи
SE	240	Конец вспомогательной процедуры согласования
NOP	241	Нет операции
Data Mark	242	Часть потока данных для синхронизации
Break	243	Символ BRK виртуального терминала сети
Interrupt Process	244	Прервать текущий процесс
Abort Output	245	Отменить вывод без прекращения процесса
Are You There	246	Показывает, что связь не разорвана
Erase Character	247	Удалить предшествующий символ
Erase Line	248	Удалить текущую строку
Go Ahead	249	Запрос на ввод (для полудуплексных соединений)
SB	250	Начало вспомогательной процедуры согласования
WILL	251	Предложение начать выполнение факультативной команды (или подтверждение, что вы ее выполняете)
WON'T	252	Отказ выполнить (или продолжить выполнение)
DO	253	Требование к партнеру выполнить факультативную команду (или подтверждение ожидания выполнения другой стороной)

Таблица 28.1 *(окончание)*

Команда	Десятичное значение	Описание
DON'T	254	Требование к партнеру прекратить выполнение факультативной команды (или подтверждение того, что больше не ожидается выполнение операции другой стороной)
IAC	255	Интерпретировать как команду

Протокол Telnet предусматривает единый TCP-канал и для данных пользователя, и для управления. Поскольку по протоколу Telnet команды управления чередуются с данными, командам должен предшествовать специальный символ, называемый IAC (Interpret as Command, интерпретировать как команду) с кодом 255. В том случае, если необходимо передать символ данных с десятичным кодом 255, следует его передать дважды.

Команды протокола Telnet имеют размер не менее двух байтов. Первый из них всегда символ перехода в командный режим — IAC. Второй — команда протокола Telnet.

Программа-клиент telnet

Программа telnet — стандартный Telnet-клиент, входящий во все операционные системы UNIX-семейства и в практически все операционные системы Windows.

Для подключения к удаленной системе обычно используется команда вида:

```
telnet имя_хоста
```

Основные команды программы telnet приведены в табл. 28.2.

Таблица 28.2. *Команды программы telnet*

Команда	Описание
Open host [port]	Начать telnet-сессию с машиной host по порту port. Адрес машины можно задавать как в форме IP-адреса, так и в форме доменного адреса
close	Завершить telnet-сессию и вернуться в командный режим
Quit	Завершить работу программы telnet
Z	"Заморозить" telnet-сессию и перейти в режим интерпретатора команд локальной системы. Из этого режима можно выйти по команде Exit

Таблица 28.2 *(окончание)*

Команда	Описание
`Mode type`	Если значение `type line`, то используется буферизованный обмен данными, если `character` — обмен небуферизованный
`? [command] help [command]`	Список команд или описание конкретной команды
`Send argument`	Данная команда используется для ввода команд и сигналов протокола Telnet, которые указываются в качестве аргумента

Программа-сервер telnetd

Программа telnetd — это сервер, который обслуживает протокол Telnet. Программа telnetd обслуживает TCP-порт 23, но ее можно сконфигурировать на использование другого порта.

При установке взаимодействия с удаленным клиентом telnetd обменивается командами настройки — включение режима эха, обмен двоичной информацией, тип терминала, скорость обмена, переменные окружения.

Применение Telnet и безопасность

Протокол Telnet долгие годы был единственной универсальной возможностью удаленно работать с различными консольными программами. По своей простоте, нетребовательности к ресурсам и полосе пропускания он до сих пор не имеет себе равных. Помимо этого, клиентская программа telnet позволяет устанавливать соединения и с другими сервисами — например с почтовым сервером SMTP или POP3, что дает возможность производить различные манипуляции (например, просмотреть без почтового клиента содержимое своего почтового ящика или отправить письмо). Однако при всех его достоинствах протокол Telnet имеет один огромный недостаток — весь трафик пересылается в открытом виде. Из-за этого любому злоумышленнику не составляет труда перехватить логин и пароль пользователя, а также другую информацию. В современном мире, где новые вирусы и троянские программы возникают ежедневно, а скандальные взломы различных серверов случаются еженедельно, использование протокола Telnet недопустимо. В качестве альтернативы протоколу Telnet используются программные пакеты SSH или OpenSSH. Но о них чуть позже.

Семейство r-команд

Приблизительно в то же время, что и протокол Telnet, были созданы программы, предназначенные для удаленного администрирования и работы, так называемые r-команды (remote-команды).

Команда *rlogin*

Команда `rlogin` (remote login) предназначена для захода удаленным терминалом между UNIX-хостами. Стандарт RFC1282 содержит спецификацию протокола rlogin. Программа rlogin использует одно TCP-соединение между клиентом и сервером. Для нормального функционирования необходимо создать файл .rhosts, содержащий список хостов и пользователей, которым разрешено удаленно регистрироваться в системе. Каждая строка представляет собой пару "хост—пользователь", разделенную пробелом.

Команда *rsh*

Команда `rsh` (remote shell) используется для запуска командной оболочки на удаленном компьютере, после чего на нем возможно выполнять различные программы.

Команда *rcp*

Команда `rcp` (remote copy) используется для копирования файлов между компьютерами, причем эта операция может производиться между двумя удаленными компьютерами. Данная команда может копировать как один файл, так и группу файлов или каталогов.

Команда *rsync*

Команда `rsync`, аналогично команде `rcp`, позволяет копировать файлы между удаленными компьютерами. Однако, в отличие от `rcp`, может значительно ускорить процесс копирования часто изменяемых пользователем файлов, поскольку благодаря используемым алгоритмам передает только измененные части файлов. Также позволяет копировать ссылки (links), специальные устройства (device), владельца и группу файла, права доступа.

Команда *rdist*

Эта команда позволяет осуществить массовую автоматическую рассылку файлов с локального хоста на большую группу хостов с проверкой наличия места, рассылкой извещений о проблемах, исполнением завершающих процедур и т. п. Сохраняет имя владельца, имя группы, права доступа и время модификации файла.

Применение r-команд и безопасность

Как и в ситуации с Telnet, r-команды имеют ту же проблему с безопасностью — абсолютно незащищенную передачу информации, поэтому использование r-команд категорически не рекомендуется.

SSH и OpenSSH

Протокол SSH обеспечивает возможность удаленного выполнения команд и копирования файлов с аутентификацией клиента и сервера и шифрованием передаваемых данных, в том числе имени и пароля пользователя. Дополнительно обеспечивается шифрование данных X Windows и перенаправление любых TCP-соединений. Существует несколько программных реализаций, в частности, коммерческий SSH и бесплатный пакет с открытым исходным кодом OpenSSH.

Принцип работы SSH

SSH представляет собой протокол транспортного уровня, аутентификации и соединения и программные средства безопасного доступа к компьютерам по небезопасным каналам связи (telnet, X11, rsh, FTP). Аутентификация производится с использованием асимметричного шифрования с открытым ключом (SSH1 — RSA, SSH2 — RSA/DSA). Обмен данными — симметричное шифрование. Целостность переданных данных проверяется с помощью специальных контрольных сумм. Протокол транспортного уровня работает поверх TCP и использует 22-й порт. В качестве ключа берется случайная строка, которую генерирует клиент, шифрует с помощью открытого ключа сервера — и передает серверу. Протокол аутентификации работает поверх протокола транспортного уровня и обеспечивает аутентификацию клиента для сервера. Шифрование трафика начинается после аутентификации сервера, но до аутентификации клиента, таким образом пароли в открытом виде не передаются. Возможно соединение произвольных портов TCP по защищенным каналам. Предусматривается возможность сжатия.

Существует две версии протокола — SSH1 и SSH2. По своей реализации — совершенно разные протоколы. Протокол SSH2 был разработан с учетом найденных в первом варианте уязвимостей.

OpenSSH

Некоммерческая реализация протокола SSH с открытым кодом. Программный пакет способен работать с протоколами SSH1 и SSH2. Имеется также поддержка r-команд.

Для дистрибутива Red Hat установка пакета OpenSSH включена в стандартную инсталляцию и никаких проблем не вызывает.

Конфигурирование OpenSSH

Конфигурирование OpenSSH очень сильно зависит от вашей концепции обеспечения безопасности и необходимости поддержки старого типа протокола. Поскольку использование протокола SSH1 из-за найденных уязвимо-

стей не рекомендуется — при конфигурировании необходимо запретить его использование. Также рекомендуется запретить использование r-команд и всего, что с ними связано. При конфигурировании OpenSSH необходимо произвести настройку сервера и клиента. Конфигурационный файл сервера называется sshd_config, а клиента — ssh_config.

Файл sshd_config

Файл sshd_config задает параметры SSH-серверу и может содержать внушительный список различных параметров. Ниже приведены его основные конфигурационные параметры:

- `AllowGroups список-имен-групп-через-пробел` — вход разрешен только пользователям, чья группа входит в этот список;
- `AllowTcpForwarding yes/no` — разрешает или запрещает TCP Forwarding;
- `AllowUsers список-имен-через-пробел` — вход разрешен только перечисленным пользователям;
- `AuthorizedKeysFile имя-файла-с-публичным ключом` — задает имя файла, содержащее публичный ключ;
- `Banner сообщение-перед-аутентификацией` — текст сообщения, выводимого сервером перед аутентификацией клиента;
- `Ciphers` — список алгоритмов симметричного шифрования для SSH2: `aes128-cbc`, `3des-cbc`, `blowfish-cbc`, `cast128-cbc`, `arcfour`;
- `ClientAliveInterval секунд` — определяет интервал в секундах, через который сервер будет производить проверку, произошло или нет отключение клиента;
- `ClientAliveCountMax число` — данный параметр определяет число неудачных проверок существования связи с пользователем до разрыва сессии;
- `DenyGroups список-имен-групп-через-пробел` — определяет список групп пользователей, которым запрещено устанавливать соединение с сервером;
- `DenyUsers список-имен-через-пробел` — определяет список пользователей, которым запрещено устанавливать соединение с сервером;
- `GatewayPorts no/yes` — данный параметр определяет, разрешать или нет удаленным хостам доступ к перенаправленным портам;
- `HostbasedAuthentication no/yes` — разрешить или запретить аутентификацию, используя имя хоста (только для SSH2);
- `HostKey имя-файла-содержащего-приватный-ключ` — с помощью данного параметра можно указать серверу, где расположен файл, содержащий секретный ключ шифрования;

- `IgnoreRhosts yes/no` — этот параметр позволяет определить, использовать или нет файлы .rhosts и .shosts для аутентификации. Для увеличения безопасности системы рекомендуется запретить использование этих файлов;
- `IgnoreUserKnownHosts no/yes` — параметр позволяет запретить использование файла ~/.ssh/known_hosts во время аутентификации rhosts+RSA;
- `KeepAlive yes/no` — параметр позволяет использовать механизм регулярных сообщений для проверки разрыва связи;
- `KerberosAuthentication yes/no` — параметр позволяет запретить использование kerberos при аутентификации;
- `KerberosOrLocalPasswd yes/no` — в том случае, если аутентификация через kerberos не прошла, данный параметр позволяет использовать /etc/passwd для аутентификации;
- `KeyRegenerationInterval 3600` — параметр задает интервал регенерации ключа сервера;
- `ListenAddress 0.0.0.0` — параметр определяет, к каким адресам прислушиваться; при использовании необходимо также определить параметр `Port`;
- `LoginGraceTime секунд` — данный параметр определяет, через сколько секунд произойдет разрыв соединения, если при аутентификации пользователь за это время не введет пароль;
- `LogLevel INFO` — параметр определяет, какой уровень использовать при создании сообщений в журнал системы. Можно использовать следующие уровни — `QUIET`, `FATAL`, `ERROR`, `INFO`, `VERBOSE`, `DEBUG`;
- `MACs алгоритмы-проверки-целостности-данных` — этот параметр определяет, какой алгоритм будет использоваться для проверки целостности данных — `hmac-md5`, `hmac-sha1`, `hmac-ripemd160`, `hmac-sha1-96`, `hmac-md5-96`;
- `MaxStartups 10` — данный параметр задает максимально возможное количество соединений, ожидающих аутентификации;
- `PasswordAuthentication yes/no` — параметр разрешает аутентификацию по паролю;
- `PermitEmptyPasswords no/yes` — параметр разрешает использование пустых паролей;
- `PermitRootLogin yes/no/without-password/forced-commands-only` — параметр разрешает пользователю root подключаться к серверу;
- `PidFile имя-файла` — параметр задает имя файла, в котором будет храниться PID процесса сервера;
- `Port 22` — параметр определяет, какой порт слушает сервер;

- `PrintMotd yes/no` — параметр разрешает использование /etc/motd при входе пользователя в систему для выдачи сообщения;
- `Protocol 2` — параметр определяет, с какой версией протокола работает сервер;
- `PubkeyAuthentication yes/no` — параметр разрешает использовать публичный ключ при аутентификации;
- `ReverseMappingCheck no/yes` — параметр разрешает после определения адреса по имени хоста производить проверку того, что обратная зона для этого адреса указывает на тот же самый хост;
- `RhostsAuthentication no/yes` — параметр разрешает аутентификацию только на основании файлов .rhosts или /etc/hosts.equiv;
- `RhostsRSAAuthentication no/yes` — параметр разрешает аутентификацию на основе .rhosts- и RSA-аутентификации;
- `RSAAuthentication yes/no` — данный параметр используется только для протокола SSH1;
- `ServerKeyBits 768` — данный параметр определяет длину ключа;
- `StrictModes yes/no` — параметр разрешает проверять права доступа к файлам с частными паролями при запуске;
- `SyslogFacility AUTH` — параметр задает тип сообщений, передаваемых на syslog: `DAEMON`, `USER`, `AUTH`, `LOCAL0`, `LOCAL1`, `LOCAL2`, `LOCAL3`, `LOCAL4`, `LOCAL5`, `LOCAL6`, `LOCAL7`;
- `UseLogin no/yes` — параметр разрешает использовать login для интерактивных сессий;
- `X11DisplayOffset 10` — параметр определяет первый доступный номер дисплея при передаче X11.

Файлы на сервере, используемые при входе SSH

При входе SSH на сервере используются следующие файлы:

- /etc/nologin — при наличии этого файла запрещается вход пользователей, кроме root. Содержимое файла выдается в качестве сообщения о причине;
- /etc/hosts.allow — при компиляции с libwrap используется для разрешения доступа;
- /etc/hosts.deny — при компиляции с libwrap используется для запрещения доступа;
- ~/.rhosts — файл содержит пары "хост—пользователь", разделенные пробелом. Для указанного пользователя с данного хоста разрешается заходить без ввода пароля при использовании `RhostsAuthentication` и `RhostsRSAAuthentication`. Также используется семейством r-команд;

- ~/.shosts — аналогично файлу .rhosts, но не используется семейством r-команд;
- /etc/hosts.equiv — список хостов, пользователи с которых могут заходить, не указывая паролей, под теми же самыми именами. За именем хоста можно указывать имя конкретного пользователя. Также используется семейством r-команд;
- /etc/shosts.equiv — аналогично файлу hosts.equiv, но не используется семейством r-команд;
- ~/.ssh/environment — содержит пары вида `имя=значение`, которые помещаются в окружение при входе.

Файлы ключей сервера

В качестве ключей сервера используются следующие файлы:

- /usr/local/etc/ssh_host_key — приватный ключ хоста;
- /usr/local/etc/ssh_host_rsa_key — приватный ключ хоста, алгоритм шифрования RSA;
- /usr/local/etc/ssh_host_dsa_key — приватный ключ хоста, алгоритм шифрования DSA;
- /usr/local/etc/ssh_host_key.pub — публичный ключ хоста;
- /usr/local/etc/ssh_host_rsa_key.pub — публичный ключ хоста, алгоритм шифрования RSA;
- /usr/local/etc/ssh_host_dsa_key.pub — публичный ключ хоста, алгоритм шифрования DSA.

Файл ssh_config

Файл предназначен для конфигурации SSH-клиента и разделен на секции директивами `Host`. Секция применяется при работе с хостом, удовлетворяющим шаблону секции:

- `Host шаблоны` — следующие опции применимы к хостам, подходящим под один из шаблонов; имя хоста берется из командной строки, в шаблонах используются символы `*` и `?`;
- `BatchMode no|yes` — параметр разрешает не запрашивать пароль/парольную фразу;
- `CheckHostIP yes|no` — позволяет дополнительно проверять адрес сервера в known_hosts;
- `Cipher 3des|blowfish` — этот и следующий параметры определяют алгоритм шифрования данных;
- `Ciphers aes128-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, arcfour, aes192-cbc, aes256-cbc`;

- `ClearAllForwardings no|yes` — данный параметр позволяет сбросить все перенаправления портов;
- `Compression no|yes` — параметр разрешает производить сжатие передаваемых данных;
- `CompressionLevel уровень-сжатия` — параметр определяет уровень компрессии для протокола SSH1;
- `ConnectionAttempts число-попыток-соединения` — параметр задает число попыток установления соединения;
- `EscapeChar символ|^символ|none` — параметр позволяет определить символ для использования вместо тильды;
- `FallBackToRsh no|yes` — параметр разрешает использовать rsh в том случае, если сервер не имеет SSH-сервера;
- `ForwardAgent no|yes` — параметр определяет, передавать ли запрос к агенту аутентификации на удаленный хост;
- `GatewayPorts no|yes` — параметр разрешает удаленным хостам соединяться на перенаправленные локальные порты;
- `GlobalKnownHostsFile имя-файла` — параметр разрешает использовать указанный файл вместо /usr/local/etc/ssh_known_hosts;
- `HostKeyAlgorithms ssh-rsa,ssh-dss` — параметр определяет используемые алгоритмы шифрования (SSH2);
- `IdentityFile имя-файла` — параметр определяет файл, хранящий RSA- или DSA-приватный ключ;
- `KeepAlive yes|no` — параметр позволяет заметить разрыв связи или аварийное завершение на удаленном конце;
- `KerberosAuthentication yes|no` — параметр разрешает использовать kerberos-аутентификацию;
- `LogLevel INFO` — параметр определяет, какой уровень использовать при создании сообщений в журнал системы. Можно использовать следующие уровни — `QUIET`, `FATAL`, `ERROR`, `INFO`, `VERBOSE`, `DEBUG`;
- `MACs hmac-md5, hmac-sha1, hmac-ripemd160, hmac-sha1-96, hmac-md5-96` — параметр определяет используемые алгоритмы для создания контрольной суммы;
- `NumberOfPasswordPrompts 3` — параметр определяет количество попыток ввода пароля пользователя;
- `PasswordAuthentication yes/no` — параметр разрешает аутентификацию по паролю;
- `Port 22` — параметр определяет, к какому порту будет подключаться клиент;

- `PreferredAuthentications publickey, password, keyboard-interactive` — параметр определяет приоритеты аутентификации (SSH2);
- `Protocol список-версий-протокола` — параметр задает список версий протокола в порядке предпочтительности;
- `ProxyCommand` — использование этого параметра позволяет использовать дополнительную команду для соединения с сервером;
- `PubkeyAuthentication yes|no` — параметр разрешает использовать при аутентификации публичный ключ (SSH2);
- `RhostsAuthentication yes|no` — параметр разрешает использовать при аутентификации файл .rhosts (SSH1);
- `StrictHostKeyChecking ask|no|yes` — параметр разрешает не добавлять незнакомые или изменившиеся хосты в know_hosts;
- `UsePrivilegedPort yes|no` — параметр разрешает использовать привилегированные порты для установления соединения;
- `User имя-пользователя` — параметр задает имя пользователя;
- `UserKnownHostsFile файл-known_hosts` — параметр определяет местоположение файла known_hosts;
- `UseRsh no|yes` — параметр разрешает использовать rsh в том случае, если SSH на хосте отсутствует.

Файлы ключей клиента

В качестве ключей клиента используются следующие файлы:

- ~/.ssh/identity — приватный RSA1-ключ пользователя;
- ~/.ssh/id_dsa — приватный DSA2-ключ пользователя;
- ~/.ssh/id_rsa — приватный RSA2-ключ пользователя;
- ~/.ssh/identity.pub — публичный RSA1-ключ пользователя;
- ~/.ssh/id_dsa.pub — публичный DSA2-ключ пользователя;
- ~/.ssh/id_rsa.pub — публичный RSA2-ключ пользователя.

Ключи запуска сервера SSH

Помимо конфигурационного файла, некоторые особенности функционирования сервера SSH можно задать, используя ключи запуска. Ниже приведены основные ключи запуска:

- `-D` — не отсоединяться от терминала при запуске;
- `-b bits` — ключ задает число битов ключа сервера (SSH1), по умолчанию 768;

- `-d` — переводит сервер в отладочный режим, использование нескольких ключей увеличивает количество отладочной информации;
- `-e` — ключ разрешает выводить сообщения на stderr вместо syslog (то есть не журналировать сообщения, а сразу выводить на стандартное устройство для вывода ошибок — обычно это текстовый терминал);
- `-f имя-конфигурационного-файла` — ключ определяет положение конфигурационного файла, удобно использовать при отладке;
- `-g grace-time` — ключ определяет время ожидания между вводом логина и пароля пользователя;
- `-h файл-ключей-хоста` — ключ определяет местоположение файла ключей;
- `-k keygen-time` — параметр задает интервал регенерации ключа сервера;
- `-p порт` — параметр определяет порт, который будет слушать сервер;
- `-q` — ключ запрещает выдачу информации на syslog (то есть запрщает журналирование событий);
- `-t` — с помощью этого параметра можно произвести проверку на отсутствие ошибок в конфигурационном файле и ключах;
- `-u число` — вместо имен хостов, превышающих число, в журнале utmp будет записываться IP-адрес: `-u0` вызывает безусловную запись IP-адресов;
- `-4` — использование протокола IPv4;
- `-6` — использование протокола IPv6.

Ключи запуска клиента SSH

Как и для сервера, для изменения некоторых параметров клиента можно воспользоваться ключами запуска:

- `-a` — запретить перенаправление агента аутентификации;
- `-A` — разрешить перенаправление агента аутентификации;
- `-b адрес` — позволяет для хоста с несколькими интерфейсами использовать конкретный адрес;
- `-c blowfish|3des` — задает используемый алгоритм шифрования (SSH1);
- `-c список-алгоритмов-шифрования-через-запятую` — алгоритм в начале списка имеет наибольший приоритет; по умолчанию: `aes128-cbc`, `3des-cbc`, `blowfish-cbc`, `cast128-cbc`, `arcfour`, `aes192-cbc`, `aes256-cbc` (SSH2);
- `-D локальный-порт` — эмуляция SOCKS4-сервера (специфического proxy-сервера) по защищенному каналу;

- `-e символ | ^символ | none` — определяет escape-символ вместо тильды; `none` обеспечивает прозрачную передачу данных;
- `-f` — перейти в фоновый режим после запроса пароля или парольной фразы;
- `-F имя-конфигурационного-файла` — разрешает использовать указанный файл в качестве конфигурационного;
- `-g` — разрешать удаленному хосту подсоединяться к локальным перенаправленным портам;
- `-i имя-файла` — определяет файл, хранящий RSA/DSA-приватный ключ;
- `-k` — запретить перенаправление Kerberos;
- `-l имя-пользователя` — от имени какого пользователя устанавливается соединение;
- `-m список-алгоритмов-обеспечения-целостности-соединения` — определяет алгоритмы подсчета контрольной суммы;
- `-n` — направить /dev/null на stdin и перейти в фоновый режим;
- `-p порт` — соединиться с указанным хостом на удаленном хосте;
- `-P` — использовать непривилегированный порт для исходящего соединения, чтобы обойти ограничения сетевого экрана;
- `-R локальный-порт:хост:удаленный-порт` — если происходит соединение на удаленный порт, то оно перенаправляется по защищенному каналу на локальный порт;
- `-s` — запуск подсистемы на сервере — например, sftp; имя подсистемы задается последним параметром;
- `-t` — требовать выделения псевдо-tty;
- `-T` — не выделять псевдо-tty;
- `-x` — запретить перенаправление X11;
- `-X` — разрешить перенаправление X11;
- `-1` — использовать только SSH1-протокол;
- `-2` — использовать только SSH2-протокол;
- `-4` — использовать IPv4;
- `-6` — использовать IPv6.

Программы, входящие в пакет OpenSSH

Помимо клиента и сервера, в пакет OpenSSH входят программы, предназначенные для генерации ключей, аутентификации, и программы, призванные заменить набор r-команд.

Программа ssh-keygen

Программа ssh-keygen предназначена для генерации, преобразования и управления ключами. По умолчанию генерирует RSA-ключ. При генерации запрашивается парольная фраза. Забытую парольную фразу восстановить невозможно. Число битов по умолчанию — 1024. Имя файла для хранения публичного ключа образуется из имени файла для частного ключа добавлением суффикса .pub. Ключ хоста должен иметь пустую парольную фразу.

Возможные строки запуска:

- генерирует ключ по указанному пользователем алгоритму:

```
ssh-keygen [-t rsa1|dsa|rsa] [-b бит] [-N парольная-фраза]
           [-C комментарий] [-f имя-файла-записи] [-q]
```

- изменить комментарий:

```
ssh-keygen -c [-P парольная-фраза] [-C комментарий]
              [-f файл-с-ключами]
```

- читает приватный или публичный ключ в форматах OpenSSH и преобразует его в формат SECSH для экспорта в другие реализации SSH:

```
ssh-keygen -e [-f файл-с-ключами]
```

- читает приватный или публичный ключ в формате SSH2 или SECSH и преобразует его в формат OpenSSH:

```
ssh-keygen -i [-f файл-с-ключами]
```

- позволяет изменить парольную фразу:

```
ssh-keygen -p [-P старая-парольная-фраза] [-N новая-парольная-фраза]
              [-f файл-с-ключами]
```

- читает приватный ключ OpenSSH DSA и выдает публичный ключ OpenSSH DSA:

```
ssh-keygen -y [-f файл-с-ключами]
```

Программа ssh-agent

Программа ssh-agent позволяет производить RSA/DSA-аутентификацию. Запускается в начале сессии и устанавливает переменные окружения, с помощью которых остальные программы могут использовать ее для автоматической аутентификации SSH. Параметрами являются имя команды и ее аргументы, ssh-agent завершается при завершении команды. Если имя команды не указано, то ssh-agent запускается в фоновом режиме, а на stdout (стандартный выход, обычно текстовый терминал) выдаются команды экспортирования необходимых переменных окружения.

Опции командной строки ssh-agent:

- `-c` — позволяет выдавать на stdout команды в стиле csh;

- `-s` — позволяет выдавать на stdout команды в стиле sh;
- `-k` — завершить работу агента — по переменной `SSH_AGENT_PID`.

Программа ssh-add

Эта программа используется для добавления приватных ключей. Программа ssh-add запрашивает парольную фразу, расшифровывает приватный ключ и посылает его ssh-agent. Если терминал недоступен, но определена переменная `DISPLAY`, то для ввода парольной фразы используется программа, определенная переменной `SSH_ASKPASS`. Таким образом, парольная фраза запрашивается только один раз за сеанс, а не при каждом вызове ssh/scp/sftp.

Опции командной строки ssh-add:

- имя файла с приватным ключом; по умолчанию используется ~/.ssh/identity;
- `-L` — выдает публичные ключи, хранящиеся в ssh-add;
- `-d` — удалить приватный ключ;
- `-D` — удалить все ключи.

Программа sftp

Программа sftp (secure FTP) является клиентом для sftp-сервера, который должен быть описан в опции `Subsystem` в конфигурационном файле sshd.

Программа sftp позволяет пересылать файлы в режиме, подобном FTP-протоколу, однако осуществляет все операции поверх защищенного транспорта SSH. К сожалению, данный вариант FTP пока не получил широкого распространения.

Опции командной строки:

- `[user@]имя-хоста[:dir/]` — задает, аналогично FTP, имя пользователя, хост, к которому производится подключение, и каталог подключения;
- `-b имя-файла` — позволяет читать команды из файла вместо стандартного устройства ввода;
- `-C` — разрешает использовать сжатие пересылаемых файлов;
- `-F имя-конфигурационного-файла-ssh` — указывает, какой конфигурационный файл использовать;
- `-o опция` — передается SSH;

Интерактивные команды, используемые sftp, аналогичны FTP-командам:

- `bye` — разорвать соединение;
- `cd путь` — сменить каталог;
- `lcd путь` — сменить каталог;

- `chgrp gid имя-файла` — изменить групповой идентификатор файла на указанный в команде;
- `chmod mode имя-файла` — изменить атрибуты файла;
- `chown uid имя-файла` — изменить владельца файла;
- `exit` — выйти;
- `get [-P] имя-удаленного-файла [имя-локального-файла]` — команда для получения файла; ключ `-P` позволяет сохранить права и время создания и модификации получаемого файла;
- `help` — позволяет получить справку по командам;
- `lls [опции-ls [имя-файла]]` — получить список файлов;
- `lpwd` — пароль;
- `mkdir имя` — создать каталог;
- `put [-P] имя-локального-файла [имя-удаленного-файла]` — выгрузить на сервер файл; ключ `-P` позволяет сохранить права и время создания и модификации передаваемого файла;
- `pwd` — пароль;
- `quit` — выйти;
- `rename старое-имя новое-имя` — переименовать файл;
- `rmdir имя` — удалить каталог;
- `rm имя-файла` — удалить файл;
- `symlink старое-имя новое-имя` — создать символическую ссылку.

Программа scp

Программа scp является аналогом программы rcp и осуществляет копирование файлов между хостами, причем оба могут быть удаленными. Способы аутентификации аналогичны SSH. Вызывает SSH для организации канала передачи данных. Имя файла записывается в виде:

```
[[user@]host:]file
```

Опции командной строки:

- `-c алгоритм-шифрования` — передается SSH;
- `-i имя-файла` — файл с приватным ключом, передается в SSH;
- `-o опция` — передается SSH;
- `-p` — сохраняет время модификации, использования и права доступа к файлу;
- `-r` — позволяет рекурсивно копировать весь каталог;

- `-B` — пакетный режим — не запрашивать пароль или парольную фразу;
- `-C` — разрешает производить сжатие при передаче файла;
- `-F конфигурационный-файл` — определяет альтернативный конфигурационный файл;
- `-P port` — задает порт сервера;
- `-S программа` — разрешает использовать указанную программу вместо SSH;
- `-4` — использовать IPv4;
- `-6` — использовать IPv6.

Программа ssh-keyscan

Программа ssh-keyscan позволяет собрать публичные ключи хостов, имена хостов задаются в качестве параметров или в файле. Опрос производится параллельно. Опции командной строки:

- `-t тип-ключа` — задает тип шифрования ключа (rsa1, rsa, dsa);
- `-T секунд` — определяет тайм-аут;
- `-f имя-файла` — определяет файл, в котором каждая строка содержит имя или адрес хоста;
- `-4` — использовать IPv4;
- `-6` — использовать IPv6;
- `-p удаленный-порт` — определяет порт.

Ссылки

- RFC 854. Описание протокола Telnet.
- **www.bog.pp.ru/work/ssh.html** —Bog BOS: SSH и OpenSSH: принципы работы, установка и настройка.
- **www.ssh.com** — сайт коммерческой реализации SSH.
- **www.openssh.com** — сайт некоммерческой реализации SSH.
- **www.tigerlair.com/ssh/faq/** — SSH FAQ.
- **lib.ru/LABIRINT/telnet.htm** — доступ к ресурсам Интернета в режиме удаленного терминала.
- **www.mnet.uz/citforum/internet/services/index.shtml** — Храмцов П. Б. Администрирование сети и сервисов Internet. Учебное пособие.

Глава 29

Firewall

Эта глава посвящена одному из аспектов сетевой безопасности, а конкретно — защите сети от вторжения извне и изнутри. Для защиты локальной сети используется комплекс программного обеспечения, в литературе известный как firewall (брандмауэр), или межсетевой экран. Брандмауэр позволяет "отгородить" систему (или сеть) от жестокого внешнего мира. Он используется для предотвращения получения посторонними данных (или ресурсов) защищаемой сети, а также для контроля за внешними ресурсами, к которым имеют доступ пользователи вашей сети. Конечно, стопроцентной защищенности от проникновения извне или нарушения работоспособности вашей сети или сервисов не даст ни одна система, однако использование брандмауэра (при правильной его конфигурации) может сильно усложнить взломщику задачу.

Чаще всего брандмауэр — это программы маршрутизации и фильтрации сетевых пакетов. Такие программы позволяют определить, можно ли пропустить данный пакет и если можно, то отправить его точно по назначению. Для того чтобы брандмауэр мог сделать это, ему необходимо определить набор правил фильтрации. Главная цель брандмауэра — контроль удаленного доступа извне или изнутри защищаемой сети или компьютера.

Брандмауэр позволяет лишь частично решить проблемы, связанные с обеспечением безопасного функционирования вашей сети. Как бы хорошо он ни был настроен, если вы вовремя не обновили программный пакет, в котором была найдена уязвимость, или кто-то узнал ваши логин и пароль — ждите больших неприятностей. Основная задача брандмауэра — разрешать функционирование только тем службам, которым было явно разрешено работать в вашей сети или защищаемом компьютере. В результате мы получаем маленькую дверцу, через которую в уютный внутренний мирок могут попасть только те гости, которых пропустила ваша охрана, а список этих гостей рекомендуется сузить до минимального.

Основными компонентами брандмауэра являются:

- политика безопасности сети;

- механизм аутентификации;
- механизм фильтрации пакетов.

О практической реализации этих компонентов мы поговорим несколько позже, а пока разберемся, какие бывают брандмауэры.

Совет

Мы настоятельно рекомендуем ознакомиться с книгой Роберта Зиглера "Брандмауэры в Linux", в которой очень подробно и доходчиво объясняется конфигурирование брандмауэров.

Типы брандмауэров

При построении брандмауэра обычно используется компьютер (компьютеры), непосредственно подключенный к Интернету и содержащий базовый набор средств, реализующих брандмауэр. Такой компьютер иногда называют *бастионом*.

Термин "брандмауэр" может приобретать различные значения в зависимости от принципа, положенного в основу работы средств защиты, сетевой архитектуры и схемы маршрутизации. Брандмауэры обычно подразделяют на три типа:

- брандмауэр с фильтрацией пакетов;
- прикладной шлюз;
- универсальный proxy-сервер.

Брандмауэр с фильтрацией пакетов, как правило, действует на сетевом и транспортном уровнях и реализуется в составе операционной системы. Источником информации для фильтрации является содержимое заголовков IP-пакетов, на основе которого брандмауэр принимает решение, по какому маршруту следует направить пакет.

Прикладной шлюз реализуется посредством выбора сетевой архитектуры и конфигурации системы. Сетевой трафик никогда не проходит через компьютер, на котором выполняется прикладной шлюз. Чтобы получить доступ в Интернет, локальный пользователь должен зарегистрироваться на прикладном шлюзе. Компьютер, содержащий прикладной шлюз, может быть защищен брандмауэрами с фильтрацией пакетов как извне, так и из локальной сети.

Proxy-сервер обычно реализуется в виде независимого приложения, управляющего доступом к различным типам сетевых служб. Для клиентов proxy-сервер выполняет роль сервера, предоставляющего информацию. Вместо того чтобы непосредственно обращаться к удаленным серверам, клиентские программы обращаются к proxy-серверу. Получив обращение клиента, proxy-сервер устанавливает связь с удаленным узлом от своего имени, при этом он заменяет в пакете адрес клиента своим адресом. Подобный сервер

может контролировать целостность данных, осуществлять проверку на наличие вирусов и обеспечивать выполнение правил системной политики, определяющих обмен высокоуровневыми данными.

Помимо этого, брандмауэры можно разделить по типу построения защиты:

- пороговый и его разновидность — бастионного типа;
- организующий так называемую демилитаризованную зону.

Брандмауэр порогового типа призван защитить локальную сеть от атак извне, а при соответствующей настройке и от атак изнутри. Такого типа брандмауэры обычно используются для защиты небольшой сети или даже одного компьютера. Как правило, сетевые службы, предоставляющие услуги наружу (HTTP, FTP и т. п.), размещаются на том же компьютере, что и брандмауэр.

Организация демилитаризованной зоны оправдана тогда, когда в сети выделено несколько специальных компьютеров для интернет-сервисов, предоставляемых большому миру, а так же при отсутствии уверенности в благонадежности собственных сотрудников. Для организации демилитаризованной зоны используются, по меньшей мере, два брандмауэра — один для защиты демилитаризованной зоны от проникновения извне, а второй — от проникновения из вашей собственной локальной сети. Организация демилитаризованной зоны сложнее, чем организация брандмауэра бастионного типа, но взамен вы получаете большую защиту ваших данных.

Брандмауэр с фильтрацией пакетов

Брандмауэр с фильтрацией пакетов представляет собой "сито" для проходящих через него входящих и исходящих пакетов. В операционной системе Linux реализован брандмауэр, позволяющий контролировать ICMP-, UDP- и TCP-пакеты. Брандмауэр с фильтрацией пакетов организован как механизм, реализующий для входящих и исходящих пакетов набор разрешающих и запрещающих правил. Этот набор правил определяет, какие пакеты могут проходить через конкретный сетевой интерфейс.

Брандмауэр с фильтрацией пакетов может производить с проходящим пакетом всего три действия:

- переслать пакет на узел назначения;
- удалить пакет без уведомления посылающей пакет стороны;
- вернуть передающему компьютеру сообщение об ошибке.

Несмотря на такие простые действия, в большинстве случаев их достаточно для организации эффективной защиты. Как правило, брандмауэр устанавливается для того, чтобы контролировать данные, которыми компьютеры обмениваются с Интернетом. В результате работы фильтрующего брандмауэра

отсеиваются недопустимые обращения к узлам внутренней сети и запрещается передача из внутренней сети в Интернет для пакетов, определенных правилами фильтрации.

В целях получения более гибкой системы правила фильтрации пакетов составляются для каждого сетевого интерфейса, в них учитываются IP-адреса источника и назначения, номера портов TCP и UDP, флаги TCP-соединений и ICMP-сообщений. Причем правила для входящих и исходящих пакетов различаются. Это значит, что при настройке фильтрующего брандмауэра правила для конкретного сетевого интерфейса представляются как отдельные правила для входящей и исходящей информации, поскольку входящие и исходящие пакеты обрабатываются брандмауэром независимо друг от друга. Списки правил, которые управляют фильтрацией сетевых пакетов, поступающих извне в локальную сеть и отправляемых из локальной сети в Интернет, принято называть *цепочками* (chains). Термин "цепочка" используется потому, что при проверке пакета правила применяются последовательно одно за другим, пока не обнаружится подходящее правило для сетевого пакета или список правил не будет исчерпан.

Описанный механизм фильтрующего брандмауэра достаточно эффективен, однако он не обеспечивает полной безопасности локальной сети. Брандмауэр — это всего лишь один из элементов общей схемы защиты. Анализ заголовков сетевых пакетов — операция слишком низкого уровня для того, чтобы реально выполнять аутентификацию и контролировать доступ. В процессе фильтрации пакетов практически невозможно распознать отправителя сообщения и проанализировать смысл передаваемой информации. Из всего набора данных, пригодных для аутентификации, на рассматриваемом уровне доступен только IP-адрес отправителя, однако этот адрес очень легко подделать, на чем и базируется множество способов сетевых атак. Несмотря на то, что средства фильтрации пакетов позволяют эффективно контролировать обращение к портам, использование протоколов обмена и содержимое пакетов, проверку данных необходимо продолжить на более высоком уровне.

Политика организации брандмауэра

При построении брандмауэров используются два основных подхода:

- запрещается прохождение всех пакетов, пропускаются лишь те, которые удовлетворяют явно определенным правилам;
- разрешается прохождение всех пакетов, за исключением пакетов, удовлетворяющих определенным правилам.

Или, перефразируя, запрещено все, что не разрешено, и разрешено все, что не запрещено.

С практической точки зрения лучше использовать подход, при котором поступающий пакет по умолчанию отвергается (запрещено все, что не разрешено). В этом случае организация безопасности сети достигается достаточно просто, но с другой стороны, приходится предусматривать возможность обращения к каждой сетевой службе и использование каждого конкретного протокола. Это означает, что администратор сети, занимающийся настройкой брандмауэра, должен точно знать, какие протоколы применяются в его локальной сети. При использовании подхода, предусматривающего запрет по умолчанию, приходится предпринимать специальные меры всякий раз, когда необходимо разрешить доступ к какому-то ресурсу, однако эта модель с нашей точки зрения более надежна, чем противоположный вариант.

Политика разрешения по умолчанию позволяет добиться функционирования системы малыми усилиями, но при этом необходимо предусмотреть каждый конкретный случай, при котором требуется запретить доступ. Может случиться так, что необходимость внесения запретов станет ясна лишь тогда, когда в результате несанкционированного доступа сети будет нанесен значительный ущерб.

В обоих случаях для конфигурации брандмауэра используются цепочки правил. Каждая цепочка представляет собой набор правил, заданных явным образом, и политику по умолчанию. Пакет проверяется на соответствие каждому из правил, а правила выбираются из списка последовательно до тех пор, пока не будет обнаружено соответствие сетевого пакета одному из них. Если пакет не удовлетворяет ни одному из заданных правил, с сетевым пакетом производятся действия, определенные политикой по умолчанию.

В процессе работы брандмауэр может *пропустить* сетевой пакет, *запретить* прохождение сетевого пакета (deny) либо отказать сетевому пакету в прохождении, т. е. *отклонить* его (reject). С прохождением сетевого пакета все ясно, а чем же отличаются запрет и отклонение сетевого пакета? При отклонении сетевого пакета (reject) он удаляется, а его отправителю возвращается ICMP-сообщение об ошибке. При запрете прохождения сетевого пакета (deny) он удаляется, но отправитель не оповещается об его удалении.

В большинстве случаев запрет сетевого пакета считается лучшим решением, чем отказ в прохождении сетевого пакета. Во-первых, отправка сообщения об ошибке увеличивает сетевой трафик, а во-вторых, сообщения об ошибке могут быть использованы для организации атаки с целью вывода из строя сервера. Помимо этого, любое ответное действие на "неправильные" пакеты предоставляет взломщику дополнительную информацию о конфигурации вашей системы.

Фильтрация сетевых пакетов

Рассмотрим, на основании каких данных можно производить фильтрацию входящих и исходящих сетевых пакетов, а также каким образом определять "неправильные" сетевые пакеты.

Фильтрация входящих пакетов

Фальсификация исходящего адреса и недопустимые адреса

Рассмотрим признаки, по которым можно однозначно судить о поддельности сетевого пакета, поступающего из Интернета, или о проблемах прикладного программного обеспечения. На основании этих признаков нужно будет задать соответствующие правила фильтрации, чтобы ваш брандмауэр, обнаружив такой "неправильный" исходящий адрес в пакете, мог запретить прохождение сетевого пакета.

- Если в заголовке сетевого пакета в качестве исходного адреса указан адрес вашего компьютера. В процессе сетевого обмена невозможна ситуация, при которой сетевой пакет, отправленный с вашего компьютера, вернулся бы через внешний интерфейс. Следовательно, такой сетевой пакет — поддельный.
- Если в качестве исходящего IP-адреса указан адрес, попадающий в зарезервированный диапазон адресов, предназначенных для внутреннего применения. Согласно правилам распределения IP-адресов, в каждом из классов IP-адресов A, B и C существуют группы IP-адресов, выделенных для организации внутренних локальных сетей. В Интернете эти адреса не используются. При правильной конфигурации программного обеспечения через внешний порт не может прийти пакет с адресом источника, попадающий в один из перечисленных ниже диапазонов:
  - класс A — в диапазоне от 10.0.0.0 до 10.255.255.255;
  - класс B — в диапазоне от 172.16.0.0 до 172.31.255.255;
  - класс C — в диапазоне от 192.168.0.0 до 192.168.255.255.
- Если в качестве исходящего IP-адреса указан IP-адрес класса D, предназначенный для группового вещания. Адреса класса D, специально выделенные для организации группового вещания, находятся в диапазоне адресов от 224.0.0.0 до 239.255.255.255 и ни при каких обстоятельствах не могут выступать в качестве адреса источника.
- Если в качестве исходящего IP-адреса использован зарезервированный IP-адрес класса E. Класс E зарезервирован для будущего использования, ему принадлежат адреса в диапазоне от 240.0.0.0 до 247.255.255.255. Если брандмауэр встретит пакет с исходным адресом класса E, он должен предпринять меры, необходимые для того, чтобы такой пакет не попал в локальную сеть.

- Если в качестве исходящего IP-адреса использован адрес, принадлежащий интерфейсу обратной петли. Интерфейс обратной петли предназначен для локального использования сетевыми службами. Как правило, для обращения к интерфейсу обратной петли используется адрес 127.0.0.1, а вообще за интерфейсом локальной сети зарезервирована целая подсеть 127.x.x.x. Адрес интерфейса обратной петли не может присутствовать в заголовке пакета, полученного через внешний сетевой интерфейс.
- Если в качестве исходящего IP-адреса использован некорректный широковещательный адрес. Широковещательный адрес — это специальный тип адреса, определяющий передачу сетевого пакета на все компьютеры в сети. В качестве исходного адреса при широковещательной передаче может выступать обычный IP-адрес или адрес 0.0.0.0.

Фильтрация на основе адреса источника

При фильтрации пакетов единственный способ идентификации отправителя сетевого пакета — проверка IP-адреса источника в заголовке пакета. Одним из самых распространенных приемов при организации сетевых атак является фальсификация сетевых пакетов, при которой отправитель заменяет свой IP-адрес в заголовке сетевого пакета другим значением. Для подмены может быть выбран несуществующий или реальный IP-адрес, принадлежащий другому узлу.

Блокирование ненадежных узлов

Еще одна схема фильтрации, основанная на анализе IP-адресов источников, — это блокирование доступа с компьютеров, IP-адреса которых попадают в определенный диапазон. Как правило, таким образом отсекаются "подозрительные" компьютеры и целые сети, в частности обычно это происходит с сетями различных учебных заведений или разнообразных интернет-клубов, поскольку именно там молодежь любит "пошалить" в сети.

Работа с ограниченным набором удаленных узлов

В том случае, если вы организуете корпоративную сеть, не исключено, что вам потребуется таким образом настроить брандмауэр, чтобы некоторые типы пакетов принимались только в том случае, если они были отправлены с компьютеров с определенными адресами. Например, для организации системы передачи приватной информации.

Фильтрация на основе адреса назначения

В большинстве случаев фильтрация на основе адреса назначения выполняется автоматически. Сетевой интерфейс игнорирует пакеты, не адресованные непосредственно ему. Исключением являются широковещательные пакеты, адресованные всем узлам сети.

Фильтрация на основе порта источника

Номер порта источника, содержащийся в заголовке пакета, предназначен для идентификации программы-отправителя сетевого пакета, выполняющейся на удаленном узле. В запросах удаленных клиентов к вашему серверу содержатся различные номера портов, а в ответах сервера клиентам — один и тот же порт.

Фильтрация на основе порта назначения

Порт назначения определяет программу на вашем компьютере, которой предназначен пакет. В запросах удаленных клиентов, передаваемых на сервер, содержится один и тот же порт назначения, а в ответах сервера клиентам — различные номера портов.

Фильтрация на основе информации о состоянии TCP-соединения

В правилах обработки сетевых пакетов могут использоваться флаги, определяющие состояние TCP-соединения, поскольку любое сетевое соединение проходит через определенные состояния. Состояния клиента и сервера различаются между собой.

В первом пакете, отправленном удаленным клиентом, установлен флаг SYN, а флаг ACK сброшен. Передача такого пакета является началом в установлении TCP-соединения. Во всех последующих сетевых пакетах, предаваемых клиентом, установлен флаг ACK, а флаг SYN сброшен.

Пакеты, передаваемые удаленными серверами, всегда являются ответами на предыдущие обращения клиентов. В каждом пакете, поступившем от удаленного сервера, должен быть установлен флаг ACK, поскольку TCP-соединение никогда не устанавливается по инициативе сервера.

На основе анализа флагов можно отсеивать "неправильные" сетевые пакеты, которые могут являться признаком сетевой атаки.

Фильтрация исходящих пакетов

Фильтрация исходящих сетевых пакетов позволит исключить попадание в Интернет сетевых пакетов, передаваемых по локальной сети, а также избежать нежелательных обращений к серверам с узлов локальной сети. Источником таких обращений могут быть неверно сконфигурированные или вредоносные программы, запускаемые пользователями на их компьютерах.

Фильтрация на основе адреса источника

При этом типе фильтрации необходимо сформировать правила фильтрации таким образом, чтобы пакет, в котором указан адрес источника, не совпадающий ни с одним из адресов компьютеров вашей локальной сети, не был

пропущен брандмауэром. Это может вызвать некоторые затруднения, если в вашей организации разветвленная локальная сеть или IP-адреса выдаются динамически. Однако эти проблемы решаемы.

Фильтрация на основе адреса назначения

Как уже упоминалось выше, возможна ситуация, при которой вам потребуется ограничить передачу сетевых пакетов за пределы локальной сети адресами отдельных сетей или отдельных компьютеров. Эти адреса или диапазоны адресов могут быть указаны в правилах, задаваемых брандмауэру.

Фильтрация на основе порта источника

Проверка портов, указанных в заголовках сетевых пакетов, может выполняться как для клиентов, запущенных в локальной сети, так и для серверов. Такая проверка позволяет убедиться в том, что программы работают корректно и защищают Интернет от попадания в него внутреннего трафика локальной сети.

Пакеты, передаваемые сервером, обязательно должны содержать в заголовке порт источника, совпадающий с номером порта, выделенным для службы данного типа. Проверка номера порта представляет собой проверку конфигурации сетевых протоколов.

Фильтрация на основе порта назначения

Поскольку локальные клиенты могут обращаться к удаленным серверам лишь по конкретным номерам портов, фильтрация исходящих пакетов является одновременно средством контроля за использованием протоколов. Запрет прохождения сетевых пакетов на основе порта назначения не дает возможности пользователям локальной сети проводить сканирование портов удаленных компьютеров, ведь обычно сканирование портов — предвестник сетевой атаки.

Защита локальных служб

Как правило, локальные сервисы используются только внутри вашей сети, и предоставление доступа к этим службам извне нецелесообразно, а зачастую и вредно. Поэтому самый простой способ уберечься от проникновения в систему через один из сервисов — запретить доступ к сервису извне. Однако существуют службы, которые могут вызвать большое количество проблем при организации запрета доступа, например ICQ.

Один из способов защитить службы, предназначенные для внутреннего использования, — отказаться от размещения соответствующих серверов на компьютерах, доступных из глобальной сети. Однако в небольших сетях

обычно существует один-единственный сервер, зачастую выполняющий роль брандмауэра, поэтому в некоторых случаях компромиссы неизбежны.

Для защиты сервера от обращений из Интернета можно применить брандмауэр, выполняющий фильтрацию пакетов по порту назначения. Наличие такого брандмауэра позволяет запускать в локальной сети большое количество служб, не подвергая серьезной опасности сетевые ресурсы.

Программа ipchains

Как уже упоминалось выше, брандмауэр — это набор программных средств для организации защиты вашей сети. Большая часть функциональности брандмауэра интегрирована в ядре операционной системы Linux, но для создания и управления цепочками правил используются внешние программы. Для этих целей и предназначена программа ipchains. В последнее время активно внедряется программа iptables, однако она еще не получила такого широкого распространения, как программа ipchains.

Правила фильтрации пакетов, составляющих цепочки `input`, `output` и `forward` (входящие, исходящие и переадресация), содержатся во внутренних таблицах ядра операционной системы Linux. Каждое правило может быть включено в начало цепочки или добавлено в ее конец. Для определенности будем считать, что все правила, определяемые в данной главе, добавляются в конец цепочки. Порядок, в котором задаются правила, определяет порядок, в котором они будут включены в цепочку, и последовательность их применения к каждому пакету.

При поступлении на сетевой интерфейс извне информационного пакета содержимое заголовка этого пакета проверяется. Правила, принадлежащие цепочке input данного сетевого интерфейса, применяются последовательно одно за другим до тех пор, пока не будет найдено такое, которому удовлетворяет данный сетевой пакет. Соответственно, каждый сетевой пакет, отправляемый вовне, проверяется на соответствие правилам, содержащимся в цепочке output сетевого интерфейса. При обнаружении первого соответствия правилу проверка прекращается и к пакету применяется действие, указанное в составе правила `ACCEPT`, `REJECT` или `DENY`. Если пакет не удовлетворяет ни одному из правил, содержащихся в цепочке, вступает в действие политика по умолчанию. Таким образом, при работе брандмауэра пакет обрабатывается по первому из правил, которым он удовлетворяет.

Программе ipchains параметры передаются при вызове в командной строке. Формат командной строки приведен ниже:

```
ipchains   -A|I [цепочка] [-i интерфейс] [-p протокол] [  [!]  -y]
          [-s адрес [порт [: порт]]]
          [-d адрес [порт [: порт]]] — j действие [l]
```

В правилах, управляющих работой брандмауэра, предусматривается проверка адреса источника и адреса назначения. Для сравнения могут использоваться IP-адрес узла, диапазон IP-адресов, символьное имя узла и имя домена.

Программа ipchains позволяет задавать после IP-адреса дескриптор маски. *Дескриптор маски* — это целое число, которое может принимать значения от 0 до 32, и определяет число битов в маске. Дескриптор маски указывает, сколько старших битов адреса узла должны в точности совпадать с адресом, заданным в составе правила. Дескриптор маски, равный 32, означает, что адрес узла должен полностью совпадать с адресом, указанным в правиле. Если дескриптор маски отсутствует, считается, что он равен 32. Так, адрес 192.168.0.45 означает то же самое, что и выражение 192.168.0.45/32.

Опции ipchains

В табл. 29.1 приведены наиболее часто применяемые опции программы ipchains.

Таблица 29.1. *Опции программы ipchains*

Опция	Описание
`-A [цепочка]`	Добавляет правило к концу цепочки. Используются встроенные цепочки `input`, `output` и `forward`. Если цепочка не указана, правило добавляется ко всем цепочкам
`-I [цепочка]`	Включает правило в начало цепочки. Используются встроенные цепочки `input`, `output` и `forward`. Если цепочка не указана, правило включается во все цепочки
`-i интерфейс`	Определяет сетевой интерфейс, к которому должно применяться данное правило. Если сетевой интерфейс не указан, правило применяется ко всем сетевым интерфейсам
`-p протокол`	Определяет протокол семейства TCP/IP, к которому должно применяться правило. Если опция `-p` не указана, правило применяется ко всем протоколам. В качестве имен протоколов могут быть заданы `tcp`, `udp`, `icmp` и `all`. Разрешается использование имен и числовых значений, указанных в файле /etc/protocols
`-y`	В пакете, содержащем запрос на установление TCP-соединения, флаг `SYN` должен быть установлен, а флаг `ACK` — сброшен. Если данная опция не указана, проверка состояния флагов `SYN` и `ACK` не производится
`! -y`	В пакете, который передается в ответ на запрос на установление TCP-соединения, а также во всех последующих пакетах флаг `ACK` должен быть установлен. Если опция `! -y` не указана, состояние флага `ACK` не проверяется

Таблица 29.1 *(окончание)*

Опция	Описание
`-s адрес [порт]`	Определяет исходящий адрес пакета. Если исходящий адрес не указан, обрабатываются пакеты, переданные с любого узла. Если указан порт или диапазон портов, правило применяется только к пакетам, содержащим заданный номер порта. Если порт не указан, правило применяется ко всем пакетам, независимо от номера порта источника. При указании диапазона портов задаются начальный и конечный номера, разделенные двоеточием (например, 1024:65535). Если в опции `-s` задается порт, адрес также должен быть указан
`-d адрес [порт]`	Определяет адрес назначения пакета. Если адрес назначения не указан, обрабатываются все пакеты, передаваемые любому узлу. Если указан порт или диапазон портов, правило применяется только к пакетам, содержащим заданный номер порта. Если порт не указан, правило применяется ко всем пакетам, независимо от номера порта назначения. При указании диапазона портов задаются начальный и конечный номера, разделенные двоеточием (например, 1024:65535). Если в опции `-d` задается порт, адрес также должен быть указан
`-j действие`	Определяет действие, которое должно быть выполнено над пакетом (`ACCEPT`, `REJECT` или `DENY`). Для цепочки `forward` данный параметр также может принимать значение `MASQ` (masquerade — маскировка)
`-l`	Если пакет удовлетворяет правилу, в файл протоколов (по умолчанию /var/log/messages) должно быть записано информационное сообщение ядра

Символьные константы

Для улучшения удобочитаемости правил фильтрации и для удобства сопровождения в сценариях брандмауэра рекомендуется использовать символьные имена. В табл. 29.2 приведены некоторые символьные константы, используемые при конфигурировании брандмауэра.

Таблица 29.2. *Символьные константы, используемые при описании правил фильтрации*

Константа	Описание
`EXTERNAL_INTERFACE = "eth0"`	Сетевой интерфейс, подключенный к Интернету
`Internal_ INTERFACE = "eth1"`	Сетевой интерфейс, подключенный к локальной сети (для брандмауэра бастионного типа)

Таблица 29.2 *(окончание)*

Константа	Описание
`LAN_1="192.168.1.0/24"`	Диапазон адресов внутренней сети
`LAN_IPADDR_1="192.168.1.1"`	Адрес внутреннего интерфейса
`LOOPBACK_INTERFACE = "lo"`	Интерфейс локальной петли
`IPADDR = "ipaddress"`	Адрес вашего компьютера
`ANYWHERE ="any/0"`	Произвольный адрес
`MY_ISP = " ip range"`	Диапазон адресов провайдера
`LOOPBACK="127.0.0.0/8"`	Диапазон адресов обратной петли
`CLASS_A ="10.0.0.0/8"`	Адреса класса A для локальных сетей
`CLASS_B ="172.16.0.0/12"`	Адреса класса B для локальных сетей
`CLASS_C ="192.168.0.0/16"`	Адреса класса C для локальных сетей
`CLASS_D_MULTICAST ="224.0.0.0/4"`	Адреса класса D для группового вещания
`Class_E_Reserved_Net ="240.0.0.0/5"`	Адреса класса E. Зарезервировано
`BROADCAST_SRC  ="0.0.0.0"`	Исходящий широковещательный адрес
`BROADCAST_DEST ="255.255.255.255"`	Широковещательный адрес
`NAMESERVER = "mydns"`	Адрес DNS-сервера
`SMTP_GATEWAY="isp.server"`	Адрес почтового шлюза провайдера
`POP_SERVER="isp.server"`	Адрес POP-сервера провайдера
`NEWS_SERVER="isp.server"`	Адрес NEWS-сервера провайдера
`IMAP_SERVER="isp.server"`	Адрес IMAP-сервера провайдера
`PRIVPORTS="0:1023"`	Номера привилегированных портов
`UNPRIVPORTS="1024:65535"`	Номера непривилегированных портов
`SSH_PORTS="1000:1023"`	Номера привилегированных портов для протокола SSH — ограничиваем 24-мя одновременно возможными соединениями

Создание правил фильтрации

В этом разделе мы рассмотрим создание правил фильтрации для нашего брандмауэра. Эти правила ориентированы для среднестатистического брандмауэра, который полностью удовлетворяет потребностям одного ком-

пьютера или малой локальной сети. Для более серьезных случаев вы сможете оформить свой набор правил, используя нижеприведенные как базис.

Удаление существующих правил

Начиная создание своих правил фильтрации, обязательно удалите уже существующие правила. Это необходимо с точки зрения элементарной предосторожности — вдруг в системе уже определены некоторые правила фильтрации, идущие вразрез с вашей политикой безопасности.

Удаление правил фильтрации называется сбросом цепочки. Для сброса встроенных цепочек необязательно обращаться к ним явно. Все три цепочки — `input`, `output` и `forward` — можно сбросить с помощью одной команды, приведенной ниже:

```
ipchains -F
```

Определение политики по умолчанию

После удаления правил фильтрации автоматически устанавливается политика фильтрации сетевых пакетов по умолчанию, согласно которой разрешается прохождение всех сетевых пакетов. Таким образом, до тех пор, пока вы не внесете изменения в политику фильтрации сетевых пакетов, фильтрация сетевых пакетов производиться не будет.

Для обеспечения безопасности вашей операционной системы политика по умолчанию должна быть выбрана так, чтобы входящие сетевые пакеты удалялись без передачи сообщений на хосты, посылающие сетевые пакеты. Исходящим сетевым пакетам должно быть отказано в прохождении, а компьютеры, с которых эти сетевые пакеты были отправлены, должны получать ICMP-сообщения об ошибке. Обратившись к компьютеру вашей сети, программа, выполняющаяся на одном из компьютеров, размещенных в Интернете, не получит никакой информации о том, существует ли сервер, указанный в запросе. Такая же программа на локальном компьютере сразу получит сообщение о том, что операция, которую она собиралась выполнить, недопустима.

В следующих трех строках мы задаем, что для входящей цепочки сетевые пакеты будут уничтожаться, для исходящей цепочки и цепочки маршрутизации сетевые пакеты будут отклоняться, а компьютеры, посылающие сетевые пакеты, получат уведомление об ошибке.

```
ipchains -P input   DENY
ipchains -P output  REJECT
ipchains -P forward REJECT
```

После ввода в действие вышеприведенных правил весь сетевой трафик оказывается блокированным, в том числе и весь трафик, проходящий через интерфейс обратной петли.

Разрешение прохождения пакетов через интерфейс обратной петли

Поскольку в предыдущем разделе мы заблокировали весь сетевой трафик, автоматически возникнут проблемы с рядом программ, исполняемых на вашем компьютере, поскольку многие из них для нормального функционирования используют интерфейс обратной петли. Поэтому мы должны обеспечить прохождение всего сетевого трафика через интерфейс обратной петли. Так как этот интерфейс недоступен из-за пределов системы, подобные установки не могут повлечь за собой нежелательных последствий.

Правила, разрешающие прохождение сетевых пакетов без ограничений, очень просты. В данном случае надо нейтрализовать влияние политики по умолчанию на интерфейс обратной петли. Для этого введем следующие правила:

```
ipchains — A  input    -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains — A  output  -i $LOOPBACK_INTERFACE -j ACCEPT
```

Таким простым способом прохождение трафика через интерфейс обратной петли будет восстановлено.

Запрет прохождения пакетов с фальсифицированными адресами

Как уже упоминалось ранее, фальсификация адресов — один из признаков сетевых атак, поэтому следует бороться с сетевыми пакетами, имеющими сфальсифицированный адрес. Первое и очевидное правило — запретить прием сетевых пакетов, якобы отправленных с внешнего интерфейса вашего узла:

```
ipchains -A input  -i  $EXTERNAL_INTERFACE  -s  $IPADDR  -j  DENY  -1
```

Это правило отсекает входящие сетевые пакеты, содержащие в качестве адреса источника сетевой адрес вашего внешнего интерфейса. В том случае, если вы посылаете сетевой пакет на свой компьютер, он пройдет не через внешний сетевой интерфейс, а через интерфейс обратной петли. Если система настроена нормально, сетевой пакет, направленный на локальный компьютер, никогда не попадет на внешний интерфейс. В противном случае — либо у вас проблемы с сетевыми настройками, либо кто-то пытается пробраться к вам в систему, поскольку весь сетевой трафик, идущий через интерфейс обратной петли, проходит внутри системы, и любой сетевой пакет, содержащий такой адрес, является поддельным.

Следующие два правила запрещают пакеты, содержащие в качестве исходящего адреса интерфейс обратной петли:

```
ipchains -A input    -i $EXTERNAL_INTERFACE  -s  $LOOPBACK  -j DENY
ipchains -A output  -i  $EXTERNAL_INTERFACE  -s  $LOOPBACK  -j DENY  -l
```

Далее необходимо отсечь сетевые пакеты, исходящие адреса которых попадают в диапазон IP-адресов, выделенных для использования во внутренних сетях. Маршрутизаторы не должны обрабатывать пакеты с исходящими адресами, принадлежащими внутренним сетям.

Тем же самым ограничениям должны подвергаться и исходящие сетевые пакеты, у которых адреса назначения попадают в диапазон IP-адресов, выделенных для использования во внутренних сетях, поскольку в Интернете не могут существовать адреса, предназначенные для использования исключительно в локальных сетях.

Приведенные ниже наборы правил запрещают прохождение входящих и исходящих сетевых пакетов в случае, если адрес источника или адрес назначения принадлежит диапазонам сетевых адресов классов A, B и C, выделенных для использования в локальных сетях.

```
#  Запретить прохождение сетевых пакетов,
#  которые содержат адрес источника,
#  принадлежащий диапазону адресов класса A,
#  предназначенных для внутреннего использования.
ipchains -A input    -i  $EXTERNAL_INTERFACE  -s  $CLASS_A  -j DENY
ipchains -A input    -i  $EXTERNAL_INTERFACE  -d  $CLASS_A  -j DENY
ipchains -A output  -i  $EXTERNAL_INTERFACE  -s  $CLASS_A  -j DENY  -l
ipchains -A output  -i  $EXTERNAL_INTERFACE  -d  $CLASS_A  -j DENY  -l

#  Запретить прохождение сетевых пакетов,
#  которые содержат адрес источника,
#  принадлежащий диапазону адресов класса B,
#  предназначенных для внутреннего использования.
ipchains -A input    -i  $EXTERNAL_INTERFACE  -s  $CLASS_B  -j DENY
ipchains -A input    -i  $EXTERNAL_INTERFACE  -d  $CLASS_B  -j DENY
ipchains -A output  -i  $EXTERNAL_INTERFACE  -s  $CLASS_B  -j DENY  -l
ipchains -A output  -i  $EXTERNAL_INTERFACE  -d  $CLASS_B  -j DENY  -l

#  Запретить прохождение сетевых пакетов,
#  которые содержат адрес источника,
#  принадлежащий диапазону адресов класса C,
#  предназначенных для внутреннего использования.
ipchains -A input    -i  $EXTERNAL_INTERFACE  -s  $CLASS_C  -j DENY
ipchains -A input    -i  $EXTERNAL_INTERFACE  -d  $CLASS_C  -j DENY
ipchains -A output  -i  $EXTERNAL_INTERFACE  -s  $CLASS_C  -j DENY  -l
ipchains -A output  -i  $EXTERNAL_INTERFACE  -d  $CLASS_C  -j DENY  -l
```

Следующее правило, которое мы должны создать, необходимо для блокирования сетевых пакетов, содержащих недопустимые широковещательные адреса.

```
ipchains -A input  -i $EXTERNAL_INTERFACE -s  $BROADCAST_DEST  -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -d  $BROADCAST_SRC  -j DENY  -l
```

Первое из приведенных правил запрещает пакеты с исходящим адресом 255.255.255.255. Второе правило запрещает пакеты с адресом назначения 0.0.0.0. Подобные пакеты появляются не в результате ошибки, а являются признаком попытки атаки на вашу сеть.

Адреса группового вещания могут применяться лишь в качестве адреса назначения. Следующие правила выявляют фальсифицированные сетевые пакеты и фиксируют случаи их появления в файлах протоколов.

```
# Запретить пакеты, содержащие адреса класса D.
ipchains -A input -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST -j REJECT -l
```

Групповое вещание производится с использованием протокола UDP. В сетевых пакетах, передаваемых посредством группового вещания, адрес назначения отличается от пакетов, которые передаются в процессе обычного обмена между двумя узлами. Следующее правило запрещает передачу сетевых пакетов группового вещания с локального узла:

```
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_D_MULTICAST -j
REJECT  -l
```

Фильтрация ICMP-сообщений

Сообщения ICMP передаются при возникновении различных ситуаций, в том числе ошибочных. Они генерируются программами, анализирующими состояние сети, например ping или traceroute. В табл. 29.3 приведены ICMP-сообщения, которые могут представлять интерес для администратора сети.

Таблица 29.3. *Часто встречающиеся типы ICMP-сообщений*

Тип сообщения	Символьное имя	Описание
0	echo reply	Отклик программы ping
3	Destination unreachable	Сообщение об ошибке: один из маршрутизаторов по пути следования сетевого пакета не может доставить данные на следующий узел
4	source quench	Сообщение, предназначенное для управления потоком между двумя маршрутизаторами или между маршрутизатором и обычным узлом
5	Redirect	Если маршрутизатор знает о наличии более короткого пути, данное сообщение возвращается узлу, с которого был передан сетевой пакет

Таблица 29.3 *(окончание)*

Тип сообщения	Символьное имя	Описание
8	echo request	Запрос программы ping
11	time exceeded	Данное сообщение передается, когда количество узлов, через которые прошел сетевой пакет, превышает максимально допустимое
12	parameter problem	В заголовке сетевого пакета была обнаружена недопустимая запись

Сообщения об ошибках и управляющие сообщения

При настройке брандмауэра необходимо обеспечить прохождение четырех типов сообщений:

- Source Quench — подавление источника;
- Parameter Problem — некорректный параметр;
- Destination Unreachable (подтип Fragmentation Needed) — узел назначения недоступен (для входящих сообщений);
- Destination Unreachable (подтип Fragmentation Needed) — узел назначения недоступен (для исходящих сообщений).

Еще четыре типа ICMP-сообщений также могут быть разрешены для прохождения через брандмауэр. Это Echo Request (эхо-запрос), Echo Reply (эхо-ответ), различные подтипы исходящих сообщений Destination Unreachable, а также сообщение Time Exceeded (превышение времени). Остальные сообщения желательно игнорировать, чтобы они были удалены в соответствии с политикой по умолчанию.

Из всех типов сообщений, которые следует игнорировать, в таблице приведено только сообщение Redirect (перенаправление). Данное сообщение может быть использовано для организации атаки с целью вывода из строя сервисных средств. Остальные типы сообщений в основном предназначены для организации взаимодействия маршрутизаторов.

В последующих разделах описываются типы сообщений, поддержка которых необходима для обеспечения работы хостов локальной сети.

Управляющее сообщение Source Quench

ICMP-сообщение типа Source Quench (подавление источника) передается в тех случаях, когда маршрутизатор передает пакеты быстрее, чем принимающий узел может их обработать. Source Quench — одно из простейших средств контроля обмена данными на сетевом уровне. Обычно такими сообщениями обмениваются компьютеры, непосредственно связанные между собой.

Следующие правила разрешают прохождение входящих и исходящих ICMP-сообщений Source Quench:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 4 -d
$IPADDR -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 4 -d
$IPADDR -j ACCEPT
```

Если компьютер, которому предназначены пакеты, возвращает сообщение Source Quench, передающий узел должен снизить скорость обмена. Со временем он начинает повышать скорость передачи данных и делает это до тех пор, пока не получает следующее сообщение Source Quench.

Сообщение Parameter Problem

ICMP-сообщение типа Parameter Problem (некорректный параметр) возвращается в том случае, когда в заголовке сетевого пакета содержится недопустимая запись, либо если контрольная сумма заголовка сетевого пакета не соответствует контрольной сумме, указанной передающим хостом.

Следующие правила разрешают прохождение входящих и исходящих ICMP-сообщений Parameter Problem:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 12 -d
$IPADDR -j ACCEPT
pchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 12 -d
$IPADDR -j ACCEPT
```

Сообщение об ошибке Destination Unreachable

ICMP-сообщение типа Destination Unreachable (узел назначения недоступен) представляет собой сообщение об ошибке. В заголовке пакета данного типа содержится код, обозначающий ошибку, которая имела место.

Следующие правила разрешают прохождение входящих и исходящих ICMP-сообщений Destination Unreachable:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 3 -d
$IPADDR -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 3 -d
$IPADDR -j ACCEPT
```

С точки зрения безопасности — достаточно неоднозначный пакет, поскольку можно использовать такого типа сообщения для сбора информации об адресах узлов и портах. Кроме того, сообщения Destination Unreachable могут быть использованы для организации атаки с целью вывода узла из строя.

Тем не менее, подтип Fragmentation Needed сообщения Destination Unreachable необходим для нормальной работы сетевых средств. С его помощью взаимодействующие узлы договариваются об особенностях разбиения передаваемых сетевых пакетов на фрагменты.

Так же, если требуется, чтобы компьютеры вашей локальной сети отвечали на входящие запросы программы traceroute, необходимо разрешить передачу исходящих пакетов, содержащих подтип Port Unreachable сообщения Destination Unreachable.

Сообщение Time Exceeded

ICMP-сообщение типа Time Exceeded (превышение времени) передает сведения о том, что число узлов, через которые проходил сетевой пакет по пути его следования, превысило максимально допустимое значение. В настоящее время сообщение Time Exceeded обычно передается в ответ на UDP-запрос программы traceroute.

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 11 -d
$IPADDR -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $IPADDR 11 -d
$MY_ISP -j ACCEPT
```

Если требуется, чтобы ваша система отвечала на входящие запросы traceroute, необходимо разрешить передачу исходящих ICMP-сообщений Time Exceeded. Приведенные выше правила допускают обращения traceroute лишь с компьютера провайдера. Если вы хотите использовать traceroute на локальном узле, вы должны разрешить входящие сообщения Time Exceeded. Так как описываемая конфигурация брандмауэра не является маршрутизатором общего назначения, сообщения Time Exceeded используются только с описанной выше целью.

Программа ping: сообщения Echo Request и Echo Reply

Программа ping использует два типа ICMP-сообщений: Echo Request (эхо-запрос) и Echo Reply (эхо-ответ). Программа ping применяется для проверки связи с конкретными узлами сети.

Следующие два правила дают возможность передавать пакеты ping по любому адресу:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $IPADDR 11 -d
$MY_ISP -j ACCEPT
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 11-d
$IPADDR -j ACCEPT
```

Приведенные ниже правила позволяют принимать пакеты ping только с определенных узлов, а конкретно — из сети вашего провайдера:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $MY_ISP 8 -d $IPADDR
-j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $IPADDR 0 -d $MY_ISP
-j ACCEPT
```

В данном примере набор внешних узлов, которым разрешена передача вашей системе пакетов ping, ограничен компьютерами провайдера. Сделано

это для того, чтобы администратор провайдера мог в любой момент проверить, как происходит обмен данными с внешним интерфейсом вашего компьютера. Прием ping с остальных хостов запрещен.

Противодействие smurf-атакам

При организации атаки типа smurf пакеты ping, содержащие сообщения Echo Request, передаются в широковещательном режиме. Исходный IP-адрес в составе пакета подменяется IP-адресом "жертвы" — IP-адресом того узла, против которого направлена атака. В результате все узлы сети, получившие сообщения Echo Request, передают ответы по адресу "жертвы", загружая линии связи ICMP-пакетами. В результате, если у вас наружный канал не очень широкий — вы лишаетесь доступа в Интернет.

Приведенные ниже правила предназначены для протоколирования попыток smurf-атаки. Поскольку прохождение широковещательных ICMP-пакетов явно не разрешено ни одним из правил, эти пакеты будут удалены по умолчанию. Обратите внимание, что в правилах указаны не только Echo Request, но и другие типы сообщений. Дело в том, что возможности для атаки не ограничиваются сетевыми пактами ping.

```
# Противодействие smurf-атаке
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -d $BROADCAST_DEST -j
DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -d $BROADCAST_DEST -j
REJECT -l


# Маска сети
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -d $NETMASK -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -d $NETMASK -j REJECT -
l


# Адрес сети
ipchains -A input -i $EXTERNAL_INTERFACE  -p icmp -d $NETWORK -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -d $NETWORK -j REJECT -l
```

Разрешение функционирования служб

Выше мы определили правила, позволяющие отклонять сетевые пакеты с сомнительными адресами, а также разрешили локальному компьютеру работать через интерфейс обратной петли. В результате мы получили нормально функционирующий локальный компьютер с полностью отсутствующим доступом в Интернет. Наша дальнейшая задача — обеспечить нормальное функционирование локального компьютера (сети) в Интернете. Для того

чтобы ваш компьютер мог принимать и отправлять почту, работать по FTP, HTTP и т. п., необходимо разрешить прохождение сетевых пакетов с определенными портами. На первый взгляд — задача объемная, впрочем, необходимо обеспечить прохождение пакетов всего от десятка служб, что не так уж и много.

Служба DNS

Служба DNS использует в работе порт 53 и протоколы UDP и TCP. Соединение может устанавливаться как между клиентом и сервером, так и между двумя серверами. Для разрешения взаимодействия между клиентом и сервером необходимо добавить следующее правило:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p udp -s $IPADDR $UNPRIVPORTS -
d $NAMESERVER 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp -s $IPADDR $UNPRIVPORTS
-d $NAMESERVER 53 -j ACCEPT
```

В том случае, если ответ сервера не помещается в одной UDP-датаграмме, между клиентом и сервером устанавливается TCP-соединение. Обычно это происходит при передаче данных зоны между первичным и вторичным DNS-серверами. Для этого случая необходимо в цепочку правил добавить следующее правило:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNPRIVPORTS -
d $NAMESERVER 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNPRIVPORTS
-d $NAMESERVER 53 -j ACCEPT
```

В том случае, если у вас есть локальный DNS-сервер, и вы предоставляете его услуги каким-либо клиентам (например, компьютерам вашей локальной сети), желательно ограничить конкретным списком компьютеров доступ к вашему локальному DNS-серверу. Для этого воспользуйтесь следующими правилами:

```
# разрешение обмена между клиентом и DNS-сервером

ipchains -A input -i $EXTERNAL_INTERFACE -p udp -s <clients.addr>
$UNPRIVPORTS -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp -s <clients.addr>
$UNPRIVPORTS -d $IPADDR 53 -j ACCEPT


# разрешение обмена между DNS-серверами

ipchains -A input -i $EXTERNAL_INTERFACE -p udp -s <clients.addr> 53 -d
$IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp -s <clients.addr> 53 -d
$IPADDR -j ACCEPT
```

Следующие правила используются тогда, кода необходима передача с использованием протокола TCP:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s <dns.sec>
$UNIPRIVPORTS -d $IPADDR 53 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IPADDR 53 -d
<dns.sec> $UNIPRIVPORTS -j ACCEPT
```

E-mail

Для приема и пересылки электронной почты используются следующие протоколы:

- SMTP порт 25 TCP;
- POP3 порт 110 TCP;
- IMAP порт 143 TCP.

При создании правил, разрешающих функционирование SMTP-протокола, будем считать, что наша почта отправляется через провайдера.

Для передачи почты по SMTP-протоколу на почтовый сервер провайдера необходимо добавить следующие правила:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNIPRIVPORTS
-d $SMTP_GATEWAY 25 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $SMTP_GATEWAY 25
-d $IPADDR $UNIPRIVPORTS -j ACCEPT
```

В том случае, если у вас в локальной сети присутствует свой собственный SMTP-сервер, правила фильтрации несколько изменяются и принимают следующий вид:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNIPRIVPORTS
-d $ANYWHERE 25 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $ANYWHERE 25 -d
$IPADDR $UNIPRIVPORTS -j ACCEPT
```

Для получения электронной почты используется протокол POP3 или протокол IMAP. Для нормального функционирования POP3-протокола необходимо для нашего брандмауэра добавить следующие правила:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNIPRIVPORTS
-d $POP_SERVER 110 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $POP_SERVER 110
-d $IPADDR $UNIPRIVPORTS -j ACCEPT
```

Если вы хотите предоставить некоторым внешним хостам доступ к вашему POP3-серверу — используйте следующие правила:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s <pop.clients>
$UNIPRIVPORTS -d $IPADDR 110 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IPADDR 110 -d
<pop.clients> $UNIPRIVPORTS -j ACCEPT
```

В том случае, если у вас используется IMAP-протокол, добавьте следующие правила:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNIPRIVPORTS
-d $IMAP_SERVER 143 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IMAP_SERVER 143
-d $IPADDR $UNIPRIVPORTS -j ACCEPT
```

Если вы хотите предоставить некоторым внешним хостам доступ к вашему IMAP-серверу — используйте следующие правила:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s <pop.clients>
$UNIPRIVPORTS -d $IPADDR 143 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IPADDR 143 -d
<pop.clients> $UNIPRIVPORTS -j ACCEPT
```

NNTP

Сервер новостей использует порт 119 и протокол TCP. Для обеспечения нормального функционирования сервера новостей необходимо добавить три набора правил.

Если вы используете сервер новостей вашего провайдера, то для получения и отправки статей в группы новостей необходимо добавить следующие правила:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNIPRIVPORTS
-d $NEWS_SERVER 119 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $NEWS_SERVER 119
-d $IPADDR $UNIPRIVPORTS -j ACCEPT
```

В том случае, если у вас в локальной сети есть свой собственный сервер новостей, и вы хотите разрешить извне доступ определенным хостам — воспользуйтесь следующими правилами:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s <ip.clients>
$UNIPRIVPORTS -d $NEWS_SERVER 119 -j ACCEPT
```

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y -s $NEWS_SERVER 119
-d <ip.clients> $UNIPRIVPORTS -j ACCEPT
```

А поскольку вашему локальному серверу новостей необходимо получать и передавать статьи от сервера новостей провайдера — воспользуйтесь следующими правилами:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNIPRIVPORTS
-d $NEWS_SERVER 119 -j ACCEPT
```

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y -s $NEWS_SERVER 119
-d $IPADDR $UNIPRIVPORTS -j ACCEPT
```

Telnet

Telnet использует порт 23 протокол TCP. Применение протокола удаленного доступа Telnet было очень популярно еще три-четыре года назад, однако из-за того, что этот протокол абсолютно не защищен, а также вследствие появления альтернативы в виде протокола SSH — категорически не рекомендуется разрешать доступ извне по данному протоколу.

SSH

Протокол SSH использует порт 22 и протокол TCP. Защищенная замена Telnet и r-командам. При функционировании использует привилегированные порты 513—1023.

Чтобы применять протокол SSH для доступа из локальной сети к Интернету SSH-серверам необходимо ввести следующие правила:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNIPRIVPORTS
-d $ANYWHERE 22 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y -s $ANYWHERE 22 -d
$IPADDR $UNIPRIVPORTS -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $ANYWHERE $SSH_PORTS
-d $IPADDR 22 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IPADDR 22 -d
$ANYWHERE $SSH_PORTS -j ACCEPT
```

Следующие правила разрешают доступ удаленным клиентам к вашим локальным SSH-серверам:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $ANYWERE $UNIPRIVPORTS
-d $IPADDR 22 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IPADDR 22 -d
$ANYWHERE $UNIPRIVPORTS -j ACCEPT

ipchains -A -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $SSH_PORTS -d
$ANYWERE 22 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y -s $ANYWHERE 22 -d
$IPADDR $SSH_PORTS -j ACCEPT
```

FTP

Пожалуй один из сложных протоколов, поскольку использует несколько портов (TCP 21, 20). Протокол предусматривает два режима передачи — активный и пассивный канал передачи, что несколько осложняет конфигурацию брандмауэра.

Следующие правила разрешают доступ к удаленным FTP-серверам:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNPRIVPORTS
-d $ANYWERE 21 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y -s $ANYWHERE 21 -d
$IPADDR $UNPRIVPORTS -j ACCEPT
```

Следующие правила разрешают устанавливать соединения в режиме активного канала передачи данных:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y -s $ANYWHERE 20 -d
$IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNPRIVPORTS
-d $ANYWERE 20 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNPRIVPORTS
-d $ANYWERE $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y -s $ANYWHERE
$UNPRIVPORTS -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Для того чтобы к вашему локальному FTP-серверу был доступ извне — необходимо ввести следующие правила:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $ ANYWERE $UNPRIVPORTS
-d $IPADDR 21 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IPADDR 21 -d
$ANYWERE $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR 20 -d
$ANYWERE $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y -s $ANYWHERE
$UNPRIVPORTS -d $IPADDR 20 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $ANYWHERE $UNPRIVPORTS
-d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IPADDR
$UNPRIVPORTS -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

HTTP

Протокол HTTP использует порт 80 и протокол TCP. Для того чтобы локальные клиенты могли получить доступ к Web-серверам Интернета, необходимо ввести следующие правила:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -s $IPADDR $UNPRIVPORT -
d $ANYWHERE 80 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp !-y -s $ANYWHERE 80 -d
$IPADDR $UNIPRIVPORTS -j ACCEPT
```

В том случае, если у вас в локальной сети есть свой собственный Web-сервер, и вы хотите разрешить доступ извне — воспользуйтесь следующими правилами:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -s $ANYWHERE
$UNIPRIVPORTS -d $IPADDR 80 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y -s $IPADDR 80 -d
$ANYWHERE $UNIPRIVPORTS -j ACCEPT
```

Помимо перечисленных сервисов у вас могут действовать и другие, однако, зная протокол, используемые порты и опираясь на ранее приведенные пра-

вила, не составит труда добавить соответствующие правила для нормального функционирования ваших сервисов.

Запрет доступа с "неблагонадежных" узлов

Если вы обнаружите попытки сканирования портов или другие сомнительные действия, которые периодически предпринимаются с одного и того же хоста, желательно вовсе запретить обращение к системе с этого адреса.

Пример запрещающего правила приведен ниже:

```
ipchains -I input  -i $EXTERNAL_INTERFACE  -s <адрес/маска> -j DENY
```

Согласно этому правилу удаляется любой пакет, независимо от протокола и номера исходного порта или порта назначения.

Поддержка обмена в локальной сети

Для поддержки локальной сети, стоящей за брандмауэром, следует добавить некоторые правила. Эти правила необходимы для того, чтобы разрешить доступ к внутреннему сетевому интерфейсу брандмауэра и направить трафик в глобальную сеть. Как только на компьютере-брандмауэре будет реализована поддержка двух или более интерфейсов, он превратится в бастион.

Разрешение доступа к внутреннему сетевому интерфейсу брандмауэра

При работе с небольшими сетями вряд ли есть основание ограничивать доступ к брандмауэру из локальной сети. Следующие правила разрешают все виды взаимодействия между брандмауэром и локальной сетью:

```
ipchains -A input -i $INTERNAL_INTERFACE -s LAN -j ACCEPT
ipchains -A output -i $Internal_INTERFACE -s LAN_1 -j ACCEPT
```

Обратите внимание, что данные правила разрешают обмен лишь с брандмауэром. Доступ к Интернету отсутствует, поскольку по умолчанию компьютер, выполняющий роль брандмауэра, не производит динамической маршрутизации пакетов и не поддерживает статических маршрутов. Чтобы обеспечить маршрутизацию пакетов через брандмауэр, надо задать дополнительные правила.

Выбор конфигурации для пользующейся доверием локальной сети

Пакеты, передаваемые компьютерами локальной сети, можно условно разбить на две категории. Первая категория представляет собой данные, которыми

локальные узлы обмениваются с брандмауэром. Вторая категория — это данные, направляемые через внешний интерфейс брандмауэра в Интернет.

При работе небольшой сети вряд ли может возникнуть потребность в фильтрации пакетов, проходящих через внутренний интерфейс брандмауэра, однако некоторая обработка все-таки необходима. Речь идет о маскировке.

Если компьютер, выполняющий роль брандмауэра, имеет реальный IP-адрес, а всем остальным машинам, подключенным к локальной сети, присвоены адреса, предназначенные для внутреннего использования, то для обеспечения доступа локальных машин в Интернет брандмауэр должен взять на себя функции proxy-сервера.

По сути, компьютер, выполняющий маскировку пакетов, представляет собой низкоуровневый proxy-сервер. Такой сервер обслуживает клиентов, устанавливая соединения с удаленными узлами от своего имени. Поскольку исходный адрес в пакете, передаваемом в Интернет, заменяется исходным адресом компьютера, выполняющего маскировку, то с точки зрения удаленного узла обмен данными с ним проводит proxy-сервер. В пакетах, передаваемых удаленным узлом, адрес назначения заменяется на адрес локального компьютера.

Организация доступа из локальной сети к брандмауэру бастионного типа

Если вы занимаетесь администрированием небольшой сети, то, скорее всего, вы не захотите ограничивать доступ локальных компьютеров к брандмауэру бастионного типа. Следующее правило организует неограниченный доступ к внутреннему сетевому интерфейсу брандмауэра:

```
ipchains -A input -i $INTERNAL_INTERFACE -S LAN_1 -j ACCEPT ipchains -A
output -i $INTERNAL_INTERFACE -d LAN_1 -j ACCEPT
```

Перенаправление трафика

Если несколько локальных сетей должны обмениваться информацией, вам необходимо разрешить передачу пакетов между соответствующими интерфейсами. Конечно, делать это надо лишь в том случае, если маршрутизация не выполняется какими-либо другими средствами.

Чтобы брандмауэр можно было использовать в качестве маршрутизатора, объединяющего две локальные сети, необходимо добавить следующие правила:

```
# Приведенные правила разрешают доступ к брандмауэру
ipchains -A input -i $LAN_INTERFACE_1 -s LAN_1 -j ACCEPT ipchains -A out-
put -i $LAN_INTERFACE_1 -d LAN_1 -j ACCEPT
ipchains -A input -i $LAN_INTERFACE_2 -s LAN_2 -j ACCEPT ipchains -A out-
put -i $LAN_INTERFACE_2 -d LAN_2 -j ACCEPT
```

Следующие правила обеспечивают передачу трафика между локальными сетями в двух направлениях без выполнения маскировки:

```
ipchains -A forward -i $LAN_INTERFACE_2 -s LAN_1 -d LAN_2 -j ACCEPT
ipchains -A forward -i $LAN_INTERFACE_1 -s LAN_2 -d LAN_1 -j ACCEPT
```

Разрешение доступа к Интернету из локальной сети: IP-перенаправление и маскировка

На данном этапе на обмен данными между машинами локальной сети и внутренним интерфейсом брандмауэра не накладывается никаких ограничений. Однако локальные компьютеры не имеют доступа к Интернету. Для обеспечения такого доступа необходимо реализовать перенаправление и маскировку пакетов.

Механизм перенаправления реализуется на уровне ядра системы и позволяет компьютеру под управлением Linux выступать в роли маршрутизатора, перенаправляя трафик из одной сети в другую. Однако, даже если IP-перенаправление будет реализовано путем выбора конфигурации сети, пакеты не будут передаваться между интерфейсами до тех пор, пока не будут созданы правила, разрешающие такую передачу.

Перенаправления пакетов, адресованных различным узлам глобальной сети, не всегда достаточно для нормального взаимодействия локальных компьютеров с Интернетом. Если компьютерам локальной сети присвоены IP-адреса классов A, B и C, предназначенные для внутреннего использования, необходимо выполнить их маскировку, т. е. заменить исходящий адрес локального узла в пакете IP-адресом внешнего интерфейса брандмауэра. Эта возможность также реализована на уровне ядра операционной системы. Но даже если компьютеры локальной сети имеют обычные IP-адреса, допустимые для использования в Интернете, маскировка остается одним из самых эффективных средств защиты внутренней сети.

Несмотря на то, что перенаправление и маскировка — это совершенно различные механизмы, на уровне программы ipchains они представлены как одна процедура. Пакеты, поступившие на внутренний интерфейс брандмауэра, передаются на его внешний интерфейс. Перед тем как пакет будет помещен в очередь внешнего интерфейса, средства маскировки заменяют адрес источника IP-адресом внешнего интерфейса брандмауэра. Наличие средств перенаправления и маскировки превращает брандмауэр в proxy-сервер с возможностями фильтрации.

Приведенное ниже правило позволяет перенаправить трафик с внутреннего интерфейса на внешний, попутно выполняя маскировку пакетов:

```
ipchains -A forward -I $EXTERNAL_INTERFACE -s LAN_1 -j MASQ
```

Действия ACCEPT и DENY, указанные в цепочке output внешнего интерфейса, производятся после того, как перенаправление будет выполнено. Таким образом, несмотря на то, что передача от внутреннего к внешнему интерфейсу разрешена для всех пакетов, в Интернет попадут лишь те из них, для которых существуют разрешающие правила, связанные с внешним интерфейсом.

Правила маскировки позволяют задавать адреса источника и назначения, а также номера портов.

При перенаправлении трафик передается между сетевыми интерфейсами без изменений. Если компьютеры внутренней сети имеют IP-адреса, допустимые в Интернете, и брандмауэр выполняет перенаправление графика, то с точки зрения стороннего наблюдателя между локальной машиной и хостом Интернета устанавливается непосредственное соединение. При обращении удаленного компьютера к локальному узлу пакеты перенаправляются в локальную сеть.

При наличии маскировки передача трафика перестает быть симметричной. В этом случае разрешены лишь обращения из локальной сети к внешним серверам. При передаче пакета в Интернет исходный адрес, принадлежащий локальному компьютеру, заменяется IP-адресом внешнего интерфейса брандмауэра. При получении ответа от сервера производится обратное преобразование пакета — IP-адрес брандмауэра заменяется адресом локального компьютера, которому адресован пакет.

Как перенаправление, так и маскировка пакетов выполняются на уровне ядра операционной системы, поэтому ядро операционной системы должно быть скомпилировано с поддержкой маскировки и перенаправления пакетов.

Разрешить маскировку можно с помощью программы ipchains. Для того чтобы система выполняла маскировку всего трафика, направленного из локальной сети к удаленным узлам, необходимо задать следующее правило:

```
ipchains -A forward -i $EXTERNAL_INTERFACE \ -s LAN_1 -j MASQ
```

Независимо от того, выделены ли для компьютеров локальной сети допустимые IP-адреса или им присвоены адреса, предназначенные для внутреннего использования, при настройке брандмауэра рекомендуется отказаться от прямого перенаправления пакетов и использовать маскировку. Маскировка локальных сетей — мощное средство защиты. При использовании маскировки хосты Интернета не могут обращаться к компьютерам вашей локальной сети. Более того, локальные машины не видны извне. С точки зрения Интернета вся ваша локальная сеть состоит из одного хоста — компьютера, на котором реализован брандмауэр.

Дополнительной мерой защиты могут стать proxy-фильтры прикладного уровня, такие как SOCKS. И в этом случае при обмене с удаленным узлом создается впечатление, что запросы генерируются брандмауэром. Преиму-

щество фильтров прикладного уровня также состоит в том, что с их помощью можно организовать специальную обработку трафика, учитывающую специфику обмена с конкретными службами.

Организация демилитаризованной зоны

Конфигурация брандмауэра, описанная в начале главы, вполне подходит для защиты одного компьютера от нежелательных воздействий извне. Брандмауэр с двумя сетевыми интерфейсами может использоваться для защиты локальной сети. Брандмауэр бастионного типа защищает локальную сеть до тех пор, пока система, на которой установлен брандмауэр, не будет взломана. Даже если в процессе фильтрации участвует не только внешний, но и внутренний интерфейс, это не спасет систему. Если злоумышленнику удастся взломать компьютер, выполняющий роль брандмауэра, то ваша локальная сеть остается беззащитной перед лицом взломщика. Поэтому брандмауэр бастионного типа представляет собой единственную линию обороны. Такой тип защиты распространен в небольших организациях.

В средних и крупных организациях применение брандмауэра бастионного типа является недостаточной мерой. В таких сетях обычно применяются proxy-серверы либо система из двух брандмауэров, между которыми располагается демилитаризованная зона, или граничная сеть. Внешний интерфейс первого брандмауэра используется для соединения с Интернетом, а внутренний принадлежит демилитаризованной зоне, как правило использующей свою локальную сеть. Второй брандмауэр, который обычно называется *заглушкой*, также имеет два интерфейса. Внешний интерфейс подключен к демилитаризованной зоне, а внутренний — к внутренней сети предприятия. Обычно в демилитаризованной зоне размещаются серверы, которые должны быть доступны из Интернета. Для реализации описанной архитектуры требуется намного больше компьютеров и большая численность обслуживающего персонала, чем в случае применения брандмауэра бастионного типа.

Защита подсетей с помощью брандмауэров

Для организации демилитаризованной зоны обычно используется один из двух способов. Первый способ предполагает применение брандмауэра бастионного типа с тремя сетевыми интерфейсами. Один сетевой интерфейс подключается к Интернету, а два остальных — к двум изолированным локальным сетям. Одна из сетей выполняет роль демилитаризованной зоны, в ней размещаются общедоступные серверы. Во второй сети располагаются службы, предназначенные для внутреннего использования и компьютеры пользователей.

Другой способ состоит в использовании второго брандмауэра, называемого заглушкой (choke). Компьютер, на котором реализован брандмауэр-заглушка,

выполняет роль шлюза между демилитаризованной зоной и локальной сетью. Внутренний сетевой интерфейс брандмауэра-заглушки подключен к локальной сети, а внешний — к демилитаризованной зоне. Данные, передаваемые компьютерами локальной сети, маскируются. Таким образом, с точки зрения брандмауэра-бастиона и машин, принадлежащих демилитаризованной зоне, вся внутренняя сеть представлена одним брандмауэром-заглушкой.

Бастион маскирует трафик внутренней сети, поэтому, на первый взгляд, брандмауэр-заглушка не должен выполнять маскировку. Однако, если вся внутренняя сеть имеет один адрес, принадлежащий брандмауэру-заглушке, набор правил бастиона упрощается.

Подобная структура реализует две линии обороны локальной сети. Локальная сеть расположена за заглушкой и полностью изолирована от бастиона и, тем более, от Интернета. При использовании описанной системы необязательно задавать полный набор правил для внутреннего интерфейса бастиона, достаточно, если правила, осуществляющие фильтрацию пакетов, будут связаны с внешним интерфейсом заглушки.

Таким образом, система защиты внутренней сети содержит как минимум четыре набора правил — по одному для внутреннего и внешнего интерфейса каждого из брандмауэров. Правила для внешнего интерфейса бастиона практически совпадают с правилами брандмауэра, описанного ранее.

Реально описанная здесь система отличается от ранее рассмотренной наличием демилитаризованной зоны, а также новыми правилами, заданными для внутреннего интерфейса бастиона и для внешнего интерфейса заглушки. Указанные два набора правил, по сути, представляют собой зеркальное отражение друг друга.

Отладка брандмауэра

Брандмауэр установлен, настроен и активизирован, но функционирует не так, как хотелось. Даже если брандмауэр работает, рекомендуется сразу после установки и настройки произвести проверку правильности функционирования брандмауэра.

Общие рекомендации по отладке брандмауэра

Ниже приведены рекомендации, позволяющие облегчить отладку брандмауэра.

- Перед запуском сценария убедитесь, что в первой строке находится команда удаления существующих правил, а следующая команда устанавливает политику по умолчанию.

- Не рекомендуется производить изменения в правилах брандмауэра в X Window, поскольку неправильно заданное правило может "подвесить" X Window.
- Производите отладку с текстовой консоли. Не производите отладку брандмауэра с удаленной машины. В случае обрыва связи или неправильного конфигурирования вы рискуете остаться без доступа в Интернет.
- По возможности производите добавление правил по одному. В этом случае гораздо проще выявить причину неисправности. Сразу после добавления правил рекомендуется проверить их работоспособность.
- Обработка сетевого пакета определяется первым правилом, которому удовлетворяет этот сетевой пакет, поэтому порядок следования правил имеет большое значение.
- Помните, что существуют как минимум две не зависящие друг от друга цепочки: `input` и `output`. Если правила, содержащиеся в одной цепочке, обрабатывают пакет корректно, причина неисправности, очевидно, находится в другой цепочке.
- Если сценарий "зависает", возможно, что правило, в котором содержится доменное имя узла, вступает в действие раньше, чем правило, разрешающее доступ к DNS. Если какое-либо правило предшествует правилам, определяющим взаимодействие с DNS, в нем должны быть указаны IP-адреса. Использование доменных имен в таких правилах недопустимо, поскольку у вас еще нет доступа к серверу DNS.
- Проверяйте синтаксис команд программы ipchains. При составлении правил легко перепутать адрес или порт источника с адресом или портом назначения либо неверно задать регистр опции.
- При наличии синтаксической ошибки выполнение сценария брандмауэра завершается, и последующие правила не устанавливаются. Чтобы определить неверно составленное правило, запускайте сценарий с опциями `-x` или `-v`. Если указана опция `-v`, строки сценария выводятся в тот момент, когда они читаются интерпретатором команд. Опция `-x` задает вывод строк по мере выполнения команд оболочкой.
- Если какой-либо из серверов не работает, включите протоколирование удаляемых пакетов, указав опцию `-l` программы ipchains. Проанализируйте записи в файле /var/log/messages.
- Если вы обмениваетесь данными с Интернетом, работая на компьютере-брандмауэре, но не можете сделать этого с узла локальной сети, проверьте установки в /etc/sysconfig/network, связанные с перенаправлением пакетов.
- Если сервер доступен в пределах локальной сети, но попытка обратиться к нему извне оканчивается неудачей, включите протоколирование паке-

тов, проходящих через внутренний интерфейс. Постарайтесь выполнить всю проверку как можно быстрее, в противном случае в файле /var/log/messages появятся сотни записей.

- Если одна из служб не работает, временно включите в начало сценария брандмауэра правила, разрешающие прохождение пакетов в обоих направлениях, и задайте протоколирование, указав опцию `-l`. Проверьте, доступен ли сервер. Если это так, просмотрите записи в файле /var/log/messages и определите, какие порты используются при его работе.

Отображение списка правил брандмауэра

Чтобы убедиться, что правила брандмауэра инсталлированы именно так, как вы это планировали при составлении сценария, можно вывести содержимое цепочек. Сделать это позволяет опция `-L` программы ipchains. Если опция `-L` задана, ipchains выводит содержащиеся в соответствующей таблице ядра правила в той последовательности, в которой они применяются при обработке пакета. Для вывода содержимого цепочек используются следующие команды:

```
ipchains -L input
ipchains -L output
ipchains -L forward
```

Различные опции программы ipchains позволяют выводить содержимое одной и той же цепочки с различной степенью детализации. Форматы вывода для цепочек `input`, `output` и `forward` совпадают.

Утилиты

В состав пакета ipchains входит утилита ipchains-save, с помощью которой вы можете получить текущую конфигурацию брандмауэра на стандартный вывод, а затем перенаправить его в файл. Так же есть утилита — близнец ipchains-restore, которая может получать информацию из стандартного ввода.

Ссылки

- **bog.pp.ru/work/ipchains.html** — Bog BOS: ipchains: фильтрация пакетов в Linux: принципы работы, установка и настройка.
- Ipchains-HOWTO *(см. гл. 13)*.

Глава 30

Организация шлюза в Интернете для локальной сети

В этой главе мы займемся созданием точки доступа в Интернет для локальной сети. Обычно для этого используется выделенная линия, по концам которой установлены модемы, подключаемые к последовательному порту. В последнее время все чаще для организации подключения по выделенной линии используются технологии *x*DSL, при которых специальные модемы подключаются по интерфейсу Ethernet напрямую к сетевой карте, причем эти достаточно дорогие модемы сами уже и являются маршрутизаторами. Для определенности будем считать, что у нас есть модем, подключенный к последовательному порту.

Обычно в небольших локальных сетях выделяется один компьютер, который и выполняет роль маршрутизатора между локальной сетью и Интернетом, а также — счетчика трафика, брандмауэра, ограничителя скорости Web-сервера и т. п. Почти все, что необходимо для создания такой многопрофильной системы мы уже описывали выше, поэтому в этой главе остановимся только на тех проблемах, которые еще не рассматривались.

Начальные установки

Как правило во всех современных дистрибутивах Linux ядро собрано так, что работает как маршрутизатор пакетов между разными сетями и поддерживает механизм защиты маршрутизируемых пакетов и подсчет статистики.

Однако не будет лишним убедиться перед началом настройки системы, что в ядре вашей операционной системы присутствуют следующие необходимые для построения маршрутизатора элементы (функции):

- Networking support (поддержка сетевых свойств);
- TCP/IP networking (поддержка TCP/IP);
- IP forwarding/gatewaying (поддержка IP-маршрутизации);
- IP multicasting (поддержка специфических свойств IP-протокола);

- IP firewalling (поддержка брадмауэров);
- IP accounting (поддержка управления IP);
- Network device support (поддержка сетевых устройств).

Помимо этого, ядро операционной системы должно уметь работать с сетевыми картами, установленными на вашем компьютере, и поддерживать протокол PPP (Point-to-Point Protocol).

Само собой, следует правильно настроить сетевое оборудование, IP-адреса и т. п.

Связь с провайдером

Для подключения локальной сети к Интернету при помощи модема обычно используют два варианта. Первый из них предназначен для тех, кто платит за трафик, а второй используется теми, кто оплачивает проведенное в Интернете время.

В первом случае выход в Интернет осуществляется при помощи стандартного для Linux набора программ — pppd, chat и, возможно, еще нескольких скриптов. Происходит это следующим образом — вначале маршрутизатор дозванивается до провайдера и устанавливает с ним связь по протоколу PPP или по протоколу SLIP, который сейчас используется крайне редко. После установления соединения полученным каналом может пользоваться любой компьютер в вашей локальной сети (при соответствующей настройке). Канал удерживается до тех пор, пока не выключится ваш маршрутизатор или администратор явным образом не разорвет соединение.

Второй вариант — модификация первого, в англоязычной литературе он носит название dial on demand (звонок по требованию). Для его организации дополнительно используется программа diald, с помощью которой можно организовать работу таким образом, что если в течение заранее обусловленного времени не происходит обмена данными между локальной сетью и Интернетом, то diald разрывает соединение. При первой же попытке пользователя подключиться к Интернету diald снова дозванивается и устанавливает связь.

Поскольку второй вариант более сложный — будем рассматривать его как основной для организации нашего маршрутизатора.

Схема организации подключения локальной сети

Ниже приведены требования, которым должно удовлетворять подключение локальной сети к Интернету.

- Возможность доступа в Интернет — модем, телефонный номер и подключение к провайдеру.

- Набор программ для организации связи — pppd, chat и diald.
- Средство для управления брандмауэром — утилиты ipchains или iptables.
- Средство для ограничения трафика (если необходимо).
- Программное обеспечения для организации proxy-сервера.
- Программное обеспечение для учета и просмотра статистики.

Теперь, когда цели и средства известны, можно приступать к настройке программ.

Организация связи по коммутируемому соединению

Старейший вариант соединения с провайдером, и, к сожалению, наиболее распространенный в нашей стране. По сравнению с организацией связи по выделенному каналу представляет собой схему более сложную, поэтому рассмотрим ее первой.

Настройка программ

Будем считать, что на компьютере, который будет выходить в Интернет, правильно настроены сетевые параметры, и вы убедились в работоспособности локальной сети. Следующий шаг — добиться устойчивой связи с провайдером на вашем компьютере-маршрутизаторе.

Настройка связи с провайдером

Настроим подсистему дозвона и соединения с провайдером. Для удобства разобьем работу на два этапа:

1. Настройка PPP-соединения.
2. Установка и конфигурирование демона дозвона по требованию (diald).

Настройку модемного соединения мы здесь рассматривать не будем, поскольку это достаточно простая задача, и очень подробно рассмотрена в работе одного из отечественных патриархов Linux — В. Водолазкого "Установка PPP-соединения в Linux".

Почему мы используем протокол PPP? Основные преимущества протокола PPP по сравнению с протоколом SLIP состоят в следующем:

- назначение IP-адресов в PPP реализуется с помощью демона pppd, что значительно упрощает процесс конфигурирования при использовании динамических IP-адресов;

- коррекция ошибок, возникающих при передаче данных, осуществляется между компьютером провайдера и клиента, а не между удаленным компьютером, откуда берутся данные, и потребителем, как в протоколе SLIP.

Для организации связи между провайдером и клиентом необходимо получить данные, представленные в табл. 30.1.

Таблица 30.1. *Необходимые данные для настройки модемного соединения*

Необходимые данные	Значения в примере
Имя пользователя (login)	myname
Пароль пользователя (password)	vasya
IP-адрес пользователя (если есть)	192.168.0.100
IP-адрес сервера DNS	192.168.10.1

Процесс установления связи между вами и провайдером состоит из следующих этапов:

- соединения с компьютером провайдера с помощью модема;
- регистрации пользователя в удаленной системе;
- установки PPP-соединения.

Для решения этих задач в Linux используется небольшой набор скриптов, каждый из которых выполняет какую-то небольшую функцию. А поскольку это набор скриптов — никто не мешает на их базе определить именно те действия, которые необходимы вам при установлении или обрыве PPP-соединения.

Размещение скриптов зависит от настройки и предпочтений вашего дистрибутива. В современных версиях дистрибутива Red Hat используется два места — каталоги /etc/ppp и /etc/sysconfig/network-scripts. Наименования скриптов так же могут быть произвольными и очень часто зависят от предпочтений сборщика дистрибутива или системного администратора.

Для нашего случая будем считать, что у нас есть следующие файлы:

- /etc/ppp/chap-secrets — этот файл используется для аутентификации пользователя провайдером по протоколу chap. Обычно содержит имя и пароль пользователя для входа к провайдеру. В нашем случае это будет выглядеть следующим образом:

```
myname * vasya
```

- /etc/ppp/pap-secrets — этот файл используется для аутентификации пользователя провайдером по протоколу pap. Обычно содержит имя и пароль

пользователя для входа к провайдеру. В нашем случае это будет выглядеть следующим образом:

```
myname * vasya
```

- /etc/ppp/ip-up — данный скрипт используется для соединения с провайдером. Зачастую этот файл содержит только следующую строку:

  ```
  /usr/sbin/pppd
  ```

  Здесь можно настроить установление модемом соединения с провайдером или вызвать необходимый скрипт или программу;
- /etc/ppp/ip-down — этот файл используется для разрыва соединения с провайдером;
- /etc/ppp/options — это, пожалуй самый сложный и ответственный файл. Он определяет параметры нашего модема, скорость передачи по последовательному интерфейсу данных, настройки программы pppd и некоторые другие параметры. Обычно файл /etc/ppp/options оставляют неизменным, а для конфигурирования параметров соединения создают копию файла с именем /etc/ppp/options.ttySX, где ttySX — имя последовательного порта, к которому подключен наш модем. Пусть для определенности модем подключен к ttyS0 (COM1).

  ```
  # Устройство
  /dev/ttyS0
  # Скорость
  115200
  mru 1500
  # наш интерфейс : удаленный интерфейс
  192.168.0.100:192.168.0.101
  # маска подсети
  netmask 255.255.255.0
  bsdcomp 0
  chap-interval 15
  debug
  crtscts
  defaultroute
  ```

  Первые две строки определяют последовательный порт, к которому подключен наш модем, и скорость, на которой будет происходить обмен между модемом и последовательным портом. Далее — обратите внимание на строку со следующим содержимым:

  ```
  192.168.0.100:192.168.0.101
  ```

Эта строка определяет IP-адреса нашего последовательного интерфейса и провайдера. Такую строку необходимо добавить, если провайдер выдал нам постоянный IP-адрес. Как правило, в современном мире с коммутируемыми соединениями такого не происходит. Для статического IP-адреса также необходимо задать маску подсети.

Поскольку наш компьютер является маршрутизатором для локальной сети, необходимо настроить маршрутизацию. Для этого воспользуйтесь программой route и идущей с ней документацией. В том случае (а мы предположили, что точка подключения к провайдеру у нас одна) если у вас одно подключение к провайдеру, то можно в конец файла вписать команду `defaultroute`, что позволит вам добавить маршрут в системную таблицу маршрутизации, используя удаленную сторону как шлюз.

Команды pppd

Далее мы рассмотрим основные команды программы pppd (табл. 30.2).

Таблица 30.2. *Основные команды программы pppd*

Команда	Описание
`asyncmap 0`	Async-карта символов — 32-bit hex; каждый бит — символ, который надо представить в виде escape-последовательности, чтобы pppd мог его принять
`auth`	Требует от удаленной стороны назвать себя перед тем, как начнется обмен пакетами
`bsdcomp 0`	Определяет использование сжатия передаваемого трафика. На обычном модемном соединении не используется, позволяет в некоторых случаях почти в два раза увеличить количество передаваемых данных за единицу времени
`chap-interval` *интервал*	Определяет, что pppd будет заново вызывать удаленную сторону каждые *интервал* секунд
`chap-restart` *интервал*	Устанавливает интервал рестарта `chap` (пауза возобновления передач challenges) в *интервал* секунд
`chap-max-challenge` *значение*	Устанавливает максимальное число передач `chap challenge`
`connect` *<программа>*	Определяет программу для установки соединения
`Crtscts`	Предписывает использовать аппаратное управление потоком данных для управления потоком данных на последовательном порту

Таблица 30.2 *(продолжение)*

Команда	Описание
`Debug`	Предписывает увеличить уровень отладки. Если эта опция есть, pppd будет записывать в журнал все прибывшие и отправленные пакеты в понятной для человека форме. Пакеты регистрируются в log-файлах через syslog. Эта информация может быть перенаправлена в файл соответствующей установкой /etc/syslog.conf
`disconnect` *<программа>*	Предписывает запустить данную программу после того, как программа pppd завершила связь
`domain` *имя_домена*	Добавляет имя домена к имени машины
`ipcp-max-configure` *значение*	Устанавливает максимальное число передач IPCP configure-request
`ipcp-max-terminate` *значение*	Устанавливает максимальное число передач IPCP terminate-request
`ipcp-max-failure` *значение*	Устанавливает максимальное число IPCP configure-NAK, возвращенных перед началом отправки вместо configure-Rejects
`ipcp-restart` *интервал*	Устанавливает интервал перезапуска IPCP в *интервал* секунд
`local`	Предписывает не использовать линии управления модемом
`lock`	Предписывает, что pppd должна использовать `lock` в стиле UUCP для последовательного устройства
`login`	Предписывает использовать базу данных паролей для идентификации удаленной стороны
`modem`	Предписывает использовать линии управления модемом
`mru` *число*	Устанавливает значение MRU (Maximum Receive Unit, максимально принимаемый пакет) в *число*. При договоренности. pppd запросит удаленную сторону отправлять пакеты не более, чем по *число* байтов. Минимальное значение MRU — 128. Значение MRU по умолчанию — 1500. Для медленных соединений рекомендуется 296 (40 байтов для заголовка TCP/IP плюс 256 байтов данных)
`mtu` *число*	Устанавливает значение MTU (Maximum Transmit Unit-максимально передаваемый пакет) в *число*. Пока другая сторона не попросит меньшее значение при договоре о MRU, pppd будет требовать у сетевого кода ядра отправлять пакеты данных не более, чем по *число* байт через сетевой интерфейс PPP
`name` *имя_машины*	Устанавливает имя машины (для аутентификации)

Таблица 30.2 *(продолжение)*

Команда	Описание
`noauth`	Не требует удаленную сторону назвать себя перед тем, как начнется обмен пакетами
`noipdefalut`	Запрещает поведение по умолчанию, когда не указан локальный IP-адрес, которое определяет локальный IP-адрес по имени хоста. С этой опцией удаленная сторона должна обеспечить локальный IP-адрес в течение IPCP-переговоров (если она не определена явно в командной строке или в файле options)
`pap-restart` *интервал*	Устанавливает интервал возобновления передачи PAP в *интервал* секунд
`pap-max-authreq` *значение*	Устанавливает максимальное число передач PAP authenticate-request (запросов на аутентификацию по протоколу PAP)
`passive`	Разрешить опцию `passive` в LCP. С этой опцией pppd будет пытаться инициировать соединение, а если ответ от другой стороны не принят, то pppd будет пассивно ожидать правильный LCP-пакет от другой стороны вместо выхода
`silent`	С этой опцией pppd не будет передавать LCP-пакеты для инициации соединения, пока не придет правильный LCP-пакет от другой стороны
`user` *имя*	Устанавливает имя пользователя для аутентификации этой машины на другой стороне, используя PAP. Нельзя использовать вместе с `name`
`xonxoff`	Предписывает использовать программное управление потоком данных для управления потоком данных на последовательном порту
`+chap`	Двусторонняя chap-аутентификация
`+pap`	Двусторонняя pap-аутентификация
`-all`	Не разрешает договариваться о любых опциях LCP и IPCP
`-am`	Запрещает договариваться о asyncmap
`-chap`	Предписывает отказаться от chap-аутентификации
`-d`	Устанавливает уровень отладки. Если эта опция есть, pppd будет записывать в журнал все прибывшие и отправленные пакеты в понятной для человека форме. Пакеты регистрируются в log-файлах через syslog. Эта информация может быть перенаправлена в файл соответствующей установкой /etc/syslog.conf

Таблица 30.2 *(окончание)*

Команда	Описание
`-detach`	Предписывает не переходить в фоновый режим
`-ip`	Предписывает не договариваться об IP-адресе
`-mru`	Запрещает договариваться о mru
`-pap`	Предписывает отказаться от pap-аутентификации
`-pc`	Запрещает сжатие полей протокола

Как видите, параметров много и для полного понимания вопроса необходимо изучить соответствующую документацию.

Настройка diald

Обычно программа diald входит в стандартный дистрибутив, и установка ее с помощью менеджера пакетов rpm занимает совсем немного времени. После установки необходимо привести стандартную конфигурацию программы diald в соответствие с нашими реалиями.

Чтобы лучше понять то, что мы будем делать дальше, немного о принципе работы программы diald. Программа создает соединение на псевдотерминале и устанавливает маршрутизацию на получившийся интерфейс. После этого она начинает отслеживать пакеты, проходящие по виртуальному каналу. Если кто-то пытается выйти в Интернет, diald перехватывает данные, анализирует их и на основе правил, определяемых администратором, присваивает им определенные тайм-ауты. Далее пакеты отправляются по назначению, а тайм-ауты заносятся в так называемый набор соединения. Как только в наборе появляется первый тайм-аут, diald начинает дозваниваться до провайдера и пытается установить соединение. Организовав сеанс связи, демон переустанавливает маршрутизацию на реальный канал. Таким образом, связь с внешним миром оказывается установленной.

На протяжении всего времени соединения продолжает обновляться набор соединения. Истекшие тайм-ауты удаляются, новые поступают. И так продолжается, пока по какой-либо причине трафик не прекратится. Тайм-аутов в наборе становится все меньше и меньше, и когда последний из них оканчивается, diald разрывает связь.

Теперь перейдем непосредственно к конфигурированию. Этот процесс состоит из трех частей:

- создание скрипта соединения — файл /etc/diald/connect;
- настройка основной конфигурации — файл /etc/diald.conf;
- настройка правил тайм-аутов — файл /etc/diald/standard.filter.

Создание скрипта соединения: /etc/diald/connect

Как мы уже знаем, для организации сеанса связи необходимо выполнить несколько действий: дозвониться по телефону до поставщика услуг, пройти процедуру авторизации и запустить PPP-соединение. Поскольку у разных провайдеров этот процесс может коренным образом отличаться, то не имеет смысла встраивать эту процедуру в программу. Вместо этого используется внешний скрипт. Для этого достаточно подправить тот скрипт, который входит в стандартную поставку diald.

Ниже приведен вариант файла /etc/diald/connect.

```
#!/bin/sh

NIT="ATZ"    # Строка инициализации модема
PHONE="223322"    # Телефон провайдера
ACCOUNT="myname"    # логин
PASSWORD="vasya"    # пароль
# Определяем функцию для посылки
# сообщений в системный журнал
# и в FIFO-канал diald
function message ()
 {
   [ $FIFO ] && echo "message $*" >$FIFO
   logger -p local2.info -t connect "$*"
 }
# Начинаем процедуру связи
# Инициализируем модем
message "*** Initializing Modem ***"
chat "" $INIT OK ""
if [ $? != 0 ]
   then
      message "!!! Failed to initialize modem !!!"
      exit 1
fi
# Пытаемся дозвониться
message "*** Dialing system  ***"
chat \
      ABORT "NO CARRIER" \
      ABORT BUSY \
      ABORT "NO DIALTONE" \
      ABORT ERROR \
```

```
      "" ATDT$PHONE \
      CONNECT ""
case $? in
      0) message "*** Connected ***";;
      1) message "!!! Chat Error !!!"; exit 1;;
      2) message "!!! Chat Script Error !!!"; exit 1;;
      3) message "!!! Chat Timeout !!!"; exit 1;;
      4) message "!!! No Carrier !!!"; exit 1;;
      5) message "!!! Busy !!!"; exit 1;;
      6) message "!!! No DialTone !!!"; exit 1;;
      7) message "!!! Modem Error !!!"; exit 1;;
      *) esac
# Проходим авторизацию
message "*** Send login and password ***"
chat \
      login: $ACCOUNT \
      password: $PASSWORD        TIMEOUT 5 ""
if [ $? != 0 ] then
      message "!!! Failed to send !!!"
      exit 1
fi
# Все прошло удачно!
message "*** Protocol started *** "
```

Вышеприведенный скрипт — просто сценарий на языке командной оболочки, который вам необходимо немного адаптировать для ваших параметров.

Настройка основной конфигурации: /etc/diald.conf

/etc/diald.conf — основной конфигурационный файл программы diald, в котором задаются параметры устанавливаемого соединения и определяется поведение программы. Набор команд конфигурации у diald достаточно обширен, поэтому в приведенном примере будут использованы только необходимые, а подробную информацию по конфигурационным командам можно посмотреть в документации на программу diald.

Содержимое файла diald.conf:

```
# Протокол для связи с провайдером
mode ppp
# Вести журнал сеансов связи diald.log
accounting-log /var/log/diald.log
# Для управления демоном из внешних программ
```

```
# организовать канал FIFO — diald.ctl.
fifo /etc/diald/diald.ctl
# Для дозвона использовать файл /etc/diald/connect
connect /etc/diald/connect
# Далее несколько команд, описывающих применяемый модем.
# Поскольку мы уже определили параметры в /etc/ppp/options,
# то нижеприведенные команды необходимо закомментировать во избежание
# конфликтов в файле /etc/ppp/options
# device /dev/modem
# speed 115200
# modem
# lock
# crtscts
# Назначаем локальный и удаленный адреса нашего
# соединения. Если при связи с провайдером IP-адрес
# для вас выделяется динамически, то здесь можно
# поставить любые свободные адреса из диапазона,
# оговоренного при настройке нашей TCP/IP-сети.
# При запуске PPP diald сам выставит корректные значения
local 192.168.0.100
remote 192.168.0.101
# Провайдер дает нам динамический IP
dynamic
# Установить маршрут по умолчанию
# на виртуальное соединение
defaultroute
# Максимальное количество неудачных попыток дозвона
dial-fail-limit 10
# Задержка между попытками дозвона
redial-timeout 5
# время ожидания завершения скрипта connect
connect-timeout 120
# Файл с правилами для тайм-аутов
include /etc/diald/standard.filter
```

Настройка правил тайм-аутов: /etc/diald/standard.filter

Следующее, что вы должны сделать — произвести настройку правил тайм-аутов. Это самый сложный момент настройки diald, т. к. требует знания внутренней структуры IP-пакетов. Однако разработчики diald — люди доб-

рые и стандартный файл standard.filter имеет вполне приемлемые для большинства случаев настройки. Оставив в нем все, как есть, мы получим набор правил, рассчитанный на трехминутную паузу между окончанием активности в Интернете и разрывом связи с провайдером.

Комплексное тестирование

После проделанных манипуляций настало время проверить — правильно ли настроены наши программы. Для этого на компьютере желательно временно отключить все настройки брандмауэра (если вы, конечно, установили его). Затем необходимо запустить программу diald и попытаться выйти в "большой мир". Можно использовать браузер lynx (и зайти, например, на сайт **http://www.bhv.ru**), можно — программу ping.

Если все было настроено корректно, то после ввода предыдущей команды модем должен начать дозваниваться до провайдера. Через некоторое время связь будет установлена. Однако практически всегда lynx выдает сообщение о том, что не может соединиться с удаленным сервером! В данном случае — это нормальное явление. Дело в том, что при PPP-соединении с динамическими IP-адресами в силу определенных особенностей первый пакет обычно бывает утерян и не доходит до адресата. В результате мы ждем ответа от сервера, а он об этом и не подозревает. Достаточно повторить введенную ранее команду, чтобы все заработало.

Далее нам необходимо убедиться, что модем аккуратно разорвет соединение по прошествии трех минут. Дождавшись конца загрузки Web-страницы, засечем время. Примерно через три минуты diald должен дать команду на разрыв соединения.

Если у вас все прошло именно таким образом, значит система работает как надо. В противном случае проанализируйте последние строки системного журнала (/var/log/messages).

Указанными действиями мы проверили корректную работу только с нашего компьютера-маршрутизатора. Однако нам надо сделать то же самое и с любого компьютера в локальной сети, поэтому попробуем повторить описанную процедуру и на любом компьютере. Реакция diald должна быть аналогичной. Если что-то пошло не так, проверьте корректность настройки протокола TCP/IP на этой машине, в частности — настройки сетевого шлюза, которые должны указывать на наш компьютер-маршрутизатор.

Организация связи по выделенному каналу

В отличие от настройки связи по коммутируемому соединению, организация соединения по выделенному каналу намного более простая задача.

Настройка связи с провайдером

Как и в предыдущем случае, нам необходимо правильно настроить программу pppd. Поскольку параметры программы pppd мы уже рассматривали, просто приведем файл options и прокомментируем его содержание.

```
# Устройство
/dev/ttyS0
# Скорость
115200
mru 1500
noauth
# наш интерфейс : удаленный интерфейс
192.168.0.100:192.168.0.101
# маска подсети
netmask 255.255.255.0
bsdcomp 0
chap-interval 15
debug
crtscts
-detach
defaultroute
```

Первые две строки определяют последовательный порт, к которому подключен наш модем, и скорость, на которой будет происходить обмен между модемом и последовательным портом. Далее — обратите внимание на строку со следующим содержимым:

```
192.168.0.100:192.168.0.101
```

Эта строка определяет IP-адреса нашего последовательного интерфейса и провайдера. Такую строку необходимо добавить, если провайдер выдал нам постоянный IP-адрес. Для статического IP-адреса также необходимо задать маску подсети.

В том случае, если у вас одно подключение к провайдеру, то можно в конец файла вписать команду `defaultroute`, что позволит вам добавить маршрут в системную таблицу маршрутизации, используя удаленную сторону как шлюз.

Вот и все, что требовалось для конфигурации программы pppd для соединения по выделенному каналу. Правда, намного проще, чем с коммутируемым?

Осталось только отредактировать файл inittab, чтобы pppd автоматически стартовала. Для этого необходимо добавить следующую строчку:

```
7 : 2345 : respawn: /usr/sbin/pppd file /etc/ppp/options.ttyS0 >
/var/log/pppS0.log
```

Комплексное тестирование

Теперь настало время проверить — правильно ли настроено наше соединение по выделенному каналу. Для этого перезагрузите компьютер-шлюз для вступления в силу внесенных в файл inittab изменений и временно отключите все настройки брандмауэра (если вы, конечно, установили его). Затем необходимо попытаться выйти в Интернет. Быстрее всего — использовать программу ping:

```
ping http://www.bhv.ru
```

Если все было настроено корректно, то вы увидите отклик от сайта **www.bhv.ru**.

Если у вас все прошло именно таким образом, значит, система работает как надо. В противном случае проанализируйте последние строки системного журнала (/var/log/messages).

Этим действием мы проверили корректную работу только с нашего компьютера-маршрутизатора. Однако нам надо сделать то же самое и с любого компьютера в локальной сети. Если что-то пошло не так, проверьте корректность настройки протокола TCP/IP на этой машине, в частности — настройки сетевого шлюза, которые должны указывать на наш компьютер-маршрутизатор.

Итак, вы получили вполне работоспособный шлюз в Интернет для вашей локальной сети. Однако это далеко не все. Система наша открыта для любого постороннего вмешательства, а шлюз должен обеспечить беззащитную локальную сеть защитой извне и изнутри, вести учет потребленного трафика (и причем зачастую — покомпьютерно), ограничить нас от информации нежелательной или сомнительной (например баннеров), обработать статистику и красиво ее подать — лучше всего графически. Как видите — задач много, и мы будем их решать постепенно.

Защита локальной сети

Защита локальной сети — понятие комплексное и многогранное. В данном случае мы имеем в виду правильную настройку брандмауэра на нашем компьютере-шлюзе. Процедура настройки брандмауэра была описана *в гл. 29*, и к этому вопросу добавить больше нечего.

Установка proxy-сервера

Следующее, что мы должны решить для нашей локальной сети — каким образом минимизировать расходы на потребляемый Интернетом трафик и как увеличить скорость получения информации. Для решения этой проблемы используется стандартный рецепт — proxy-сервер. Что собой представля-

ет proxy-сервер? Если с помощью браузера, настроенного для работы через proxy-сервер, вы запросите из Интернета какой-либо документ, и при этом окажется, что некоторое время назад кто-то уже обращался с подобным запросом, вы получите документ незамедлительно, с максимальной скоростью, на которую способно ваше сетевое подключение, потому что направлена вам будет копия документа, взятая из кэша proxy-сервера. Если же в кэше proxy-сервера данный документ отсутствует, то proxy-сервер запросит удаленный Web-сервер, хранящий оригинал, выдаст документ вам, и одновременно положит копию документа в свой кэш на случай такого же запроса. Чем больше пользователей пользуются proxy-сервером, тем более существенной становится его помощь.

Наиболее часто используемой программой proxy является программа Squid — высокопроизводительный кэширующий proxy-сервер, поддерживающий протоколы FTP, Gopher, и HTTP. Squid сохраняет часто запрашиваемые данные в оперативной памяти компьютера, что позволяет резко увеличить производительность proxy-сервера, кэширует DNS-запросы (это свойство интересно тем, кто не имеет своего DNS-сервера). Помимо вышеперечисленных возможностей, поддерживает SSL, расширенный контроль доступа и полную регистрацию запросов.

Программа Squid описана *в гл. 22*, однако мы позволим себе напомнить некоторые интересные моменты по ее использованию.

Transparent proxy

Transparent proxy — таким образом настроенный proxy-сервер, что его использование прозрачно для пользователей. Это имеет как хорошую, так и плохую стороны. С одной стороны, пользователям не придется настраивать соединение через proxy-сервер в своей системе, а трафик гарантированно проходит через proxy-сервер. С другой стороны, теряется свобода выбора пользователя — пользоваться или нет proxy-сервером. Кроме того, некоторые сайты некорректно обрабатываются proxy.

Для организации transparent proxy необходимо таким образом настроить маршрутизатор (брандмауэр), чтобы транзитные пакеты, предназначенные для 80 порта, попадали на вход proxy-сервера. Соответствующие настройки transparent proxy приведены *в гл. 22*.

Борьба с баннерами

Наверняка вам встречались Web-страницы, на которых рекламных баннеров было больше, чем нужной информации. В этом случае можно настроить локальный сервер Squid таким образом, чтобы не происходила закачка баннеров. Борьбу с баннерами можно производить разными методами:

- настроить отдельный proxy-сервер с ограничением баннеров: хочешь — используй, не хочешь — не используй;

- совместить ограничение баннеров с transparent proxy;
- организовать proxy на локальной системе для ограничения баннеров.

Соответствующие настройки программы Squid для борьбы с баннерами приведены *в гл. 22*.

Разделение внешнего канала (ограничение трафика)

Часто бывает так, что у вас есть внешний канал — скажем, 128 Кбит, и несколько групп пользователей с определенным приоритетом.

И надо, чтобы одна группа имела фиксированную ширину наружного канала (скажем, 64 Кбит), а две другие — ширину наружного канала по 32 Кбит. Для решения этой непростой задачи мы также можем воспользоваться Squid. Соответствующие настройки программы Squid для разделения внешнего канала приведены *в гл. 22*.

Помимо Squid, для этого можно воспользоваться специализированными программами, называемыми traffic shaper. Существует несколько программ такого типа с различной функциональностью. В частности, есть traffic shaper, которая позволяет ограничить канал не по пропускной способности, а по полученным мегабайтам. Принцип действия ее оригинален и прост. Допустим, у вас выделенный канал, причем в арендную плату входит один гигабайт входящего трафика. В программе traffic shaper выставляется ограничение один гигабайт в месяц. Далее происходит следующее. В начале информация качается с той скоростью, с какой способен передавать информацию выделенный канал, но при приближении к заветной цифре — пропускная способность канала, ограниченного traffic shaper, все уменьшается и уменьшается, не позволяя вам выйти за рамки ограничения в один гигабайт в месяц. В результате, в последние дни месяца скорость канала может упасть до десятков байтов в секунду.

В качестве стабильной и хорошо конфигурируемой программы типа traffic shaper можно порекомендовать пакет CBQ. Ограничивать трафик можно и с помощью утилиты tc, входящей в пакет iproute2.

Мониторинг загрузки каналов

Для анализа загрузки интернет-канала необходимо использовать дополнительный пакет, поскольку разбираться самим в log-файлах системы — задача неблагодарная. Чтобы обеспечить требуемую наглядность, такой пакет должен выдавать информацию в графической форме, причем, желательно, с помощью Web-интерфейса. Все эти условия реализованы в программах MRTG (Multi Router Traffic Grapher) и RRDtool (Round Robin Database).

Программа MRTG

MRTG создает HTML-страницу с отображением загрузки канала за сутки, неделю, месяц и год. Для этого используется написанный на Perl скрипт, который опрашивает маршрутизатор через SNMP, а программа, написанная на C, обрабатывает получившийся результат и создает встроенные в HTML-страницу изображения в формате GIF/PNG. Помимо собранной самостоятельно информации пакет MRTG может обрабатывать информацию и из других источников (cpuinfo, df, squid и т. п.) и строить графики по полученной информации.

Большим преимуществом данной программы является постоянный размер журналов, в которых более старая информация хранится с меньшими подробностями.

Внешний вид получаемых графиков приведен на рис. 30.1.

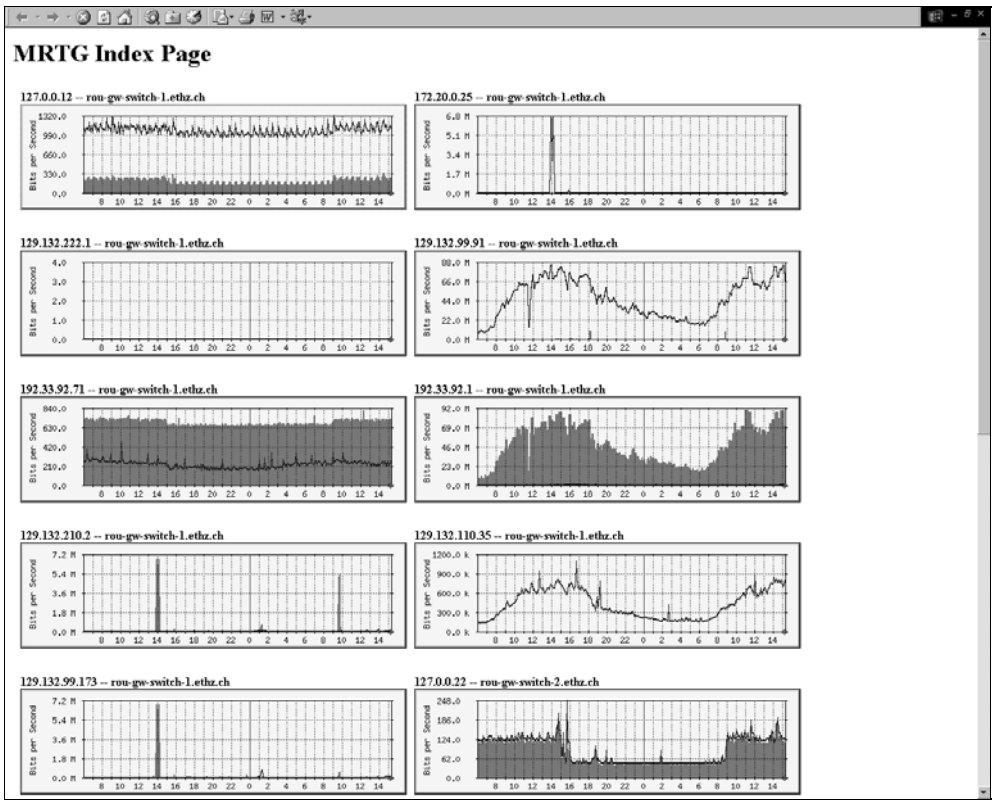

Рис. 30.1. Результат работы программы MRTG

Конфигурирование MRTG

Для конфигурирования программы MRTG используется файл mrtg.cfg, параметры которого и будут рассматриваться в данном разделе. Как обычно, будут приведены только ключевые параметры, с полным списком можно ознакомиться в документации, прилагаемой к этому программному пакету.

Правила записи параметров в конфигурационном файле:

- ключевое слово — в начале строки;
- двоеточие — разделитель, идущий сразу за ключевым словом;
- строка продолжения начинается с пробела;
- строки комментария начинаются с символа `#`.

Итак, файл mrtg.cfg может содержать следующие команды:

- `Include: имя_файла` — подключаемый файл;
- `WorkDir: имя_каталога` — задает размещение журнала, рабочих файлов и генерируемых страниц, имеет приоритет над `HtmlDir`, `ImageDir` и `LogDir`;
- `HtmlDir: имя_каталога` — задает размещение генерируемых страниц;
- `ImageDir: имя_каталога` — задает размещение генерируемых изображений; обязательно находится под `HtmlDir` — страницы генерируются в этом предположении;
- `LogDir: имя_каталога` — задает размещение журнала;
- `Refresh:` — частота перерисовки графиков в браузере;
- `RunAsDaemon: no | yes` — запуск MRTG в режиме демона;
- `Interval:` — предполагаемый интервал запуска MRTG;
- `IconDir:` — каталог, где хранятся значки;
- `Forks: число` — определяет, сколько запускать параллельных процессов опроса;
- `WriteExpire: no | yes` — создавать файлы .meta для apache;
- `NoMib2: no | yes` — не запрашивать `sysUptime`, `sysName`;
- `Language: язык_отчетов` — определяет язык отчетов, есть поддержка русского языка;
- `LogFormat: rrdtool` — формат журналов для rrdtool — динамическое создание отчетов;
- `LibAdd: адрес-библиотеки-rrdtool RRDs.pm` — адрес библиотеки rrdtool;
- `PathAdd: адрес-rrdtool` — адрес rrdtool;

Для каждого контролируемого устройства — обозначается как target, буквы преобразуются к строчным, создается отдельная секция. При работе MRTG каждый target порождает файлы журнала (target.log и target.old), картинки с графиками (target-day.gif, target-week.gif, target-month.gif, target-year.gif) и HTML-страницу (target.html).

- `Target[target]: порт:community@маршрутизатор [:port[:timeout[:retries[:backoff[:2]]]]]`

  где:

  - `порт` — номер интерфейса на маршрутизаторе;
  - `community` — пароль на чтение;
  - `маршрутизатор` — имя или IP-адрес;
  - `port` — по умолчанию стандартный порт SNMP;
  - `timeout` — время ожидания;
  - `retries` — количество попыток;
  - `backoff` — во сколько раз увеличивать timeout при каждом повторе;
  - `2` — означает использование 64-битных счетчиков;

- `Target[target]: внешняя-программа-с-параметрами-в-обратных-кавычках`

  Программа должна возвращать на стандартный вывод 4 строки:

  - значение счетчика входных байтов;
  - значение счетчика выходных байтов;
  - текстовую строку, содержащую информацию о времени работы объекта после включения;
  - строку, указывающую имя объекта;

- `RouterUptime[target]: community@маршрутизатор` — откуда брать информацию об имени маршрутизатора и его времени работы для составных target;
- `MaxBytes[target]:число` — значения переменных, которые больше этого числа, игнорируются;
- `Title[target]:` — заголовок для HTML-страницы;
- `PageTop[target]:` — текст, выдаваемый в верхней части HTML-страницы;
- `PageFoot[target]:` — текст, выдаваемый в нижней части HTML-страницы;
- `AddHead[target]:` — HTML-текст, вставляемый после `TITLE` внутри `HEAD`;
- `MaxAbs[target]: число` — если используется сжатие, то возвращаемое значение может превосходить `MaxByte`;

- `Unscaled[target]: [d][w][m][y]` — подавить масштабирование по вертикали для соответствующего графика (`d` — день, `w` — неделя, `m` — месяц, `y` — год);
- `WithPeak[target]: [w][m][y]` — показывать в недельном, месячном и годовом графиках не только средние, но и пиковые значения (`w` — неделя, `m` — месяц, `y` — год);
- `Supress[target]: [d][w][m][y]` — подавить генерацию части графиков (`d` — день, `w` — неделя, `m` — месяц, `y` — год);
- `Directory[target]: имя-каталога` — размещать в данном каталоге все файлы, относящиеся к указанному `target`;
- `XSize[target]: число` — число пикселов в графике по горизонтали;
- `YSize[target]: число` — число пикселов в графике по вертикали;
- `YTics[target]: число-вертикальных-делений`;
- `Step[target]: секунд` — определяет шаг отображения в секундах;
- `Options[target]: список-опций-через-запятую`:
  - `growright` — время движется вправо;
  - `bits` — все числа умножать на 8 (измерять в битах);
  - `perminute` — все числа умножать на 60 (измерять в единицах за минуту);
  - `perhour` — все числа умножаются на 3600 (измерять в единицах за час);
  - `transparent` — генерировать прозрачный фон картинки;
  - `gauge` — интерпретировать полученные значения как абсолютные значения. Полезно для отображения таких параметров, как загрузка процессора, дискового пространства;
  - `unknaszero` — трактовать неверные значения как 0, а не как повторение предыдущего значения;
- `kilo[target]: число` — что понимается под `kilo`. По умолчанию — 1000, можно установить 1024.
- `kMG[target]: список-префиксов-множителей` — какими буквами обозначать kilo, mega и т. п. По умолчанию: "K, M, G, T, P".
- `Colours[target]: Colouri#RRGGBB,Colouri#RRGGBB,Colouri#RRGGBB,Colouri#RRGGBB` — определение цветовой схемы, где `Colour` — текстовое имя цвета, помещаемое в легенду графика, `i` = 1, 2, 3, 4 — номера цвета, `RRGGBB` — шестнадцатеричные значения, определяющие RGB-цвет;

- `Background[target]: #RRGGBB` — задает цвет фона;
- `YLegend[target]: текстовая-строка` — по умолчанию: "Bits per second";
- `ShortLegend[target]: текстовая-строка` — по умолчанию: "b/s".

Помимо MRTG, существует еще один пакет аналогичного назначения — RRDtool.

Программа RRDtool (Round Robin Database)

Этот программный пакет обеспечивает хранение и отображение данных мониторинга — загрузку каналов, температуру и любую другую зависящую от времени последовательность данных. Задумывалась как повторная, но более правильная реализация MRTG. Объем хранимых данных не увеличивается со временем — ячейки хранения используются циклически. В отличие от MRTG, программа не упаковывает старые данные самостоятельно, сбор информации и генерация HTML-кода также производятся с помощью внешних средств. Параметры передаются в командной строке или через утилиту stdin.

Подсчет трафика

Иногда необходимо подсчитать трафик по клиентам, особенно когда организуется подключение домовой локальной сети или несколько небольших фирм совместно покупают выделенную линию для подключения к провайдеру. К сожалению, стопроцентного совпадения подсчитанного трафика с данными провайдера добиться вряд ли удастся, поскольку приведенные ниже способы подсчета трафика дают *разные* результаты. Правда, погрешность подсчета обычно не превышает 5%.

Есть несколько вариантов подсчета трафика:

- по данным, взятым из SNMP (OutOctets на интерфейсе);
- по данным, взятым из Cisco;
- по данным, взятым из /proc/tty/driver/serial;
- по данным, взятым из radacct (radius-accounting/ OutOctets);
- по ipchains;
- с помощью nacctd.

Ниже приведен достаточно простой способ подсчета трафика с использованием ipchains.

Смысл метода такой — ставим разрешительную цепочку для обсчитываемого IP-адреса, например:

```
ipchains -A output -d AA.BB.CC.DD -j ACCEPT
```

Теперь можно посчитать байты:

```
ipchains -L -v
Chain input (policy ACCEPT: 4195746 packets, 1765818402 bytes):
Chain forward (policy ACCEPT: 142999 packets, 29941516 bytes):
Chain output (policy ACCEPT: 4182597 packets, 1309541595 bytes):
pkts bytes target prot opt tosa tosx  ifname mark outsize source destination  ports
   4  308  ACCEPT all  --  0xFF 0x00  any              anywhere AA.BB.CC.DD  n/a
```

Из примера видно, что клиенту ушло 308 байтов. Со временем в столбике `bytes` будет накапливаться статистика по байтам. Далее необходимо как-то обрабатывать эти данные и выводить себе и клиенту. Для этого можно воспользоваться программой на Perl, расположенной по адресу **linux.uatel.net.ua/ipcount.perl**.

Существует также пакет, предназначенный для подсчета IP-трафика через протокол SNMP. Он так и называется — "Универсальный счетчик IP-трафика через SNMP". Адрес пакета приведен в списке литературы и ссылок.

Помимо этих двух простых способов, есть большое количество программ для подсчета трафика, в частности IpTraf, useripacct, netacct, ipacct.

Ссылки

- **www.linux.org.ru/books/gateway/** — Костарев Алексей Федорович. ОС Linux как мост между локальной сетью и Internet.
- **lin-omts.airport.sakhalin.ru/departs/ccito/guide1.htm** — как установить, настроить и запустить Web-узел UNIX не тратя лишних денег, сил и здоровья.
- **people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg.html** — описание MRTG.
- **www.mrtg.org** — официальный сайт пакета MRTG.
- **rrdtool.eu.org** — официальный сайт пакета rrdtool.
- **www.geocities.com/SiliconValley/Pines/7895/PPP.DOC** — В. Водолазкий. Установка PPP-соединения в Linux.
- **http://linux.perm.ru/doc/net/mrtg.html** — Сергей Богомолов. Мониторинг загрузки каналов (и не только) MRTG.
- **www.bog.pp.ru/work/rrdtool.html** — Сергей Богомолов. RRDtool — хранение и отображение данных мониторинга.

- **linux.uatel.net.ua/ipcount.phtml** — как оперативно подсчитать IP-трафик клиента.
- **ftp://ftp.kiev.farlep.net/pub/os/linux/soft/trafficcounter-snmp** — универсальный счетчик IP-трафика через SNMP.
- Соответствующие HOWTO *(см. гл. 13)*:
  - ISP-Hookup-HOWTO;
  - FIREWALLING_AND_PROXY_SERVER_HOWTO;
  - THE_LINUX_KERNEL_HOWTO.

Глава 31

Настройка модемного соединения

Мы уже настраивали исходящее соединение, когда организовывали шлюз в Интернет для локальной сети. В этой главе мы более полно рассмотрим настройку модемного соединения. А начнем — с теории.

Протокол PPP

Последние года три протокол PPP стал стандартом de-facto для организации соединения по коммутируемым каналам и выделенным линиям. Поэтому для нормальной работы модемного соединения и извлечения из него максимальной пользы необходимо иметь понятие о протоколе PPP. Итак, PPP — это интернет-стандарт по передаче IP-пакетов по последовательным линиям. PPP поддерживает синхронные и асинхронные линии.

Общая информация

Point-to-Point Protocol (PPP, протокол "точка-точка") разработан для инкапсуляции протоколов вида "point-to-point IP". Помимо этого, целями создания протокола PPP было упрощение выдачи и управления IP-адресами, асинхронной и синхронной инкапсуляцией, смешиванием сетевых протоколов (network protocol multiplexing), конфигурированием и тестированием качества связи, обнаружением ошибок и опциями для установления таких особенностей сетевого уровня, как настройка адресов и установка сжатия данных. Для поддержки вышеперечисленных качеств PPP должен предоставлять управление по расширенному протоколу Link Control Protocol (LCP, протокол управления соединением) и семейству протоколов Network Control Protocols (NCPs, протоколы управления сетью), которые используются для установления параметров связи. На сегодняшний день PPP поддерживает не только IP, но и другие протоколы, включая IPX и DECNet.

Свойства протокола PPP

В табл. 31.1 приведены основные возможности, реализованные протоколом PPP. Однако следует учитывать, что программное обеспечение может не в полной мере воплощать эти возможности, а зачастую и привносит что-то свое, поэтому прежде чем пытаться реализовать то или иное свойство, заявленное в стандарте протокола PPP, рекомендуется предварительно ознакомиться с описанием используемых программ, особенно в гетерогенной среде.

Таблица 31.1. *Основные возможности, реализуемые протоколом PPP*

Свойство	Описание
Demand on dial (дозвон по запросу)	Подключение PPP-интерфейса и набор телефонного номера по приходу пакета. Отключение интерфейса PPP после некоторого периода отсутствия активности
Redial	Подключение PPP-интерфейса, который потом не будет отключен и будет всегда сохранять в своем распоряжении подключенный канал
Campling	См. Redial
Scripting	Настройки через серию сообщений или промежуточных соединений для установления PPP-соединения — больше похоже на последовательности, используемые для установления связи по UUCP
Parallel	Конфигурирование нескольких PPP-линий для одного и того же подключения к хосту в целях равномерного разделения трафика между ними (в процессе стандартизации)
Filtering	Выбор, при каких пакетах имеет смысл начинать прозвон по линии, а при каких нет, отталкиваясь в принятии решения от IP- или TCP-типа пакета или TOS (Type of Service). К примеру, игнорировать все ICMP-пакеты
Header Compression (сжатие заголовка)	Сжатие TCP-заголовка в соответствии с RFC1144
Server	Принятие входящих PPP-соединений, которые могут также требовать дополнительной маршрутизации
Tunneling	Построение виртуальных сетей по PPP-соединению, через TCP-поток, через существующую IP-сеть. (Build a virtual network over a PPP link across a TCP stream through an existing IP network.)
Extra escaping	Байт-ориентированные символы, не входящие в стандартный набор символов, используемый при установлении связи, они могут быть сконфигурированы отдельно, но также не пересекаться с теми, что используются при установлении связи

Как видите, возможности протокола богатые, и не удивительно, что некоторые возможности не реализованы в полной мере.

Составляющие PPP

PPP предоставляет возможность передачи датаграмм по последовательным point-to-point-линиям и имеет три составляющие:

- метод предоставления инкапсуляции датаграмм по последовательным PPP-линиям с использованием HDLC (High-Level Data Link Control, высокоуровневого управления данными соединения) — протокол для упаковки датаграмм по PPP средствами связи;
- расширенный протокол LCP для установления, конфигурирования и тестирования физического соединения;
- семейство протоколов NCP для установления и управления другими сетевыми протоколами, что позволяет протоколу PPP поддерживать одновременно несколько сетевых протоколов.

Функционирование протокола PPP

В момент установления связи через PPP-соединение PPP-демон вначале шлет пакеты LCP для конфигурирования и тестирования линии связи. После того как связь и дополнительные возможности будут установлены посредством протокола LCP, PPP-демон посылает NCP-фреймы для изменения и настройки одного или более сетевых протоколов. По окончании процесса настройки сетевые пакеты могут передаваться через установленное соединение. Оно будет оставаться активным до тех пор, пока специальные LCP- или NCP-пакеты не закроют соединение, или до тех пор, пока не произойдет какое-нибудь внешнее событие, которое приведет к потере соединения, например, сработает таймер отсутствия активности или разорвется модемное соединение.

Поддерживаемое оборудование

Протокол PPP адаптирован для работы с любым DTE/DCE интерфейсом, включая RS-232, RS-422, RS-423, CITT V.35. Помимо этих интерфейсов протокол может работать практически на любом оборудовании, единственное требование — наличие дуплексного режима.

Структура пакета протокола PPP

Протокол PPP использует принципы, терминологию и структуру пакетов, описанных в стандартах ISO, касающихся HDLC:

- ISO 3309-1984/PDAD1 "Addendum 1: Start/stop transmission";

- ISO 3309-1979 — описывает структуру пакетов HDLC для использования в синхронных системах;
- ISO 3309:1984/PDAD1 — описывает предложения по изменениям в ISO 3309-1979, которые позволяют использовать асинхронные системы.

На рис. 31.1 изображен формат пакета протокола PPP.

Величина поля, байт	1	1	1	2	Переменный	2 или 4
Назначение	Флаг	Адрес	Управление	Протокол	Данные	Контрольная сумма

Рис. 31.1. Структура пакета протокола PPP

Рассмотрим значения полей пакета протокола PPP:

- `Флаг` — один байт, обозначающий начало или конец пакета. Поле флага содержит двоичную последовательность: 01111110;
- `Адрес` — один байт, содержащий двоичную последовательность: 11111111, стандартный широковещательный адрес. PPP не поддерживает индивидуальную адресацию станций;
- `Управление` — один байт, содержащий двоичную последовательность: 00000011, который посылается для передачи пользовательских данных в неразделенных пакетах;
- `Протокол` — 2 байта определяют протокол, упакованный в пакете протокола PPP. Значения протоколов можно узнать в соответствующем RFC;
- `Данные` — 0 или больше байтов, составляющих датаграмму протокола, указанного в поле `Протокол`. Конец информационного поля определяется нахождением заканчивающей последовательности и 2-байтной последовательности в поле контрольной суммы. По умолчанию максимальная длина поля данных 1500 байтов. Однако во время установления сеанса программы pppd могут договориться использовать другое значение поля данных;
- `Контрольная сумма` — обычно 16 битов. Однако при установлении соединения pppd могут договориться об использовании 32-битной контрольной суммы.

PPP-протокол управления соединением (LCP)

PPP-протокол управления соединением (LCP) предоставляет методы для установления, конфигурирования, поддержания и тестирования PPP-соединения. Протокол LCP выполняет функции, приведенные ниже:

- Конфигурирование и установление связи. Перед передачей какой-либо информации (к примеру, пакет IP) протокол LCP должен открыть соеди-

нение и произвести начальный обмен параметрами настройки. Этот этап заканчивается, когда пакет о подтверждении произведенной настройки будет послан и принят обратно.

- Определение качества связи. Протокол LCP позволяет (но эту возможность зачастую не используют) добавить фазу тестирования канала связи. Тестирование канала связи должно происходить сразу же за конфигурированием и установлением связи. Во время проверки качества связи определяется — способно ли соединение с достаточным качеством транспортировать какой-либо сетевой протокол.
- Установление настроек сетевого протокола. После того как протокол LCP закончит определение параметров связи, сетевые протоколы должны быть независимо друг от друга настроены соответствующими протоколами NCP, которыми могут в любой момент времени начать или прекратить пользоваться.
- Окончание связи. Протокол LCP может в любое время прервать установленную связь. Это может произойти по требованию пользователя или из-за какого-нибудь события, к примеру, потери несущей или истечению допустимого периода времени неиспользования канала.

Существуют три типа LCP-пакетов:

- пакеты установления — используются для установления и настройки связи;
- пакеты прерывания — используются для прерывания установленной связи;
- пакеты сохранения связи — используются для управления и диагностики связи.

Сокращения, используемые при описании протокола PPP

В табл. 31.2 приведены некоторые аббревиатуры, используемые при описании протокола PPP. Расшифровка содержит как английское значение, так и русский перевод.

Таблица 31.2. *Аббревиатуры, используемые при описании протокола PPP*

Аббревиатура	Расшифровка
ack	Acknowlegement — получено
AO	Active Open [state diagram] — соединение активно
C	Close [state diagram] — соединение закрыто
CHAP	Challenge-Handshake Authentication Protocol (RFC1334) — протокол аутентификации
D	Lower layer down [state diagram] — нижний уровень отсутствует

Таблица 31.2 *(продолжение)*

Аббревиатура	Расшифровка
DES	Data Encryption Protocol — протокол шифрования данных
DNA	Digital Network Architecture — архитектура цифровых сетей
IETF	Internet Engineering Task Force — организация, непосредственно отвечающая за разработку протоколов и архитектуры сети Интернет
FCS	Frame Check Sequence [X.25] — проверочная последовательность кадра
LCP	Link Control Protocol — протокол управления соединением
LQR	Link Quality Report — отчет о качестве соединения
MD4	MD4 digital signature algorithm — протокол цифровой подписи
MD5	MD5 digital signature algorithm — протокол цифровой подписи
MRU	Maximum Receive Unit — максимальная величина принимаемого кадра
MTU	Maximum Transmission Unit — максимальная величина передаваемого кадра
NAK	Negative Acknowledgement — негативный ответ
NCP	Network Control Protocol — протокол сетевого управления
PAP	Password Authentication Protocol (RFC1334) — протокол аутентификации
PDU	Protocol Data Unit — пакет
PO	Passive open — пассивное соединение
PPP	Point to Point Protocol — протокол "точка-точка"
RCA	Receive Configure-Ack — принят конфигурационный запрос
RCJ	Receive Code-Reject — принят код отклонения
RCN	Receive Configure-Nak or -Reject — принят код отклонения
RCR+	Receive good Configure-Request [state diagram] — принят нормальный запрос конфигурации
RER	Receive Echo-Request — принят эхо-запрос
RTA	Receive Terminate-Ack [state diagram] — принят запрос на разрыв соединения
RUC	Receive unknown code [state diagram] — принят неизвестный код
SCA	Send Configure-Ack [state diagram] — послан конфигурационный запрос

Таблица 31.2 *(окончание)*

Аббревиатура	Расшифровка
SCJ	Send Code-Reject [state diagram] — послан код отклонения
SCN	Send Configure-Nak or -Reject [state diagram] — послан код отклонения
ST-II	Stream Protocol — потоковый протокол
TO+	Timeout with counter > 0 [state diagram] — счетчик тайм-аута больше, чем ноль
TO–	Timeout with counter expired [state diagram] — счетчик тайм-аута превысил предел
VJ	Van Jacobson (RFC1144 header compression algorithm) — алгоритм компрессии заголовка пакетов PPP
XNS	Xerox Network Services — сетевые службы Xerox

Стандарты, описывающие протокол PPP

В табл. 31.3 приведены стандарты (RFC) протокола PPP.

Таблица 31.3. *Стандарты протокола PPP*

Номер RFC	Название
1144	Compressing TCP/IP headers for low-speed serial links — Сжатие заголовков пакетов для низкоскоростных последовательных соединений
1220	Point-to-Point Protocol extensions for bridging — Расширение протокола PPP
1332	PPP Internet Protocol Control Protocol (IPCP) — Управляющий протокол IP
1333	PPP link quality monitoring — Контроль качества соединения PPP
1334	PPP authentication protocols — Протоколы аутентификации PPP
1547	Requirements for an Internet Standard Point-to-Point Protocol — Требования для интернет-стандарта PPP
1552	The PPP Internetwork Packet Exchange Control Protocol (IPXCP) — Управляющий протокол обмена пакетами для разнородных сетей
1570	PPP LCP Extensions — Расширения протокола LCP
1598	PPP in X.25 — Использование протокола PPP в сетях X.25
1618	PPP over ISDN — Использование протокола PPP поверх протокола ISDN

Таблица 31.3 *(окончание)*

Номер RFC	Название
1619	PPP over SONET/SDH — Использование протокола PPP поверх протокола SONET/SDH
1638	PPP Bridging Control Protocol (BCP) — Протокол управления PPP
1661	The Point-to-Point Protocol (PPP) — Протокол "точка-точка"
1662	PPP in HDLC-like Framing — PPP в HDLC-подобных кадрах
1663	PPP Reliable Transmission — Надежная передача PPP
1717	The PPP Multilink Protocol (MP) — Многопоточный PPP-протокол

Настройка сервера входящих звонков (dial-in)

В этом разделе мы перейдем к сугубо практическим действиям — настроим наш сервер для приема входящих звонков. Поскольку мы уже рассматривали в *гл. 30*, как настроить компьютер с операционной системой Linux таким образом, чтобы он выступал шлюзом для вашей локальной сети, — то в этой главе нам будет значительно проще.

Итак, мы умеем настраивать систему таким образом, чтобы она выступала в роли шлюза для вашей локальной сети. Но бывает нужно получить доступ к локальной сети организации, например из дома, в том числе так же из дома выйти в Интернет через корпоративную локальную сеть. Нет ничего проще. Настраивать PPP-соединение мы уже научились, осталось только установить программу, которая умеет "поднимать трубку" модема по входящему звонку и совершать некоторые дополнительные действия. Такой программой является mgetty — признанный фаворит в своей области, умеющий помимо всего прочего посылать и принимать факсы, а также с помощью голосового модема принимать и отправлять voice mail — голосовую почту.

Настройка mgetty

Обычно mgetty, как и ppp, входит в стандартную поставку дистрибутива. Единственное, что необходимо проверить, был ли пакет mgetty скомпилирован с опцией `-DAUTO_PPP`, и если нет, то пакет следует перекомпилировать с этой опцией (в дистрибутиве Red Hat mgetty скомпилирован требуемым нам образом).

После установки mgetty нам надо отредактировать конфигурационные файлы.

В файле /etc/mgetty+sendfax/login.config мы должны написать:

```
/AutoPPP/ – a_ppp /usr/sbin/pppd auth refuse-chap require-pap login
- – /bin/login @
```

Эта строка говорит mgetty следующее:

- после установления входного соединения необходимо вызвать программу pppd;
- для пользователя требуется авторизация;
- аутентификацию по протоколу CHAP — отклонять и требовать авторизации по протоколу PAP.

После установления соединения mgetty анализирует данные, приходящие с модема, и в случае, когда приходит запрос на авторизацию по протоколу PAP, программа сразу же запускает pppd, который и проводит аутентификацию по протоколу PAP.

Далее, нам необходимо отредактировать файл /etc/mgetty+sendfax/mgetty.config приблизительно следующим образом:

```
port ttyS1
speed 115200
data-only y
debug 3
init-chat "" ATZ OK
     answer-chat "" ATA CONNECT \c \r
```

Как видите, модем подключен ко второму последовательному порту, скорость обмена 115200, строка инициализации `ATZ`.

Далее нужно добавить mgetty в файл inittab. Для этого достаточно дописать всего лишь одну строку:

```
S4:2345:respawn:/sbin/mgetty /dev/ttyS1
```

Перегрузив операционную систему, можно приступать к испытаниям — попробуйте позвонить на телефонный номер, где установлен ваш модем — если все настроено нормально — модем должен "поднять трубку".

Настройка pppd

С настройкой pppd вы уже ознакомились в *гл. 30*. Поэтому, чтобы не повторяться, просто приведем соответствующие конфигурационные файлы с небольшими комментариями.

Файл options.ttyS1 должен содержать следующие данные:

```
# Устройство
lock
```

```
login
auth
modem
crtscts
-chap
+pap
# наш интерфейс : удаленный интерфейс
192.168.10.100:192.168.10.101
# маска подсети
netmask 255.255.255.0
# адрес сервера DNS для клиента Windows
ms-dns 192.168.10.100
```

Файл /etc/ppp/pap-secrets должен содержать следующие данные:

```
user1  сервер.домен  ""      *
user2  сервер.домен  ""  *
```

где:

- `user1` — имя пользователя, причем он должен существовать в вашей системе, где установлен модем;
- user2 — сервер, на котором будет проводиться аутентификация; в нашем случае вместо `сервер.домен` необходимо поставить имя компьютера, где расположен модем;
- `""` — отсутствие пароля указывает на то, что пароли необходимо брать из файла /etc/shadow;
- `*` — абонент может производить аутентификацию с любого IP-адреса.

Вот и все — вы стали микропровайдером, причем пользователям Windows сильно облегчили жизнь, поскольку IP-адрес и адрес DNS-сервера вы выдаете автоматически, кроме того, отпадает потребность в использовании скрипта для соединения.

Настройка callback-сервера

Итак, вы настроили свой dial-in-сервер, попользовались им какое-то время и захотели чего-то другого. Например, в вашем городе повременная оплата и часами работать в Интернете из дома не получается, но руководство вашей организации не возражает против того, чтобы вы работали за ее счет. Дело за малым — организовать ваш dial-in-сервер таким образом, чтобы не вы ему звонили, а он вам. В компьютерных документах такой сервер зовется callback-сервером. Функционирует он следующим образом.

Сначала клиент дозванивается через модем к callback-серверу. Модем на сервере настроен на прием входящих звонков (установку и настройку dial-in-сервера мы только что рассмотрели). После установки соединения сервер предлагает клиенту пройти аутентификацию. Клиент подключается к нему как особый callback-пользователь. После этого модем на сервере обрывает связь и сам звонит клиенту по номеру, который закреплен за компьютером клиента. Модем на клиентском компьютере готов принять обратный звонок, и после установления соединения происходит повторная авторизация. По окончании аутентификации устанавливается PPP-соединение. Далее клиент работает обычным образом.

Конфигурация callback-сервера

После того как настройка dial-in-сервера завершена, необходимо настроить callback. Для этого надо выполнить следующие действия:

1. Создать нового пользователя back.
2. Создать пустой файл с именем callback.conf в /etc/mgetty/.
3. В файл /etc/mgetty/login.config добавить следующую строку:

   ```
   back – – /usr/sbin/callback -S 1234567
   ```

   После ключа `-S` указывается номер, по которому сервер должен сделать обратный звонок клиенту.

Конфигурация клиентов

Поскольку сервер мы уже сконфигурировали, необходимо сконфигурировать клиента и проверить, каким же образом работает callback. Начнем с операционной системы Linux.

Конфигурирование Linux-клиента

Для конфигурирования клиента Linux необходимо выполнить следующее:

1. Создать файл /etc/ppp/options, в котором должны быть такие строки:

   ```
   lock
   defaultroute
   noipdefault
   modem
   115200
   crtscts
   debug
   passive
   ```

2. Создать файл ppp-callback в /etc/ppp/peers/, в котором должны быть такие строки:

```
ttyS1 33600 crtscts
connect '/usr/sbin/chat -v -f /etc/ppp/chat-callback'
noauth
```

3. Создать файл /etc/ppp/chat-callback, в котором должны быть такие строки:

```
ABORT BUSY
ABORT VOICE
ABORT "NO DIALTONE"
ABORT "NO ANSWER"
"" ATZ
OK ATDP7654321            # Телефонный номер сервера
CONNECT \d\d
ogin: \q\dback
TIMEOUT 90
RING AT&C0S0=1
ogin: \q\dvasya
assword: \q\dpasswordforvasya
```

   В файл chat-callback необходимо вписать телефон callback-сервера, имя и пароль пользователя.

4. Создать файл /usr/bin/pppcall, в котором должны быть такие строки:

```
#!/bin/bash
/usr/sbin/pppd -detach call ppp-callback &
```

   Сделать этот файл исполняемым.

Теперь для того, чтобы позвонить на ваш сервер, достаточно запустить скрипт pppcall.

Конфигурирование клиента MS Windows 98

Для Windows конфигурация производится по-другому. Выполните команду меню **Пуск | Программы | Стандартные | Удаленный доступ к сети | Новое соединение**. Укажите данные, необходимые для дозвона к серверу. Помимо этого, в настройках модема на вкладке **Подключения** нажмите кнопку **Дополнительно** и в строке инициализации модема укажите следующее:

```
&c0s0=1
```

Теперь пробуем дозвониться до нашего сервера. После дозвона в открывшемся окне терминала вы увидите приглашения для аутентификации.

Зарегистрируйтесь в системе как back. После этого модем со стороны сервера оборвет связь, подождет несколько секунд и перезвонит вам. После установки callback-соединения вам предложат пройти повторно авторизацию. Введите ваш нормальный логин и пароль и нажмите кнопку **Продолжить** в окне терминала. Все.

Настройка модемного соединения для пользователя

С настройкой сервера вы уже знакомы. Пора приступать к настройке модемного соединения клиента. Но предварительно поговорим о модемах.

Модемы бывают трех классов.

- Наружный модем, подключаемый к последовательному порту (нормальный, аппаратный модем).
- Внутренний модем (нормальный, аппаратный модем, обычно с интерфейсом ISA).
- Win-модем (наружный модем, подключаемый к USB-порту, или внутренний модем с интерфейсом PCI).

С первыми двумя понятно — поставил, настроил, работай.

С Win-модемом все несколько сложнее. Идея этого модема заключается в том, чтобы упростить и удешевить модем за счет того, что вся обработка сигнала после преобразования из аналогового вида в цифровой возложена на процессор компьютера и драйвер модема. Поэтому в требованиях к аппаратным средствам для этого модема указан процессор не ниже Pentium 166 и объем оперативной памяти не менее 32 Мбайт. А Win-модемом такие устройства назвали потому, что драйверы первоначально были написаны только для Windows. Сказать, что Win-модем работает хорошо, особенно на наших телефонных линиях, — нельзя. На нормальной телефонной линии и цифровой АТС Win-модем может устойчиво работать, правда скорости выше 44 000 бит/с вы никогда не получите, а реальная скорость будет где-то возле 28 800—33 600 бит/с. Причем, по опыту работы, Win-модем на чипе от Lucent более послушный в настройке и несколько лучше себя ведет, чем модемы на чипе Conexant или Pctel.

Предположим, вы купили Win-модем и хотите его настроить под операционной системой Linux. Еще года полтора назад это бы не удалось — производители модемов драйверы под Linux не выпускали, спецификаций на модем сторонним разработчикам не давали, а самостоятельно реализовать в драйвере протокол V.34 доступно только программистам экстра-класса. Но в последнее время индустрия разворачивается к Linux лицом — выпускаются драйверы, некоторая часть даже с исходным кодом.

Первое, что следует сделать — найти на сайте производителя модема или производителя модемного чипа драйвер под Linux. Сходите также по ссылкам, приведенным в конце главы, например на **www.linmodems.org**, — наверняка это вам поможет. Далее действуйте по инструкции, прилагаемой к драйверу.

Настройка модема в текстовом режиме

Все просто, идем по пунктам:

1. Создаем файл /etc/ppp/options, в котором содержатся следующие строки:

```
lock
defaultroute
noipdefault
modem
115200
crtscts
debug
```

2. Создаем файл ppp-call в /etc/ppp/peers/, в котором содержатся следующие строки:

```
ttyS1 115200 crtscts
connect '/usr/sbin/chat -v -f /etc/ppp/chat-call'
noauth
```

3. Создаем файл /etc/ppp/chat-call, в котором содержатся следующие строки:

```
ABORT BUSY
ABORT VOICE
ABORT "NO DIALTONE"
ABORT "NO ANSWER"
"" ATZ
OK ATDP7654321            # Телефонный номер провайдера
CONNECT \d\d
ogin: \q\dvasya
assword: \q\dpasswordforvasya
```

   В файл chat-call необходимо вписать телефон дозвона провайдера, имя и пароль пользователя.

4. Создаем файл /usr/bin/pppcall, в котором содержатся следующие строки:

```
#!/bin/bash
/usr/sbin/pppd -detach call ppp-call &
```

   И делаем его исполняемым.

Теперь для того, чтобы позвонить на ваш сервер, достаточно запустить скрипт pppcall.

Настройка модема в X Window

Самый простой путь настройки модема — с помощью графических утилит. В дистрибутиве Red Hat в среде GNOME есть удобная и простая программа — rp3-config. Она находится в разделе **Программы | Интернет**, а пункт меню называется **Dialup Configuration**.

После запуска программы от имени обычного пользователя вы увидите окно (рис. 31.2), в котором необходимо ввести пароль пользователя root.

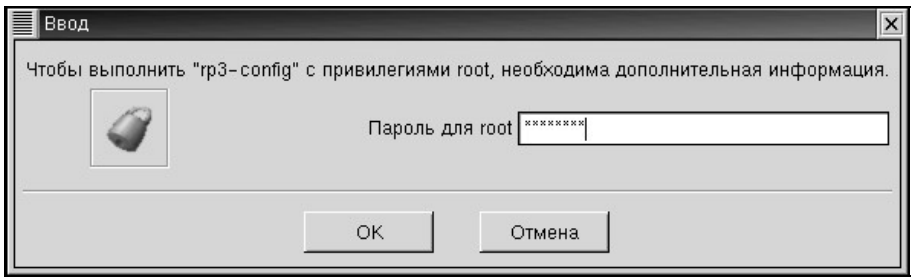

Рис. 31.2. Окно ввода пароля администратора

Вводим его и получаем следующее окно (рис. 31.3), которое уведомляет нас, что мы сейчас будем создавать новое соединение Интернета.

Нажимаем кнопку **Далее** и получаем окно предупреждения (рис. 31.4).

Все правильно, модем только что подсоединен к компьютеру и никаких действий по подключению модема не производилось. Включаем модем и нажимаем кнопку **Далее**. Система производит поиск установленных модемов и выдает список обнаруженных в системе модемов (рис. 31.5).

Если система неверно опознала модем, или ваш модем требует нестандартной скорости подключения — вы можете сейчас подправить эти данные. Идем далее. В следующем окне (рис. 31.6) мы должны определить имя соединения и номер, по которому будем дозваниваться к провайдеру. Вводим все необходимое и двигаемся дальше.

В следующем окне (рис. 31.7) мы вводим имя и пароль пользователя для подключения к провайдеру. Двигаемся дальше.

В следующем окне (рис. 31.8) выбираем обычный тип провайдера и двигаемся дальше.

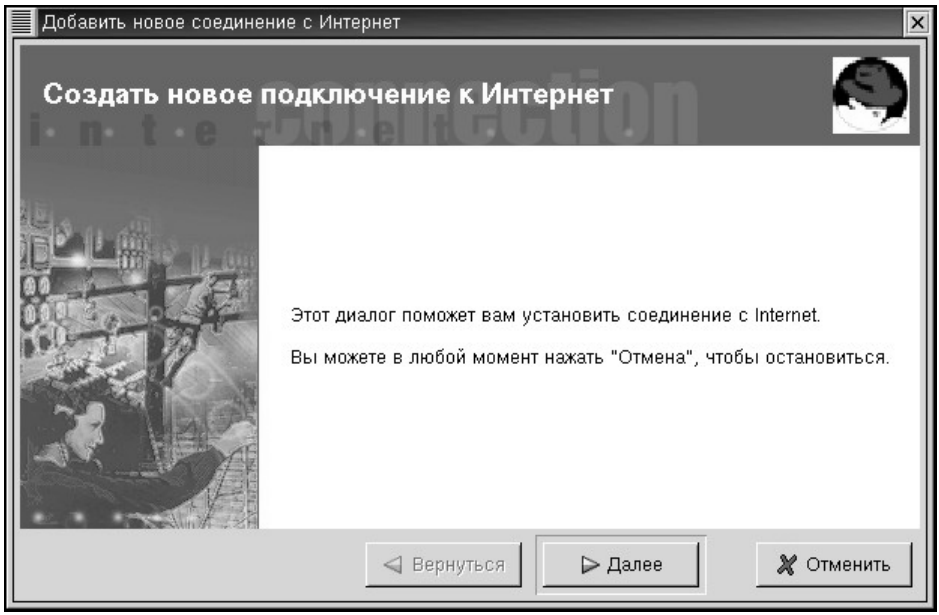

Рис. 31.3. Окно приветствия

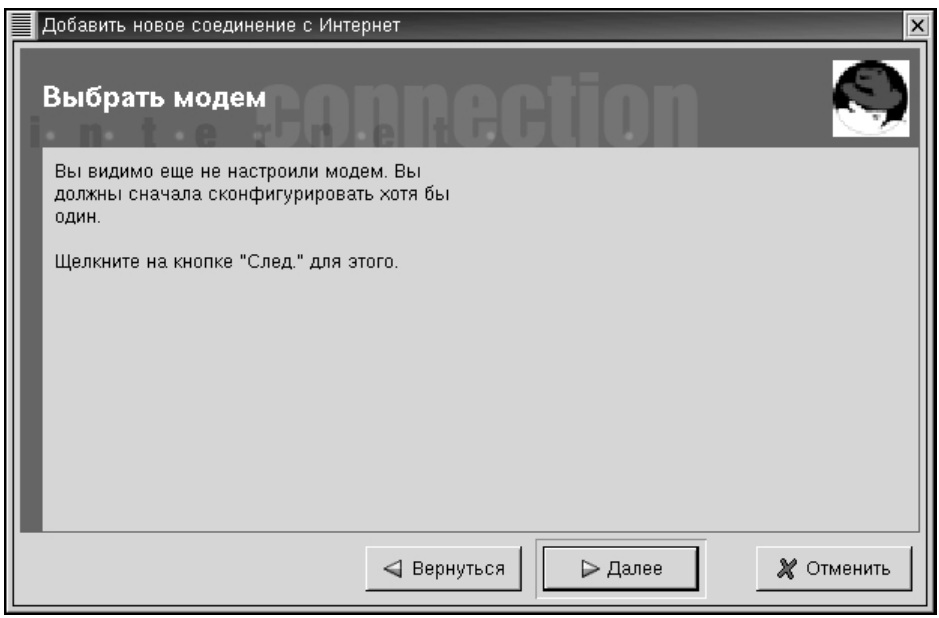

Рис. 31.4. Окно предупреждения об отсутствии сконфигурированного модема

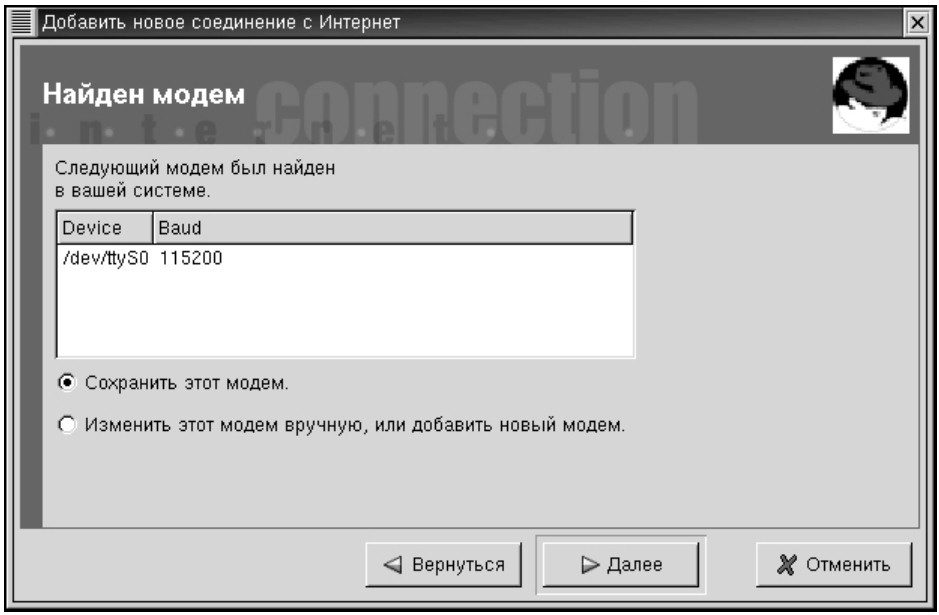

Рис. 31.5. Окно со списком обнаруженных модемов

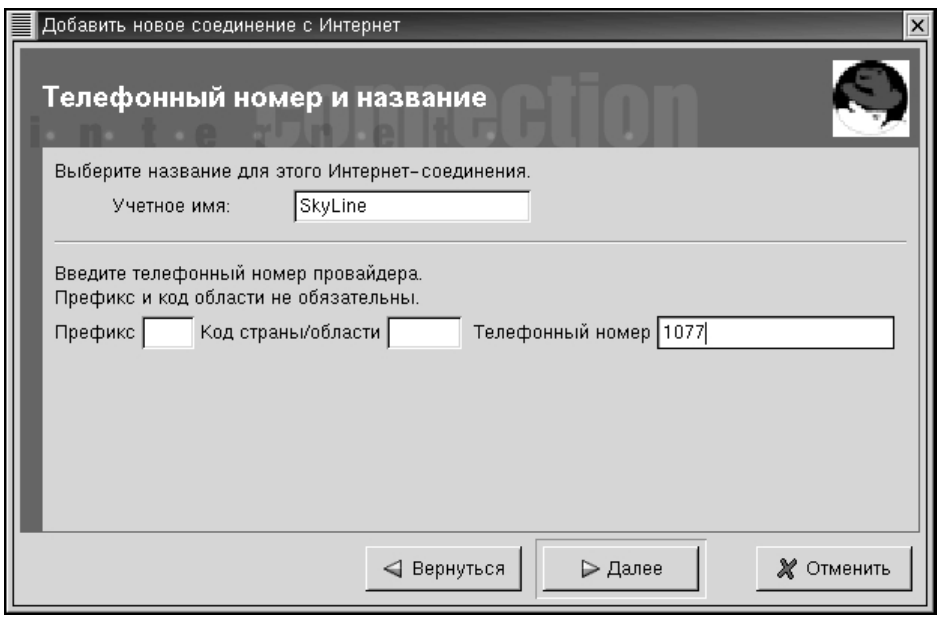

Рис. 31.6. Окно для ввода имени соединения и определения номера дозвона

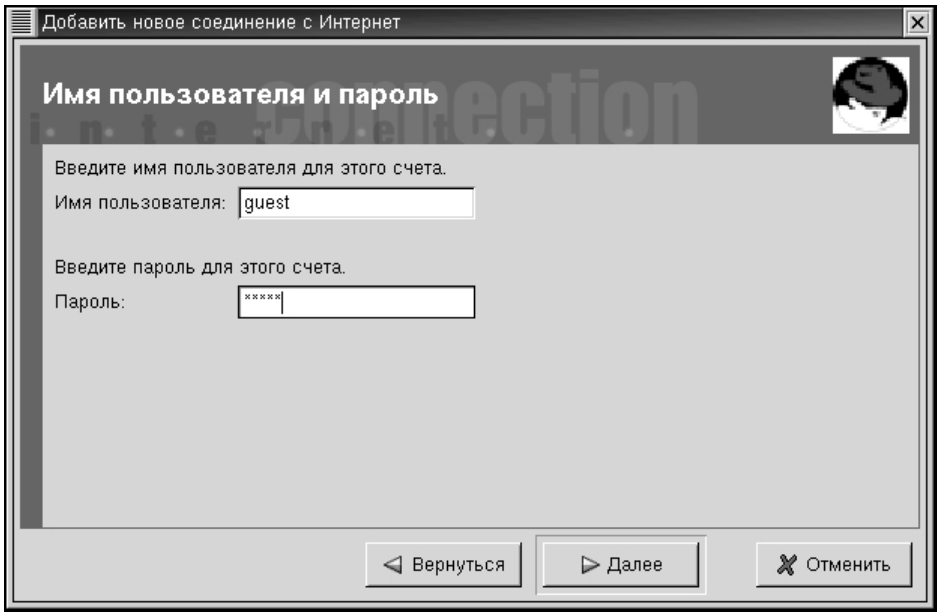

Рис. 31.7. Окно для ввода имени и пароля пользователя

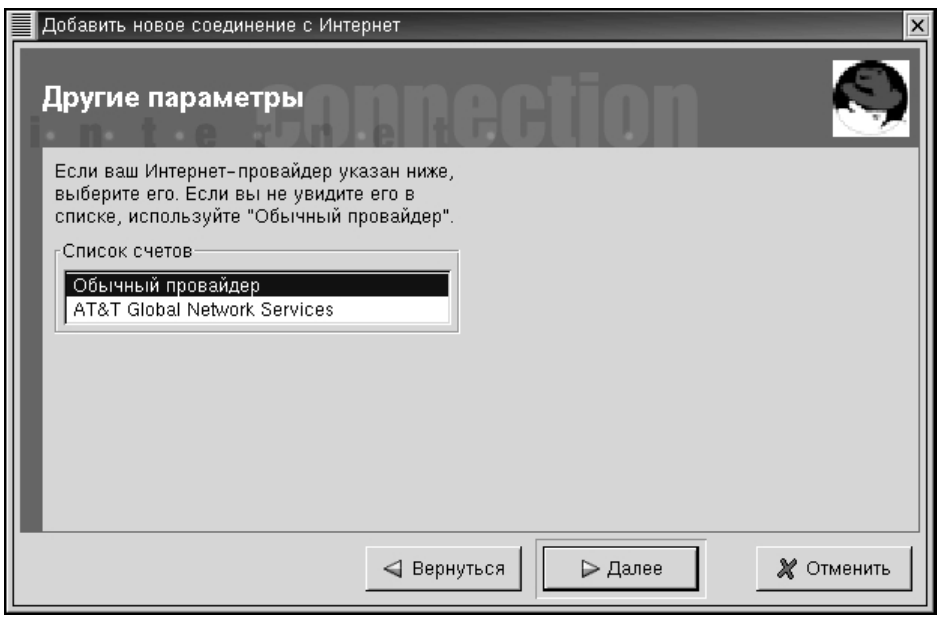

Рис. 31.8. Окно для выбора типа провайдера

Наконец — финал! Проверяем корректность введенных нами данных (рис. 31.9) и нажимаем кнопку **Завершить**. Все! Получаем окно конфигуратора PPP-соединений (рис. 31.10), где видим наше вновь созданное соединение.

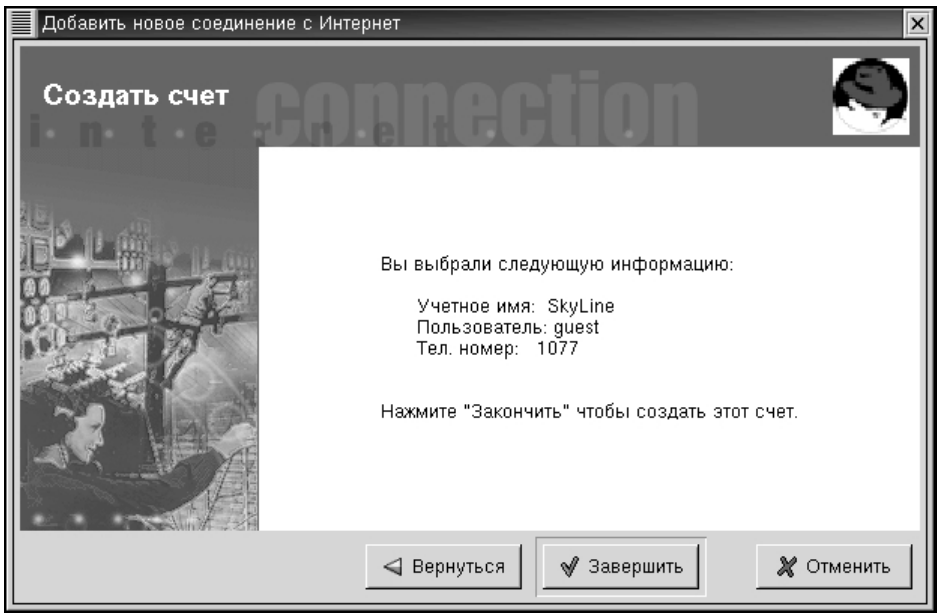

Рис. 31.9. Окно для проверки параметров пользователя

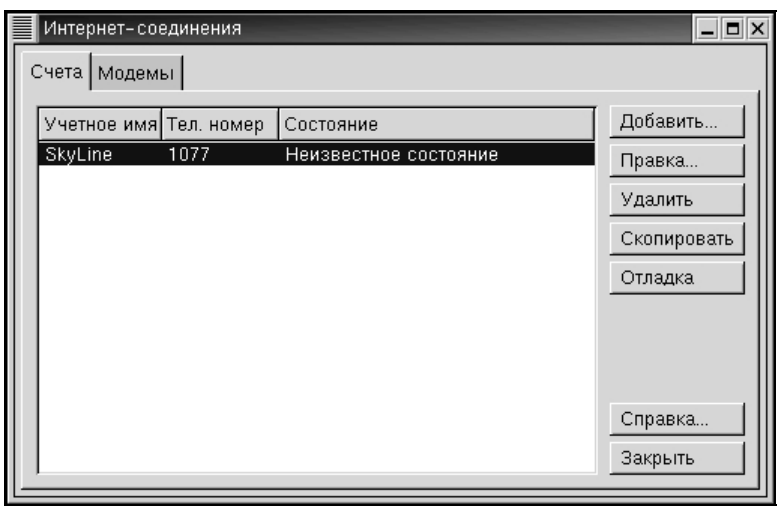

Рис. 31.10. Список интернет-соединений

Теперь переходим к проверке соединения. Для того чтобы установить модемное соединение, идем в раздел **Программы | Интернет**, а пункт меню называется **RH PPP Dialer**. Запускаем.

Появляется окошко (рис. 31.11), в котором выбираем интернет-соединение, которое будем устанавливать, и вперед. В окошке бегает туда-сюда индикатор, модем щелкнул, набрал номер и зашипел.

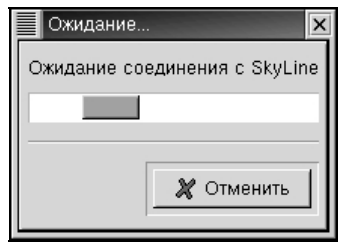

Рис. 31.11. Установка интернет-соединения

Наконец установилось соединение (рис. 31.12). Ну здесь все просто — счетчик времени соединения, график принятых и переданных байтов и одна большая кнопка для разрыва соединения.

Рис. 31.12. Установленное интернет-соединение

Теперь необходимо проверить, правильно ли у нас настроен DNS. Запускаем Mozilla и идем на Web-страничку нашего провайдера (рис. 31.13).

Как видите, все работает — страница грузится, графики переданных и принятых байтов отображаются, время в Интернете считается. Пора разрывать соединение. Нажимаем на единственную кнопку в нашей программе-звонилке и получаем сакраментальный вопрос (рис. 31.14).

Вот и все. Помимо этой утилиты, для конфигурирования можно воспользоваться программой KPP, являющейся неотъемлемой частью KDE, или утилитами pppsetup и wvDial.

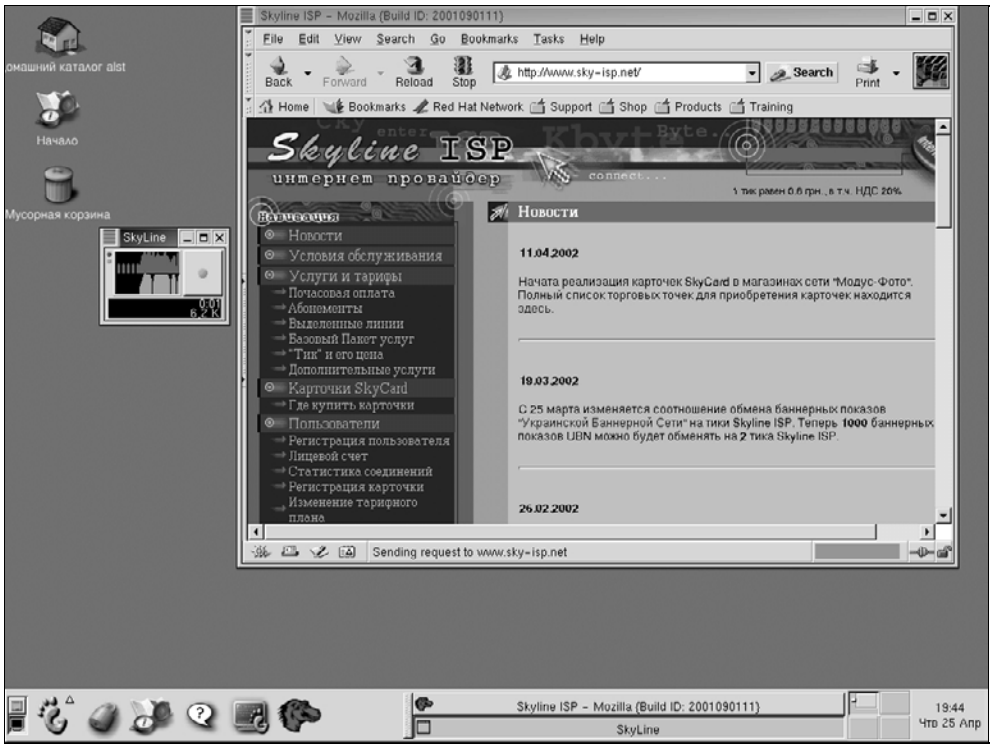

Рис. 31.13. Проверка функционирования интернет-соединения

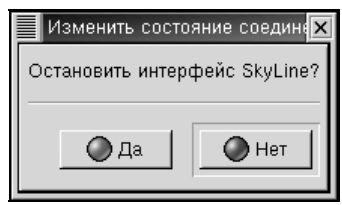

Рис. 31.14. Разрыв интернет-соединения

Ссылки

- **cs.uni-bonn.de/ppp/part1.html, netware.nwsoft.ru** — John Wobus. Протокол PPP. Перевод Виталия Горохова.
- **www.linmodems.org** — драйверы и настройки для Win-модемов на базе чипов различных производителей — Lucent, Connexant (Rockwell), Pctel.

- **www.o2.net/~gromitkc/winmodem.html** — драйверы и настройки для Win-модемов на базе чипов различных производителей — Lucent, Connexant (Rockwell), Pctel.
- **www.idir.net/~gromitkc/winmodem.html** — драйверы и настройки для Win-модемов на базе чипов различных производителей — Lucent, Connexant (Rockwell), Pctel.
- **www.olitec.com/pci56kv2.html** — драйверы для Win-модемов на базе чипа Connexant (Rockwell).
- **www.heby.de/ltmodem/** — драйверы и настройки для Win-модемов на базе чипа Lucent.
- **www.sfu.ca/~cth/ltmodem**/ — драйверы и настройки для Win-модемов на базе чипа Lucent.
- **linux.uatel.net.ua/ppp-dialin.phtml** — настройка PPP dial-in-сервера (PAP-аутентификация).
- **www.softerra.ru/freeos/12279/** — Денис Колисниченко. Пошаговая настройка dial-in-сервера.
- **www.linuxgazette.com** — Sunil Thomas Thonikuzhiyil. Настройка callback-сервера на базе Linux. Перевод Александра Куприна.
- **www.bdcol.ee/linux/callback.shtml** — Linux-callback.
- **www.leo.org/~doering/mgetty/** — документация по Mgetty+Sendfax.
- **http://koi.citforum.tula.ru/operating_systems/articles/ppp.shtml** — В. Водолазкий. Установка PPP-соединения в Linux.
- Документация к программе pppd.
- **linux.yaroslavl.ru/Howto/Howto-mini/call-back-mini-HOWTO.html** — Callback mini-howto (русский перевод).
- PPP-HOWTO *(см. гл. 13)*.

Глава 32

Бездисковые компьютеры

В этой главе мы рассмотрим организацию загрузки бездисковых машин:

- сервер — Linux, клиент — DOS/Windows 3.*x*;
- сервер — Linux, клиент — Linux.

Немного истории

Если проследить за компьютерной периодикой, можно заметить, что несколько лет назад много надежд возлагалось на так называемые "сетевые клиенты". Поясним — это компьютер без винчестера и без прочих дисководов (CD-ROM, ZIP-drive и т. п.), который в общем случае все программное обеспечение, в том числе и операционную систему, загружает из локальной сети (в более общем случае — из Интернета) и результаты работы записывает туда же. Преподносилось это как революция в информационной индустрии, которая позволит резко сократить денежные расходы и высвободит обслуживающий персонал. Постепенно энтузиазм по этому поводу сошел на нет. Давайте посмотрим, почему.

- Во-первых, эта идея была реализована еще в шестидесятых (если не раньше) годах, когда появилась возможность к одному компьютеру подключить несколько терминалов. Чем не сетевой компьютер? Работают несколько человек, информация передается по сети, терминалы стоят недорого, решена проблема с обслуживанием и унификацией программного обеспечения. После появления персональных компьютеров эту идею лишь возродили на новом уровне — есть сервер удаленной загрузки, есть локальная сеть, есть компьютеры, которые не имеют жесткого диска и загружаются с помощью дискеты или напрямую через сеть. Такая система реализовывалась различными программами применениями, в том числе UNIX, Novell, Lantastic и множеством игроков компьютерного мира. Волна "сетевой компьютеризации" периодически возникала, пару лет устойчиво держалась и сходила на нет. И на то есть причины как объективные, так и субъективные.

- Во-вторых, сетевой компьютер не получился дешевым. Правда странно? Но, тем не менее, это так. Сетевой компьютер от фирмы SUN (без монитора) сегодня стоит для конечного пользователя порядка 350—400 долларов. Это без жесткого диска и всяких сменных накопителей. Приблизительно столько, ну, может, дороже долларов на 50—70 стоит "комплектный" персональный компьютер. В результате теряется смысл покупать функционально ограниченное устройство.
- В-третьих, обычное человеческое чувство: "Мне спокойнее, когда мое находится при мне". Как результат — пользователь все равно требует жесткий диск, а если поставят на его компьютер привод CD-RW — тоже не возражает.
- И четвертое — физические ограничения сети. Те времена, когда пользователь работал в текстовой консоли, безвозвратно прошли. Теперь ему требуется только графический интерфейс, причем в разрешении 1024×1280, мультизадачность и чтобы все очень быстро работало. Как результат — операционные системы и приложения занимают десятки, а то и сотни мегабайт дискового пространства. Даже в благополучной Америке в массовых масштабах достичь такой пропускной способности Интернета в ближайшее время не удастся. Конечно, определенным вариантом может стать использование твердотельного Flash-диска, на котором находится операционная система, а все остальное (папки и приложения пользователя) — на серверах в сети. Однако стоимость самого маленького твердотельного диска (32 мегабайта) составляет порядка 50 долларов, поэтому пока не имеет смысла в массовых масштабах использовать такие устройства.

Вот по этим причинам шумиху вокруг сетевого компьютера тихо спустили на тормозах.

Почему мы об этом здесь говорим? Нет, не ради исторического курьеза. Здравые зерна в этой идее имеются. Гранды компьютерного бизнеса пытались протолкнуть сетевой компьютер в массы, "для дома, для семьи". Попытка, заранее обреченная на провал. Но у бездискового или "сетевого" компьютера есть своя законная ниша, в которой он прочно обосновался и вряд ли в ближайшее время ее покинет.

Корпоративный мир. Зачем оператору банка мощный компьютер, если он только вводит цифры и получает информацию (ему и графический интерфейс по большому счету не нужен)? Тем более, что этих операторов в банке сотни. Или кассир в супермаркете, или операторы бюро ремонта в телефонных компаниях. Можно придумать достаточно много работ (те же терминалы в библиотеках), для которых не нужен полноценный компьютер. Что дает использование бездисковых компьютеров для фирм? Первое и самое очевидное — экономия денег. Стоимость даже самого дешевого винчестера порядка 70 долларов, еще десять стоит флоппи-дисковод, тридцать — привод CD-ROM. Итого, экономим около сотни долларов на каждом компью-

тере. Для средней руки банка прямая экономия составит 5—10 тыс. долларов. Но это еще не все! Поскольку от бездисковых компьютеров много не требуется — на нем выполняются одно-два приложения, то вполне возможно использовать старые компьютеры — 486, Pentium. Работать будут до последнего.

Имеется и косвенная экономия — раз нет жестких дисков, значит они не выйдут из строя, не потеряются данные, меньше надо персонала, чтобы обслуживать компьютеры. И администрирование сети становится проще — если надо поставить новую программу, то она устанавливается на одном компьютере и копируется на сервере по пользовательским папкам. Если завис компьютер — нажал кнопку Reset, и через две минуты все снова работает. Резервирование данных намного упрощается — не надо копировать их с каждого компьютера, достаточно сделать резервную копию сервера.

И, что немаловажно, — безопасность сети и данных. На бездисковом компьютере нет жесткого диска, дисководов, приводов CD-ROM — никто не занесет вирус, не перепишет секретные данные, не сможет установить программное обеспечение, идущее вразрез с политикой компании.

Есть еще один любопытный вариант — совмещение загрузки по сети с загрузкой с жесткого диска. Это позволяет использовать компьютер с разными операционными системами или попеременно в качестве персонального или сетевого компьютера.

Конечно, нет бочки меда без ложки дегтя. Бездисковые компьютеры неприменимы для работы с большими объемами данных — графикой, расчетами, требующими сотен мегабайт данных. Ограничение по пропускной способности сети резко сужает круг используемых операционных систем. Архитектура операционных систем так же может быть не приспособлена для использования при удаленной загрузке.

Автор этой книги с бездисковыми компьютерами работает с 1995 года. На Одесской городской телефонной сети большинство компьютеров бюро ремонта являются бездисковыми станциями. Сначала это были компьютеры, загружаемые с дискеты и работавшие в связке Lantastic 6 и Windows 3.11. Потом убрали дисководы, и загрузка компьютеров производилась по сети. Приблизительно через год мы перешли на Novell NetWare 4. Объемы данных росли, нагрузка на сеть увеличилась, сервер баз данных на Novell перестал справляться с нагрузкой, а на новый сервер не выделялось финансирование. На наше счастье, появился бесплатный сервер баз данных под Linux, и после экспериментов было решено перевести на Linux все наше хозяйство. Правда существовала проблема — клиентское программное обеспечение написано под Windows 3.11, и переписывание его заняло бы порядка десяти месяцев. Поэтому приняли решение — организовать загрузку бездисковых клиентов Windows с Linux-сервера. Конечно, можно было бы загружать бездисковых клиентов с помощью MARS, но это потянуло бы за собой установку на кли-

ентах соответствующего программного обеспечения, а наши компьютеры — это, в основном, 486-е и младшие Pentium с 8 Мбайт оперативной памяти.

В качестве операционной системы на бездисковых компьютерах можно использовать Linux или UNIX. Вполне приемлема и MS-DOS. Достаточно просто заставить работать Windows 3.1*x*. Для Windows 95 необходимо достаточно много оперативной памяти и стомегабитная локальная сеть, т. к. Windows вовсю использует свопинг. Что-то более серьезное, типа Windows 98 или Windows NT заставить работать на бездисковых компьютерах не представляется возможным.

Общие вопросы

Процесс загрузки бездисковых компьютеров происходит так. Система состоит из клиентов и сервера. У бездисковых клиентов программа-клиент запрограммирована в микросхеме ПЗУ, которая находится на сетевой карте. На сервере стоит программное обеспечение, которое отслеживает обращения бездисковых клиентов и выдает им соответствующие данные.

Для успешной загрузки по сети бездисковый компьютер должен получить:

- идентификатор, однозначно определяющий этот компьютер;
- образ операционной системы;
- файловую систему, с которой этот компьютер будет работать.

Поскольку в локальной сети обычно несколько сетевых компьютеров, серверу как-то необходимо различать бездисковые компьютеры. Это достаточно просто. У каждой сетевой карты существует свой уникальный адрес (MAC-адрес), который является ее адресом в сети. Первое из требований выполнено.

Получить образ операционной системы позволяет программа, прошитая в ПЗУ, установленном в сетевой карте. Сначала эта программа посылает по сети MAC-адрес и запрос на получение необходимых данных для функционирования сетевого протокола (IP, IPX или NetBIOS — в зависимости от того, какая установлена операционная система и с каким сетевым протоколом она работает). Поскольку мы рассматриваем Linux и IP-протокол, то дальнейший материал касается только IP-протокола.

Протоколы, используемые для получения IP-адреса бездисковым компьютером, называются загрузочным протоколом (BOOTP) и протоколом динамической настройки компьютера (DHCP). Применение протокола DHCP шире, он используется и для динамической настройки обычных компьютеров.

Поскольку первоначально бездисковый компьютер посылает по сети широковещательный запрос, то первый сервер удаленной загрузки, который откликнулся на него, и выдаст бездисковому компьютеру IP-адрес и в дальнейшем произведет его загрузку.

После получения IP-адреса бездисковый компьютер получит образ операционной системы с сервера удаленной загрузки. Для этого используется протокол, имеющий название тривиального протокола передачи файлов (TFTP). Протокол TFTP можно назвать подмножеством протокола FTP, однако в нем нет подтверждения подлинности, и он использует протокол UDP, поскольку код протокола UDP легко разместить в микросхемах ПЗУ.

Передача данных происходит поблочно, после передачи каждого блока сервер удаленной загрузки ожидает подтверждения получения блока. Потерянные блоки после определенного времени ожидания передаются заново. Когда получены все блоки, микросхема ПЗУ сетевой загрузки обращается к образу операционной системы по адресу точки входа.

И последнее. Для нормального функционирования операционной системы компьютеру должна быть предоставлена корневая файловая система (для Linux и UNIX) или сетевые диски для других операционных систем. Linux и UNIX обычно используют сетевую файловую систему (NFS).

Предварительные действия

Что нам надо для того, чтобы создать сервер удаленной загрузки?

- скачать из Интернета пакет Etherboot и/или Netboot (после серии экспериментов рекомендуется Etherboot);
- получить список сетевых карт, установленных в ваших бездисковых компьютерах (тип карты и, желательно, тип микросхемы);
- найти программатор и микросхемы ПЗУ;
- внимательно изучить сопроводительную документацию;
- скачать и установить серверы TFTP и BOOTP, а для бездисковых компьютеров Linux еще и NFS.

Windows-клиенты

Начнем с Windows-клиентов, поскольку это более трудная задача. Помимо пакетов, упоминавшихся ранее, необходимо с сайта Microsoft скачать Microsoft Network Client version 3.0 for MS-DOS (**ftp://ftp.microsoft.com/bussys/clients/msclient/**). Так же необходимо иметь в своем распоряжении MS-DOS 5.0 или выше и дистрибутив Windows 3.1*x*.

План действий

План и порядок действий должны быть примерно такими:

1. Устанавливаем на сервере пакет удаленной загрузки и пакет Samba (файл-сервер для работы с Windows-клиентами, использует протокол NetBIOS поверх TCP/IP).

2. На клиентской машине с жестким диском создаем работоспособную DOS-систему (загружаемую с дискеты) с сетевым клиентом, поддерживающим протоколы NetBIOS и TCP/IP и, по желанию, устанавливаем на жесткий диск Windows и требуемые приложения.
3. Создаем ПЗУ удаленной загрузки (или загрузочную дискету), создаем на сервере Boot-образ дискеты.
4. Копируем на сервер в каталог пользователя нужные приложения.
5. Конфигурируем сетевую карту, устанавливаем ПЗУ удаленной загрузки.
6. Запускаем бездисковые станции.

Установка и настройка программного обеспечения на сервере

Установка пакета Etherboot не должна вызвать никаких сложностей. Предварительно рекомендуем прочитать файл Readme. Установка и настройка пакета Samba так же не представляют трудности, наиболее типичные настройки приведены в документации. Подробную информацию можно найти и в *гл. 24*, посвященной пакету Samba.

Далее следует установить следующие пакеты — сервер BOOTPD и сервер TFTP. После установки эти серверы полагается настроить. Для автоматического старта демона BOOTPD необходимо добавить следующую строчку в файл /etc/inetd.conf:

```
bootps dgram udp wait root /usr/sbin/tcpd bootpd
```

Затем надо создать BOOTP-базу, ставящую в соответствие MAC-адресам сетевых карт бездисковых компьютеров адреса IP и хранящую другую необходимую информацию (более подробную информацию следует смотреть в соответствующей man-странице). Эта база находиться в файле /etc/bootptab и для нашего случая содержит следующие строки:

```
client1:hd=/tftpboot:vm=auto:ip=192.168.40.33:\
:ht=ethernet:ha=008048e2eb9c:\
:bf=bootnet
```

Рассмотрим подробнее поля базы:

- `hd` — домашний каталог, где находится загрузочный образ;
- `ht` — тип устройства;
- `ha` — аппаратный адрес хоста. Для Ethernet-карты это MAC-адрес;
- `ip` — адрес для бездискового клиента;
- `bf` — имя загрузочного образа для бездисковой станции.

Для автоматического запуска сервера TFTP необходимо проверить наличие следующей строки в файле /etc/inetd.conf:

```
tftp dgram udp wait root /usr/sbin/tcpd in.tftpd -s /tftpboot
```

Настройка аппаратуры клиентской машины

Компьютер, на котором будут проводиться опыты, должен удовлетворять следующим минимальным требованиям:

- процессор 386;
- 2 Мбайт оперативной памяти;
- винчестер 20 Мбайт;
- дисковод 3,5 дюйма;
- сетевая карта.

Конфигурирование сетевой карты заключается в разрешении удаленной загрузки и выставлении адреса блока памяти, куда будет отображаться ПЗУ. Как известно, сетевые карты могут настраиваться перемычками или иметь встроенную флэш-память, из которой карта и берет при включении компьютера всю необходимую информацию о настройках.

Будем считать, что сетевая карта уже настроена для работы в обычном компьютере — выставлено прерывание и адрес ввода/вывода. Теперь необходимо разрешить карте работать с ПЗУ. Для этого на карте, конфигурируемой перемычками, необходимо включить перемычку BOOTROM ENABLED и выставить перемычками адрес блока памяти, куда будет отображаться ПЗУ (как правило, это адрес D000, D400). Важно, чтобы этот адрес не был занят системой. Для сетевой карты без перемычек в комплекте с драйверами идет программа конфигурации и тестирования сетевой карты. Документация по программе конфигурации сетевой карты находится на прилагаемой к ней дискете. После успешной конфигурации сетевой карты вставьте микросхему ПЗУ в предназначенную для нее панель.

Замечание

Некоторые сетевые карты требуют для правильной работы удаленной загрузки определить в BIOS диск A: (все равно какого типа).

Установка и настройка программного обеспечения на клиенте

Наша задача — установить и отконфигурировать программы на клиентской машине, чтобы позднее перенести их на бездисковые клиентские компьютеры.

Прежде всего, для подключения к разделяемым ресурсам сервера (каталоги пользователей, общие папки) необходим DOS-клиент, поддерживающий следующие протоколы — TCP/IP и NetBIOS, а также WinSocket (для корректной работы Windows 3.1*x* с нашей сетью). Раз мы решили использовать Windows 3.1*x*, то вполне логично в качестве клиента использовать Microsoft Network Client for MS-DOS version 3.0.

Теперь подготовим клиентский компьютер — на жесткий диск запишем инсталляцию сетевого клиента и Windows. Здесь есть некоторые сложности — поставить клиента на жесткий диск, а потом переписать на дискету и уже с нее запускать клиента не получается — при установке клиента некоторые пути прописываются *прямо* в исполняемый файл.

Поэтому сделаем следующее — создадим на жестком диске каталог \tmp и выполним команду (можно прописать ее в файле Autoexec.bat):

```
subst a: c:/tmp/
```

В результате в системе появится псевдодисковод A:.

После этого начнем установку клиента на псевдодисковод A:. Если в ходе установки клиент зависнет, следует повторить установку. При инсталляции необходимо правильно выставить параметры сетевой карты, выбрать необходимые протоколы, определить имя пользователя и рабочую группу. Будем считать, что клиент установлен успешно.

Теперь надо нормально его настроить, чтобы можно было подмонтировать ресурсы Samba как сетевые диски. Копируем установленного клиента в любой временный каталог и наводим порядок в следующих файлах:

- Hosts;
- Lmhosts;
- Networks;
- Protocol.ini;
- System.ini.

О файлах Hosts, Lmhosts и Networks мы говорить пока не будем. Остановимся на файлах Protocol.ini и System.ini.

В файле Protocol.ini есть следующая секция:

```
[TCPIP]
NBSessions=6
SubNetMask0=255 255 0 0
IPAddress0=0 0 0 0
DisableDHCP=0
DriverName=TCPIP$
BINDINGS=MS$NE2CLONE
LANABASE=1
```

Мы должны привести значения полей `SubNetMask0`, `IPAddress0`, `DisableDHCP` к следующему виду:

```
SubNetMask0=255 255 255 0
IPAddress0=192 168 40 33
DisableDHCP=1
```

где:

- `IPAddress0=192 168 40 33` — адрес бездисковой машины;
- `DisableDHCP=1` — запрещает использование динамической выдачи IP-адресов.

В файле System.ini для нас интересны следующие ключи:

```
[network]
.
computername=A
lanroot=A:\NET
username=A
workgroup=MYGROUP
```

Вот и все. Перезагружаем машину, запускаем файл net и просматриваем доступные соединения. Если вы правильно сконфигурировали клиента и Samba, то должны увидеть в списке доступных ресурсов Linux-сервер с ресурсом `имя_пользователя`. Вот в каталоге /home/имя_пользователя мы и будем держать нужные нам программы (в частности, Windows 3.1*x*).

Далее, с сайта фирмы Microsoft надо загрузить файлы 51.txt и 62.txt, доступные по адресам: **ftp://ftp.microsoft.com/bussys/winnt/kb/Q142/0/62.txt** и **ftp://ftp.microsoft.com/bussys/winnt/kb/Q128/7/51.txt**. В этих файлах описывается, как установить и заставить работать Windows 3.1*x* и Microsoft Network Client version 3.0 for MS-DOS.

После установки Windows 3.1*x* не забудьте в свойствах Windows 3.1*x* установить тип своп-файла как отсутствующий (в дальнейшем это поможет избежать перегрузки сети пересылками туда-сюда файлов подкачки).

Создание загрузочной ПЗУ (загрузочной дискеты)

Следующий шаг — найти подходящий образ ПЗУ. Обычно в нашей стране в персональные компьютеры устанавливали один из клонов сетевой карты NE2000. Пакет Etherboot в своем составе имеет большое количество образов ПЗУ, в том числе и NE2000 (полный список сетевых карт, для которых имеются образы ПЗУ, следует смотреть в документации к Etherboot).

Допустим, наша сетевая карта — клон NE2000. Образ ПЗУ для нее — файл ne.lzrom, находящийся в каталоге src-32. Следующее, что надо сделать, пе-

ред тем как программировать ПЗУ, проверить, действительно ли имеющаяся карта будет корректно работать с такой прошивкой ПЗУ. Для этого необходимо сделать специальную дискету, на которой записана загрузочная программа и образ ПЗУ для нашей карты. Напомним, что для карты NE2000 берется файл ne.lzrom, находящийся в каталоге src-32. Затем создаем загрузочную дискету командой:

```
cat floppyload.bin ne.lzrom /dev/fd0
```

(fd0 для дисковода A: и fd1 для дисковода B:).

После этого можно пробовать загрузить наш клиентский компьютер с полученной дискеты. Вы должны увидеть сообщение о старте TFTP-сервиса, о получении вашей бездисковой машиной IP-адреса и сообщение об отсутствии загрузочного образа.

Не надо пугаться этого сообщения — мы убедились, что образ ПЗУ благополучно загрузился, стартовала удаленная загрузка, но не был найден образ загрузочной дискеты. Все правильно, мы ведь ее еще не создали.

Теперь берем микросхему ПЗУ с ультрафиолетовым стиранием модели 2764 (можно, конечно, и советский аналог К273РФ6, но сейчас проще найти деталь производства Intel или других зарубежных производителей), берем программатор, файл ne.lzrom и прошиваем ПЗУ.

Наконец, у нас есть запрограммированная микросхема ПЗУ. Теперь, чтобы избежать порчи микросхемы, надо ее *правильно* установить в панель. В инструкции к сетевой карте обычно это подробно расписано. На короткой стороне микросхемы и панели есть выемка (так называемый ключ). Необходимо вставить микросхему ПЗУ так, чтобы выемки на панели и на микросхеме были с одной стороны.

Создание загрузочного образа дискеты

А теперь — изготовление загрузочного образа. Пойдем от простого к сложному, попробуем загрузить по сети операционную систему MS-DOS. Впрочем, тут мы имеем два варианта — загрузочную дискету можно делать, а можно и не делать. В отличие от Nowell или Lantastic, допускается скопировать в отдельный каталог на винчестере необходимые файлы и работать с ними вместо того, чтобы постоянно переписывать дискету.

Итак, делаем загрузочную дискету — берем DOS 5.0 (или выше, но помните, чем новее версия DOS, тем больше системные файлы при близкой функциональности), запишем еще, например, Volkov Commander, создадим файлы Config.sys и Autoexec.bat. Загружаемся с дискеты, проверяем работоспособность на Volkov Commander. Все работает.

Теперь необходимо специальной программой создать образ загрузочной системы. Генератор образа дискеты называется mknbi-linux для загрузки LINUX,

mknbi-dos для загрузки DOS. Эта утилита входит в состав пакета Netboot, который распространяется как отдельно, так и в составе Etherboot (каталог /netboot) Описание утилиты можно посмотреть по команде `man mknbi-dos`. Сама утилита mknbi-dos находится в каталоге /usr/local/bin. Если она отсутствует — необходимо ее скомпилировать.

Вводим следующую команду:

```
mknbi-dos r /dev/fd0 o bootnet
```

где:

- `/dev/fd0` — источник файлов для загрузочного образа (в данном случае — дискета);
- `bootnet` — имя файла — загрузочного образа.

Таким образом, мы получили загрузочный образ с дискеты.

Для создания загрузочного образа из файлов, находящихся на винчестере, делаем следующее — создаем каталог (например /t) и переписываем туда нужные файлы. А потом создаем образ командой:

```
mknbi-dos r /t o bootnet
```

Вот мы и получили загрузочный образ с именем bootnet. Теперь копируем (или переносим) его в заранее созданный нами каталог /tftpboot.

Загрузка бездисковой машины

Теперь пришла пора испытать нашу систему. У подопытной машины отключаем в BIOS винчестер и дисководы (для чистоты эксперимента), устанавливаем ПЗУ в сетевую карту и включаем. Если все нормально сконфигурировано, получим машину с загруженной DOS и дисководом A:, на котором находятся все файлы с ранее созданной нами загрузочной дискеты (или каталога). А теперь самое время задаться вопросами — загрузили машину по сети, виден наш привычный Volkov Commander, а дальше? Где дисководы, где ресурсы? Тут могут быть следующие варианты: во-первых, существует возможность создать загрузочный образ таким образом, что в DOS загрузочная дискета будет видеться как диск C:. Второй вариант — это, собственно то, ради чего мы все и проделали — раздача по сети ресурсов сервера и принтера, возможность загрузки Windows 3.1*x* (установка и настройка Windows 3.1*x* описана ранее).

Оптимизация бездисковой загрузки

Бездисковую загрузку следует оптимизировать. Microsoft Network Client стандартно занимает приблизительно 1,7 Мбайт. Представляется, что это очень много. Можно сократить до 800 Кбайт.

Вот список файлов, которые необходимо оставить:

- A.PWL;
- CONNECT.DAT;
- DHCP.PRM;
- DNR.EXE;
- EMSBFR.EXE;
- HOSTS;
- IFSHLP.SYS;
- LMHOSTS;
- NDISHLP.SYS;
- NE2000.DOS;
- NEMM.DOS;
- NET.EXE;
- NET.MSG;
- NETBIND.COM;
- NETWORKS;
- NMTSR.EXE;
- PROTMAN.DOS;
- PROTMAN.EXE;
- PROTOCOL;
- PROTOCOL.INI;
- SERVICES;
- SHARES.PWL;
- SOCKETS.EXE;
- SYSTEM.INI;
- TCPDRV.DOS;
- TCPTSR.EXE;
- TCPUTILS.INI;
- TINYRFC.EXE;
- UMB.COM;
- WFWSYS.CFG;
- WSAHDAPP.EXE.

Вот полное содержимое файла Protocol.ini:

```
 [network.setup]
version=0x3110
netcard=ms$ne2clone,1,MS$NE2CLONE,1
transport=tcpip,TCPIP
lana0=ms$ne2clone,1,tcpip
[TCPIP]
NBSessions=6
SubNetMask0=255 255 255 0
IPAddress0=192 168 40 33
DisableDHCP=1
DriverName=TCPIP$
BINDINGS=MS$NE2CLONE
LANABASE=1
[MS$NE2CLONE]
IOBASE=0x320
INTERRUPT=5
DriverName=MS2000$
```

```
[protman]
DriverName=PROTMAN$
PRIORITY=MS$NDISHLP
[MS$NDISHLP]
DRIVERNAME=ndishlp$
BINDINGS=MS$NE2CLONE
```

Вот полное содержимое файла System.ini:

```
 [network]
sizworkbuf=1498
filesharing=no
printsharing=no
autologon=yes
computername=A
lanroot=A:\NET
username=A
workgroup=MYGROUP
reconnect=yes
dospophotkey=N
lmlogon=0
logondomain=
preferredredir=full
autostart=full
maxconnections=8
[network drivers]
netcard=ne2000.dos
transport=ndishlp.sys,tcpdrv.dos,nemm.dos
devdir=A:\NET
LoadRMDrivers=yes
[386enh]
TimerCriticalSection=5000
UniqueDosPSP=TRUE
PSPIncrement=2
[Password Lists]
*Shares=A:\NET\Shares.PWL
A=A:\NET\A.PWL
B=A:\NET\B.PWL
```

Вот полное содержимое файла Tcputils.ini:

```
 [tcpglobal]
drivername=GLOBAL$
```

```
hostname=username
[sockets]
drivername=SOCKETS$
bindings=TCPIP
numsockets=4
numthreads=32
poolsize=3200
maxsendsize=1024
 [DNR]
drivername=DNR$
bindings=TCPIP
nameserver0=192 168 40 233
[telnet]
drivername=TELNET$
bindings=TCPIP
nsessions=0
max_out_sends=0
```

Вот полное содержимое файла Config.sys:

```
FILES=100
dos=high,umb
device=C:\WINDOWS\HIMEM.SYS
device=C:\WINDOWS\EMM386.EXE ram
LASTDRIVE=Z
device=IFSHLP.SYS
STACKS=9,256
```

Вот полное содержимое файла Autoexec.bat:

```
set path=C:\WINDOWS;c:;c:\dos;c:\vc;c:\net
PATH=C:\IDAPI;%PATH%
SET TEMP=C:\WINDOWS\TEMP
Rem следующая строчка используется при отладке на винчестерной машине
subst a: c:\a
A:\NET\net initialize
A:\NET\netbind.com
A:\NET\umb.com
A:\NET\tcptsr.exe
A:\NET\tinyrfc.exe
A:\NET\nmtsr.exe
A:\NET\emsbfr.exe
```

```
A:\NET\dnr.exe
A:\NET\sockets
A:\NET\net start
```

Существует возможность удалить из ОЗУ загрузочный образ дискеты. Как это сделать, смотрите по команде `man ethernet` и в описании rmrd.com.

Linux-клиент

Процесс создания бездискового компьютера с операционной системой Linux в целом схож. Правда, здесь вместо Samba используется сервер NFS, а в остальном разницы практически нет.

Создание загрузочной ПЗУ (загрузочной дискеты)

Для создания загрузочной дискеты предусмотрена специальная маленькая программа (512 байт), которая загружает блоки с дискеты в память и начинает выполнение. Чтобы создать загрузочную дискету, надо только соединить загрузочный блок с соответствующим образом микросхемы ПЗУ. Для этого используется следующая команда:

```
cat floppyload.bin ne.lzrom > /dev/fd0
```

Настройка сервера

Необходимо настроить на сервере удаленной загрузки три службы: BOOTP (или DHCP), TFTP и NFS. Процессы установки и настройки указанного программного обеспечения подробно описаны в документации, входящей в каждый из пакетов.

Для нормального процесса загрузки бездискового клиента необходимо настроить разделы NFS.

Исходя из требований надежности и защищенности локальной сети, использовать корневую файловую систему сервера в качестве файловой системы бездискового компьютера нежелательно, тем более, что для бездискового клиента необходимо создать свои, специфические файлы конфигурации.

В идеале, чтобы создать корневую файловую систему, вам надо знать, какие файлы требуются дистрибутиву вашей операционной системы. При загрузке необходимы файлы устройств, файлы, находящиеся в каталоге /sbin и /etc. Проще сделать копию существующей файловой системы и изменить в ней некоторые файлы для бездискового компьютера. В дистрибутиве Etherboot есть руководство и ссылки на скрипты, которые создают такую файловую систему на бездисковом компьютере из корневой файловой системы сервера.

Настроенное ядро Linux для бездискового компьютера ожидает увидеть корневую файловую систему в каталоге /tftpboot/<IP-адрес бездискового компьютера>, в рассмотренном выше случае — /tftpboot/192.168.40.33.

Далее необходимо в файл /etc/exports на сервере вставить следующую строку:

```
/tftpboot/192.168.1.100 aldebaran.foo.com(rw,no_root_squash)
```

Для некоторых служб нужны права rw. Атрибут `no_root_squash` защищает систему NFS от отображения идентификатора суперпользователя в какой-либо другой. Если этот атрибут не будет задан, то различные демоны могут не заработать.

Теперь запустите службы NFS (rpc.portmap и rpc.mountd) и снова попробуйте бездисковую загрузку. Если все прошло удачно, то ядро сможет подмонтировать корневую файловую систему и пройти все стадии загрузки до появления приглашения входа в систему. Вполне вероятно, что по ходу загрузки у вас будут выдаваться сообщения о проблемах с некоторыми службами. Так и должно быть. Дистрибутивы Linux ориентированы на операции с диском, и поэтому для бездисковой загрузки требуются небольшие изменения. Самой большой неприятностью является зависимость от файлов, находящихся в каталоге /usr во время загрузки — они в процессе загрузки поступают от сервера немного позже. Для решения этой проблемы измените пути таким образом, чтобы необходимые файлы искались в корневой файловой системе.

Конфигурация клиента

Чтобы клиент был правильно сконфигурирован, необходимо скомпилировать ядро операционной системы Linux с поддержкой корневой файловой системы на NFS. Кроме того, следует разрешить получение ядром IP-адреса из запроса BOOTP. Надо также вкомпилировать драйвер для вашей сетевой карты в ядро. Для уменьшения объема ядра можно отключить лишние свойства и опции.

Ядро, полученное после компиляции, необходимо преобразовать в загрузочный образ. Это делается аналогично тому, как мы создавали загрузочный образ дискеты для DOS. Для создания образа воспользуйтесь утилитой mknbi-linux. После создания загрузочного образа, поместите его в каталоге /tftpboot под именем, определенным в /etc/bootptab. Убедитесь, что файл доступен для чтения любому пользователю, потому что у TFTP-сервера нет специальных привилегий.

Дальнейшая проверка загрузки бездискового клиента должна подтвердить правильность наших настроек.

Ссылки

- **www.linuxfocus.org/Russian/September1998/article63.html** — Кен Яп. Введение в сетевую загрузку и протокол Etherboot.
- **alst.odessa.ua** — Алексей Стахнов. Удаленная загрузка. Сервер Linux. Клиентская часть DOS, Windows 3.1. Инструкция по установке и настройке.
- **www.slug.org.au/etherboot/**.
- **ftp.microsoft.com/bussys/clients/msclient/**.
- **ftp.microsoft.com/bussys/winnt/kb/Q142/0/62.txt**.
- **ftp.microsoft.com/bussys/winnt/kb/Q128/7/51.txt**.
- Страницы man для пакетов Etherboot, Netboot.
- Соответствующие HOWTO *(см. гл. 13)*.

Глава 33

Резервное копирование и хранение данных

Резервное копирование выполняется с целью получения копий данных, сохраняемых на случай их потери или разрушения. Подобные копии должны создаваться периодически, в соответствии с заранее установленным графиком. Схемы резервного копирования изменяются в зависимости от размеров и степени охвата резервным копированием операционной системы, а также от выдвигаемых требований по надежности сохранения жизнеспособности системы. Элементы системы резервного копирования должны включать необходимое оборудование, носители резервных копий и специальное программное обеспечение. В качестве оборудования может использоваться достаточно широкий набор аппаратных средств, начиная от обычного дисковода и заканчивая библиотекой ленточных устройств. Тип и количество носителей определяются используемым оборудованием, объемами обрабатываемых данных и выбранной схемой резервирования данных. Используемое программное обеспечение может быть очень разнородным, начиная от бесплатных утилит типа tar, cpio, gzip и заканчивая распределенными системами управления хранилищами данных.

Резервное копирование информации используется для:

- ❒ восстановления файлов, случайно удаленных пользователями или утерянных из-за отказов дисковых устройств;
- ❒ получения периодически создаваемых моментальных снимков (snapshots) состояния данных организации. Эта информация может широко использоваться для различных технических и деловых целей;
- ❒ получения данных для восстановления после аварий. Система резервного копирования обязательно является составной частью любого продуманного плана восстановления системы. В случае широкомасштабных катастроф данные доставляются из архивов, сохраняемых в отдельном помещении.

Планирование резервного копирования

При разработке системы резервного копирования одной из важнейших составляющих является выработка правильного набора требований к комплексу резервного копирования. Постарайтесь учесть все аспекты резервного копирования, сделайте два варианта сметы — минимально необходимый и желательный, и обоснуйте руководству организации необходимость дополнительного финансирования, связанного с резервным копированием данных, критичных для самого существования организации. Постарайтесь реализовать хотя бы минимальный вариант резервного копирования. Если вы работаете в банке или в другом подобном учреждении, где данные стоят очень дорого, то обычно удается реализовать схему резервного копирования по максимуму. Добейтесь письменного одобрения ваших действий (лучше всего приказа по организации), чтобы не было отступлений от утвержденной сметы.

Основным фактором, определяющим стоимость системы резервного копирования, является объем архивируемых данных и время, выделяемое на эту процедуру. Чем больше объем данных, обрабатываемых системой резервного копирования в единицу времени, тем дороже получается создаваемая система. Если на проведение резервного копирования ваше серверное хозяйство может выделить ограниченный интервал времени, или если оно функционирует круглосуточно, при планировании системы резервного копирования следует учитывать это обстоятельство, поскольку изменение резервируемых данных до завершения создания резервной копии приводит к получению некорректной резервной копии системы. В том случае, если вы можете остановить работу сервера либо по окончании рабочего дня сервер не используется, процесс резервирования становится тривиальным и, как правило, достаточно дешевым.

Ежедневные процедуры копирования должны выполняться в то время, когда данные находятся в некотором стабильном состоянии, например после окончания рабочего дня. Если не представляется возможным исключить сервер на время создания резервной копии из производственного процесса, руководство фирмы должно знать о возможных осложнениях — обычно это не совсем корректное сохранение информации. В серьезных серверах баз данных об этой проблеме знают, и существуют методы получения точной резервной копии баз данных.

Большинство систем резервного копирования строится на использовании либо команды `cron`, либо собственных утилит автоматического вызова программ по установленному расписанию. Как правило, они позволяют разрабатывать и поддерживать относительно сложные графики проведения работ. Кроме того, системы резервного копирования могут предусматривать прямое взаимодействие с приложениями с целью запуска их собственных механизмов резервного

копирования. Зачастую для целей резервного копирования используются программы и скрипты собственной разработки, учитывающие особенности функционирования вычислительной среды организации.

Выбор схемы хранения резервных данных является еще одним фактором, оказывающим существенное влияние на стоимость создаваемой системы резервного копирования. Схема хранения должна учитывать специфику организации, а также требования, устанавливаемые контролирующими органами (например для банков требования Центрального банка в отношении резервирования данных очень высоки). Кроме того, при выборе схемы хранения должны учитываться и требования, сформулированные в плане восстановления после аварий.

Большинство систем резервного копирования обеспечивают эффективное использование носителей информации за счет организации, по крайней мере, двух независимых уровней хранения данных. Так, *полная* копия содержит копии содержимого всех без исключения файлов системы. При *инкрементном* копировании в архив помещаются только те файлы, которые были изменены с момента создания последней полной или инкрементной копии. Используя различные алгоритмы резервного копирования, можно разработать стратегию резервного копирования, которая сбалансирует требования к эффективности и надежности.

Приведем пример схемы резервного копирования, которая может быть реализована в достаточно крупной фирме. Все данные копируются по субботам. С воскресенья по пятницу выполняется создание инкрементных копий. Носители информации с еженедельными и ежедневными копиями возвращаются на перезапись через месяц.

Формат хранения резервных копий должен быть таким, чтобы резервную копию при желании можно было развернуть на другой операционной системе (например, Windows). Рекомендуется пользоваться программами tar и gzip, аналоги которых существуют практически в любой операционной системе. Это позволит в случае надобности извлечь нужные файлы практически где угодно.

Чтобы не превратить библиотеку резервных копий в ненужную свалку данных, необходимо озаботиться составлением каталогов данных резервного копирования. Обычно в более или менее серьезных пакетах резервного копирования присутствуют функции ведения каталогов.

Базовые утилиты, в том числе tar и cpio, не позволяют создавать подобные каталоги данных резервного копирования. Если для копирования применяются именно они, то каталоги придется вести либо с помощью специально созданного программного обеспечения, либо вручную.

Некоторые приложения для корректного функционирования требуют абсолютной согласованности наборов данных. Так, системы управления базами

данных обычно имеют собственные средства резервного копирования. Поскольку принадлежащие такого рода системам данные часто находятся в состоянии непрерывного изменения, задача фиксирования их согласованного состояния выходит за рамки возможностей программ типа tar, cpio и dump.

Для разрешения указанной проблемы разработчики обычно включают в подобные системы специальное программное обеспечение, способное зафиксировать в копии согласованное состояние их данных. В API системы могут включаться необходимые вызовы, или администраторам предоставляются специализированные сценарии, вызываемые из приложения. Поскольку такие приложения и системы копирования не имеют единого интерфейса, потребуется самостоятельно создать связующие программы промежуточного уровня.

В тех случаях, когда деловой процесс позволяет останавливать работающие с базами данных приложения, а сами базы данных периодически закрываются, может применяться и обычная схема резервного копирования.

Те серверы баз данных, которые поддерживают репликацию, также нуждаются в создании резервных копий. Репликация не защищает от случайного или преднамеренного удаления данных. Кроме того, сетевые соединения между реплицируемыми системами могут отказывать, что вызывает нарушение согласованности данных.

Стратегия резервного копирования баз данных должна разрабатываться совместно с администраторами баз данных. В одних случаях может оказаться приемлемым разрешить администраторам баз данных выполнять процедуры резервного копирования и восстановления вручную. В других случаях эти действия должны будут выполняться при участии системного администратора.

Что такое резервное копирование

Основная идея резервного копирования — создание копий всего, что установлено на вашей системе, с некоторыми исключениями. Основными исключениями, не включаемыми в резервные копии, являются:

- файловая система /proc, т. к. она содержит только данные, которые ядро генерирует во время работы операционной системы, и нет никакого смысла сохранять их;
- файловая система /mnt, поскольку в нее монтируются сменные носители — CD-ROM, дискеты и т. п.;
- сетевые каталоги — смонтированная файловая система NFS, Samba и прочие виды сетевых данных;
- программное обеспечение, которое может быть легко повторно установлено. Здесь надо иметь в виду, что оно может иметь конфигурационные файлы, которые необходимо копировать, чтобы не выполнять работы по их настройке позже.

Носители данных

Тип носителей для резервного копирования сильно зависит от ваших финансовых возможностей и объема сохраняемой информации. Совершенно нелогично покупать дорогую магнитооптику, если объем резервируемой информации не превышает одного-двух мегабайт за неделю. Рассмотрим носители информации и приводы, которые можно использовать в целях резервного копирования.

Дискета

Пожалуй, самое простое решение для небольших резервных копий и самое дешевое — дискеты стоят весьма дешево. Однако с точки зрения надежности — решение не выдерживает никакой критики. У каждого пользователя есть печальный опыт потери информации из-за некачественных дискет, даже если применять дисководы только известных производителей (Panasonic, Sony, Teac) и хорошие дискеты фирм Verbatim, TDK, BASF. Помимо всего прочего — дискеты боятся магнитных полей.

Iomega Zip

Накопители Iomega Zip существуют в нескольких вариантах — емкостью 40, 100 и 250 Мбайт. Достаточно большая скорость обмена, хорошая сохранность данных, однако весьма высокая стоимость самого привода и носителей информации, а также небольшая по современным меркам емкость дисков не способствуют широкому распространению этих устройств.

Iomega Jaz

Накопитель Iomega Jaz — "старший брат" Iomega Zip. Емкость диска — порядка 1 Гбайт, однако большая стоимость самого привода и носителей информации также не способствует широкому распространению этих устройств.

Жесткий диск

Резервное копирование на жесткий диск, установленный в системе, — неплохой бюджетный вариант резервного копирования и достаточно надежный. Вариантов организации резервного копирования на жесткий диск достаточно много — это и использование жесткого диска как хранилища данных, и организация различного уровня RAID-массивов (стопроцентное резервное копирование "на лету" —"зеркалирование" жесткого диска). Однако у этого варианта есть свои недостатки — в случае выхода из строя контроллера жестких дисков существует достаточно большая вероятность, что

он за собой потянет и сами жесткие диски. Или какая-то программа начнет бесконтрольно писать (или стирать) данные. Вариант решения данной проблемы — резервный диск держать размонтированным и монтировать его только на время создания резервной копии. Можно так же использовать резервное копирование по сети на жесткий диск, расположенный на другом компьютере, однако в этом случае могут сказаться сетевые ошибки.

CD-RW

Благодаря дешевизне приводов и носителей информации использование устройств CD-RW для резервного копирования в последнее время становится очень популярным. Действительно, при стоимости устройства от 70 долларов и чистого диска CD-R от 30 центов — пожалуй, мы имеем самый дешевый вариант для хранения резервных копий средних размеров. Достоинством этого способа резервного копирования является дешевизна, большой срок хранения информации (некоторые производители дисков обещают двадцать лет сохранности данных), и основной плюс — доступность считывающих устройств. Недостатком данной технологии является, пожалуй, только одно — объем резервируемых на один диск данных ограничен объемом в 700 Мбайт.

DVD-RW

DVD-RW лишены недостатка CD-RW — их емкость теоретически может доходить до 9 Гбайт, однако эта технология еще не устоялась и не получила широкого распространения. Поэтому эти устройства достаточно дороги — порядка 500 долларов, и для небольшой фирмы затраты на DVD-RW себя не оправдают.

Магнитооптические диски

Магнитооптические диски существуют в разных модификациях, с емкостью носителя от 640 Мбайт и до 4,7 Гбайт, а в скором времени производители обещают еще большие емкости носителей. Наряду с ленточными накопителями (стримерами), считаются основными устройствами для резервирования данных в серьезных проектах. Для магнитооптики существуют специальные библиотеки, благодаря чему можно сохранять терабайты информации. К сожалению, магнитооптические накопители — дорогие устройства, и для небольших фирм так же не по карману.

Стримеры

Пожалуй, одно из старейших устройств резервного копирования. За свою долгую жизнь получило достаточно широкое распространение. Достоинст-

ва — отработанная десятилетиями технология, неплохая надежность и средняя себестоимость хранения информации. Недостатки — привод достаточно дорогой, чехарда с форматами кассет и хранения данных, срок службы лент составляет лишь несколько лет.

На базе ленточных накопителей создают библиотеки и роботизированные системы для хранения огромных объемов данных, однако вам вряд ли доведется столкнуться с такими устройствами.

Тестирование архивов

Обязательным элементом резервного копирования является тестирование полученных архивов. Хранение непроверенной резервной копии создает ложное ощущение защищенности. Исходя из этого, каждая резервная копия, прежде чем помещать ее в хранилище данных, должна быть проверена на целостность.

Обязательно необходимо произвести пробное восстановление системы из резервной копии, во-первых — для проверки процедуры восстановления данных, во-вторых, чтобы убедиться в корректности процедуры резервирования данных.

Еще одна проблема, связанная с восстановлением систем из резервных копий, заключается в установлении права владения файлами. Когда данные извлекаются из файла копии пользователем с правами root, утилита GNU tar предпринимает попытки восстановить существовавшие права владения (пользователя и группы) каждым файлом, но только если при ее вызове был установлен переключатель (`--preserve-permission`). В противном случае утилита tar будет использовать текущие установки UMASK. Однако, если перечисленных в файле tar пользователей и групп не существует, то право владения не сможет быть корректно установлено!

Следовательно, прежде чем выполнять восстановление любых некорневых файлов системы, следует восстановить файлы /etc/passwd и /etc/group.

Даже в том случае, когда компьютер под Linux используется в качестве рабочей станции и имеет единственного пользователя, корректное функционирование многих программ и подсистем будет зависеть от установленных прав владения и разрешений на доступ к системным файлам.

Риск при тестировании архивов

Тестирование процедур восстановления может быть достаточно рискованно, особенно когда вы только отрабатываете процедуру резервного копирования. Возможно случайное разрушение файлов, относящихся к другому приложению, которые использовались в момент проведения попытки восстановления файлов.

Самый безопасный способ выполнить контрольное восстановление системы — провести его на резервной рабочей станции, а не на находящемся в промышленной эксплуатации сервере.

Утилиты резервного копирования

В этом разделе приводятся примеры подготовки и создания полных, выборочных и инкрементных резервных копий с использованием утилит tar, cpio и dump/restore.

Для определенности будем считать, что в нашей системе установлен стример. Аналогичным образом можно использовать и другое оборудование.

Создание резервной копии утилитой tar

Самый простой вариант использования утилиты tar — просто создать архив всех каталогов, начиная с корневого. В этом случае простейшая команда для вызова утилиты tar с целью создания копии будет иметь следующий вид:

```
tar c /
```

Однако при выполнении указанной команды возникнет несколько проблем. Во-первых, по умолчанию утилита может использовать не тот тип ленточного устройства, который установлен на данном компьютере, и даже вообще осуществлять вывод не на магнитную ленту. Во-вторых, в этом примере будет считано все дерево файловой системы. Это значит, что будет обработана файловая система /proc, любые установленные CD-ROM, файловые системы NFS и Samba, а также другие разделяемые сетевые файловые системы.

Приведенный пример может вызвать и несколько других проблем. Например, при обработке подобной команды GNU tar никогда не будет обеспечивать специальной поддержки sparse-файлов (файлов, имеющих реальный размер меньше, чем зарезервировано под них места в файловой системе) и выполнять сжатие выходной информации.

Ниже приведен пример более корректного вызова утилиты.

```
tar cSlzf — $(backdirs) | buffer -o /dev/st0
```

В этом примере создается (`c`) архив с поддержкой sparse-файлов (`S`), ограниченный локальными файловыми системами (`l`). Выполняется сжатие данных (`z`) и их запись в файл (`f`) stdout, в архив включаются только указанные каталоги (`backdirs`).

Созданный архив по каналу передается программе buffer, которая записывает его (`-o`) на первое ленточное устройство с интерфейсом SCSI. Подобный подход следует использовать и при получении резервной копии от программы сжатия или через сеть.

Параметр `bakdirs` — это сценарий, в котором перечисляются каталоги и файлы, включаемые в создаваемую резервную копию. Сценарий `backdirs` может состоять просто из команд `echo`, которые перечисляют все точки входа локальных файловых систем (за исключением каталога /proc, любых каталогов /temp, установленных CD-ROM, каталогов NFS и других сетевых ресурсов). Назначение `bakdirs` состоит в просмотре и фильтрации выходных данных команд `mount`, что позволяет динамически включать в копию только требуемые файловые системы. Неудобство использования обычного статического списка состоит в том, что он не может автоматически обновляться при добавлении новых файловых систем.

Избегайте дублирования ссылок в командной строке, содержащей вызов утилиты tar. Если одновременно будут копироваться каталоги /some/moutpoint и /some/mountpoint/somedir, расположенные в одной и той же файловой системе, утилита tar дважды поместит в архив все содержимое каталога /some/mountpoint/ somedir.

Использование утилиты cpio

Утилита cpio представляет собой еще один традиционный инструмент создания резервных копий и архивирования файловых систем. В сравнении с утилитой tar ее работа организована иначе.

Во многих случаях принимаемый в утилите cpio подход к указанию подлежащих копированию или восстановлению файлов и каталогов является прямо противоположным подходу, применяемому в утилите tar. При создании архива утилите tar передается список файлов и каталогов, указываемых как параметры командной строки. Любой указанный каталог просматривается рекурсивно. При создании архива с помощью утилиты cpio ей предоставляется список объектов (имена файлов и каталогов, символические имена любых устройств, гнезда доменов UNIX, поименованные каналы и т. п.). Этот список помещается в стандартный поток stdin утилиты cpio с помощью канала и обычно генерируется командой `find`.

Простая команда, выполняющая копирование всей файловой системы, выглядит следующим образом:

```
find / -print0 I cpio -o0B > /dev/st0
```

Результаты выполнения команды `find` будут включать каталог /proc и тому подобные нежелательные для резервного копирования каталоги. Уточнив используемые параметры команды `find`, можно исправить ситуацию:

```
find /* -fstype ext2 -print0 I cpio -o0B > /dev/st0
```

В этом примере копируемые объекты ограничены только файловыми системами типа Ext2. Также будут пропущены все скрытые файлы и каталоги.

Устройства вывода информации на магнитную ленту стоят дорого. Получить *доступ к удаленным устройствам* не намного сложнее, чем к локальным:

```
find /* -fstype ext2 -print0 |  ssh $TAPEHOST "cpio -o0B I buffer -o
/dev/st0"
```

Обратите внимание, что для обращения к удаленным ленточным устройствам используется команда `buffer`.

Восстановление с локального ленточного устройства

Еще одно принципиальное различие между утилитами tar и cpio состоит в способе сохранения и восстановления абсолютных путей. В случае с утилитой tar ведущая косая черта в абсолютных именах файлов при создании копии удаляется. Утилита cpio в процессе восстановления принудительно превращает все пути в относительные.

Как правило, файлы должны восстанавливаться в тех каталогах, которые будут задаваться относительно текущего каталога (или — в некоторых случаях — каталога root). По умолчанию утилита cpio не восстанавливает каталогов, поэтому при ее вызове следует указывать параметр `-d`.

Восстановление с удаленного ленточного устройства

Восстановление с удаленных ленточных устройств осуществляется так же просто, как и копирование, например:

```
ssh $OTHERHOST 'buffer -i /devst0', I 'find /* -fstype ext2 -print0 |
    cpio -id'
```

Если необходимо восстановить только некоторые файлы, добавьте в конец команды cpio список глобальных шаблонов.

Здесь обнаруживается еще одно различие между утилитами cpio и tar, связанное с выполнением частичного восстановления. При использовании утилиты tar список требуемых файлов и каталогов можно поместить прямо в команду ее вызова. Однако утилита tar не допускает использования глобальных шаблонов.

При работе с утилитой tar типичный способ обойти это ограничение состоит в том, чтобы извлечь индекс архива в файл путем простого перенаправления вывода. Данные полученного файла фильтруются с помощью команды `grep`, после чего полученный список передается команде вызова утилиты tar для извлечения данных. Например, подготовив файл restorelist, содержащий имена требуемых файлов и каталогов, помещенные в отдельные строки, можно ввести следующую команду:

```
ssh $OTHERHOST 'buffer -i /dev/st0' I 'tar xTf /tmp/restorelist -'
```

Программа резервного копирования dump

Программа dump в корне отличается от tar — она предназначена для резервного копирования и восстановления файловой системы и создает резервные копии элементов файловой системы. Использование этой утилиты позволяет получить копию одной файловой системы быстро и эффективно. К сожалению, ее нельзя применить к отдельным каталогам. Программа restore выполняет функцию, обратную dump, она восстанавливает полную резервную копию файловой системы.

Утилита dump имеет несколько уровней резервного копирования. Уровни могут быть от 0 до 9, где уровень 0 (полная резервная копия системы), который гарантирует, что все элементы файловой системы будут скопированы. Уровни выше 0 — добавочные резервные копии, которые указывают dump копировать все новые или модифицированные после последнего копирования файлы. Чтобы быть более точным, на каждом уровне добавочного резервного копирования вы сохраняете все изменения, произошедшие после создания последней резервной копии на том же или предыдущем уровне. Это позволяет вам осуществлять инкрементное резервирование системы.

Создание резервных копий с помощью программы dump

Использование программы dump очень простое:

```
dump -0u -f /dev/st0 /home
```

Таким образом, мы создали полную копию каталога /home. Для получения инкрементной копии выполним следующую команду:

```
dump -3u -f /dev/st0 /home
```

Восстановление файлов, созданных dump

Для восстановления резервных копий, созданных утилитой dump, используется программа restore. Она восстанавливает файлы и каталоги из резервных копий, полученных программой dump. При диалоговом восстановлении файлов из копии программа restore предоставляет интерфейс, который позволяет пользователю перемещаться по дереву каталогов, выбирая файлы для извлечения, после чтения информации о каталогах из копии.

При восстановлении резервной копии мы должны перейти в раздел файловой системы, где хотим восстанавливать нашу резервную копию. Это требуется, т. к. в диалоговом режиме программа restore восстанавливает все файлы раздела файловой системы, из которой она была запущена.

Для восстановления файлов из копии в диалоговом режиме используйте команду:

```
restore -i -f /dev/st0
restore >
```

На вашем терминале вы увидите командную строку, для получения списка файлов текущего или заданного каталога используйте команду `ls`:

```
restore > ls
admin/ lost+found/ named/ quota.group quota.user wahib/
restore >
```

Чтобы внести текущий каталог или файл в список файлов для извлечения, используйте команду `add`:

```
restore > add Personal/ restore >
```

Чтобы удалить текущий каталог или заданный файл из списка файлов для извлечения, используйте команду `delete`.

Чтобы восстановить все файлы из списка для извлечения, используйте команду `extract`.

Для выхода из интерактивного режима программы restore после завершения восстановления файлов используйте команду `quit`.

Пакет AMANDA

Многие системные администраторы создают свои собственные сценарии и выполняют большую часть работы по контролю за использованием томов вручную.

Пакет AMANDA (Advanced Maryland Automatic Network Disk Archiver) контролирует процесс проведения серий полных и инкрементных резервирований данных на промежуточное дисковое хранилище хоста ленточных устройств для некоторого набора сетевых клиентов. Затем полученные наборы данных переносятся на ленточные носители. При условии корректной установки пакет AMANDA работает полностью автоматически. Процесс резервирования обычно запускается ночью и осуществляет все операции копирования, последовательно устанавливая соединения хоста копирования с каждым из клиентов. Существует возможность устанавливать эти соединения параллельно, а также контролировать и регулировать создаваемую нагрузку на сеть. Модуль планировщика определяет, какой уровень инкрементного копирования должен быть выполнен для каждой файловой системы каждого из хостов.

Программы пакета AMANDA для создания архивных файлов используют утилиты dump или tar. Поэтому оказывается возможным извлекать создан-

ные ими архивы с помощью обычных инструментов. Кроме того, для восстановления предназначена команда `amrecover` пакета AMANDA.

Команды *mt* и *mtx*

Команды `mt` и `mtx` предназначены для управления устройствами вывода на магнитную ленту. Команда `mt` предназначена для получения сведений о состоянии устройства, определения или установки абсолютной и/или логической позиции головки над носителем, перемотки носителя, извлечения носителя и поиска "по направлению вперед" конца последнего блока записанных данных.

Команда `mtx` включает расширения для большинства распространенных типов устройств автоматической смены носителей.

Команда *buffer*

Команда `buffer` осуществляет вывод непрерывной последовательности данных на носитель даже в тех случаях, когда предоставляющие эти данные команды выводят их неравномерно. Данная ситуация является типичной для работы программ сжатия данных, считывающих информацию через сетевые соединения. Подобное возможно даже при выполнении нормальных операций архивирования в среде перегруженной многозадачной операционной системы.

Многотомные резервные копии

Приведенные ранее примеры не предназначены для создания резервных копий, занимающих несколько магнитных лент. Утилиты tar и cpio предоставляют параметры, позволяющие разместить любой архивный файл на нескольких томах носителей. Размещение архива на нескольких лентах снижает его надежность, поскольку порча любой из лент делает хранение всего остального набора носителей бесполезным. Тем не менее иногда это необходимо, а при соответствующей адаптации подхода и полезно, например для резервирования на компакт-дисках.

Ссылки

- **www.veter.sky.net.ua/docs/linux/LINUXSOS/index.html** — Gerhard Mourani. Безопасность и оптимизация Linux. Редакция для Red Hat.
- **www.amanda.ocg** — сайт программы AMANDA.
- Соответствующие страницы man.

Глава 34

X Window и другие графические оболочки

Операционная система Linux давно уже немыслима без графической оболочки X Window, по крайней мере, на рабочих местах пользователей, поэтому необходимо иметь хотя бы общее представление о ее конфигурировании. В принципе, в большинстве современных дистрибутивов во время инсталляции система корректно распознает вашу аппаратуру и настраивает X Window, однако всегда существуют некоторые аспекты конфигурирования, которые хотелось бы подправить.

Конфигурирование X Window

Конфигурирование X Window включает в себя четыре основных компонента:

- конфигурирование X-сервера;
- конфигурирование диспетчеров окон Window Manager;
- конфигурирование прикладных программ;
- русификацию.

Эти действия могут быть сделаны как администратором — для всей системы сразу, так и пользователем, но только для себя. Исключениями являются лишь X-сервер, конфигурацию которого может модифицировать только root, и, частично, русификация.

Конфигурирование X-сервера

Базой системы X Window является X-сервер, выполняющий основную работу системы. Все настройки X-сервера располагаются в файле /etc/X11/XF86Config.

Этот файл состоит из нескольких секций, каждая из которых содержит настройки для определенной подсистемы — шрифтов, мыши, клавиатуры, монитора, видеоадаптера.

Общий вид секции такой:

```
Section "имя-секции"
  данные
  ...
EndSection
```

Внутри секций могут быть подсекции — они определяются парой ключевых слов `SubSection/EndSubsection`.

В табл. 34.1 приведены основные секции конфигурационного файла XF86Config.

Таблица 34.1. *Основные секции файла XF86Config*

Секция	Описание
`Files`	Секция содержит пути к используемым файлам — в основном, это каталоги со шрифтами либо используемый сокет
`InputDevice`	Секция содержит описание устройств ввода — клавиатуры, мыши
`Monitor`	Секция содержит описание монитора
`Device`	Секция содержит описание видеокарты
`Screen`	Секция содержит описание экрана — разрешение и глубина цвета
`ServerLayout`	Секция содержит описание используемых в настоящий момент секций, описывающих нашу конфигурацию
`Module`	Секция содержит описание загружаемых сервером модулей

Ниже приведен пример конфигурационного файла XF86Config-4.

```
# File generated by anaconda.

Section "ServerLayout"
        Identifier     "Anaconda Configured"
        Screen      0  "Screen0" 0 0
        InputDevice    "Mouse0" "CorePointer"
        InputDevice    "Keyboard0" "CoreKeyboard"
EndSection

Section "Files"

# The location of the RGB database.  Note, this is the name of the
# file minus the extension (like ".txt" or ".db").  There is normally
# no need to change the default.
```

```
    RgbPath   "/usr/X11R6/lib/X11/rgb"

# Multiple FontPath entries are allowed (they are concatenated together)
# By default, Red Hat 6.0 and later now use a font server independent of
# the X server to render fonts.

    FontPath   "unix/:7100"

EndSection

Section "Module"
        Load  "GLcore"
        Load  "dbe"
        Load  "extmod"
        Load "fbdevhw"
        Load "pex5"
        Load "dri"
        Load  "glx"
        Load  "pex5"
        Load  "record"
        Load  "xie"
EndSection

Section "InputDevice"
        Identifier  "Keyboard0"
        Driver      "keyboard"

#       Option "AutoRepeat"  "500 5"

# when using XQUEUE, comment out the above line, and uncomment the
# following line
#       Option    "Protocol"  "Xqueue"

# Specify which keyboard LEDs can be user-controlled (eg, with xset(1))
#    Option   "Xleds"        "1 2 3"

# To disable the XKEYBOARD extension, uncomment XkbDisable.
#    Option    "XkbDisable"
```

```
# To customise the XKB settings to suit your keyboard, modify the
# lines below (which are the defaults).  For example, for a non-U.S.
# keyboard, you will probably want to use:
#    Option    "XkbModel"    "pc102"
# If you have a US Microsoft Natural keyboard, you can use:
#    Option    "XkbModel"    "microsoft"
#
# Then to change the language, change the Layout setting.
# For example, a german layout can be obtained with:
#    Option    "XkbLayout"    "de"
# or:
#    Option    "XkbLayout"    "de"
#    Option    "XkbVariant"   "nodeadkeys"
#
# If you'd like to switch the positions of your capslock and
# control keys, use:
#    Option    "XkbOptions"     "ctrl:nocaps"
     Option    "XkbRules"       "xfree86"
     Option    "XkbModel"       "pc102"
     Option    "XkbLayout"      "ru(winkeys)"
     #Option    "XkbVariant"    ""
     Option    "XkbOptions"     "grp:ctrl_shift_toggle"
EndSection

Section "InputDevice"
    Identifier    "Mouse0"
    # Modified by mouseconfig
    Driver        "mouse"
    Option        "Device"             "/dev/mouse"
    Option        "Protocol"           "IMPS/2"
    Option        "Emulate3Buttons"    "no"
    Option        "ZAxisMapping"       "4 5"
EndSection

Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
        HorizSync    30 — 96
```

```
        VertRefresh  50 — 160
        Option "dpms"

        # -- 1400x1050 --
        # 1400x1050 @ 60Hz, 65.8 kHz hsync
        Modeline "1400x1050"  129   1400 1464 1656 1960
                              1050 1051 1054 1100 +HSync +VSync

        # 1400x1050 @ 70Hz, 76.8 kHz hsync
        Modeline "1400x1050"  151   1400 1464 1656 1960
                              1050 1051 1054 1100 +HSync +VSync

        # 1400x1050 @ 75Hz, 82.3 kHz hsync
        Modeline "1400x1050"  162   1400 1464 1656 1960
                              1050 1051 1054 1100 +HSync +VSync

        # 1400x1050 @ 85Hz, 93.2 kHz hsync
        Modeline "1400x1050"  184   1400 1464 1656 1960
                              1050 1051 1054 1100 +HSync +VSync

EndSection

Section "Device"
    # no known options
    Identifier   "NVIDIA GeForce 2 MX (generic)"
        Driver       "nv"
        VendorName   "NVIDIA GeForce 2 MX (generic)"
        BoardName     "NVIDIA GeForce 2 MX (generic)"

        #BusID
EndSection

Section "Screen"
    Identifier   "Screen0"
        Device       "NVIDIA GeForce 2 MX (generic)"
        Monitor      "Monitor0"
    DefaultDepth    16

    Subsection "Display"
            Depth       16
```

```
                Modes        "1024x768"
    EndSubsection

EndSection

Section "DRI"
    Mode 0666
EndSection
```

Секция *Files*

В этой секции задается местоположение файла со списком цветов и содержится список каталогов, в которых X-сервер должен искать шрифты.

Порядок директив, задающих пути к шрифтам, имеет значение — при подборе шрифтов по псевдонимам они будут искаться в указанном порядке. Таким образом, если поставить каталог со шрифтами koi8-r в начало списка, то во многих случаях вместо европейских будут использоваться кириллические шрифты.

Секция *Keyboard*

В этом разделе определяются параметры и поведение клавиатуры — в частности, переключатель раскладок клавиатуры.

Секция *Pointer*

В этой секции задаются параметры мыши — тип устройства, эмуляция третьей кнопки, количество кнопок и т. п.

Секция *Monitor*

Здесь указываются тип монитора и его параметры. Тип — это название, которое может быть произвольным, и на которое ссылается секция `Screen`.

Указываемые параметры — диапазоны частот горизонтальной (`HorizSync`) и вертикальной (`VertRefresh`) развертки, а также список поддерживаемых видеорежимов (директивы `ModeLine`). При ручной настройке параметров монитора настоятельно рекомендуется ознакомиться с руководством на монитор, в котором имеется информация о максимальных разрешениях монитора, кадровой и строчной развертке.

Имеющиеся в стандартном файле директивы `ModeLine` покрывают большую часть мониторов. При сканировании этих директив X-сервер выбирает из видеорежимов с одинаковым именем тот, который дает наибольшую кадровую частоту, при этом учитываются только те видеорежимы, которые совместимы с параметрами монитора и видеокарты.

Секций `Monitor` может быть несколько — используется та из них, которая указана в секции `Screen`.

Секция *Device*

В этой секции указываются параметры видеокарты. Обычно все параметры X-сервер определяет сам, считывая их непосредственно из видеокарты.

В случае же, когда требуется изменить какие-либо настройки, следует посмотреть man-страницу по используемому X-серверу.

Секция *Screen*

Здесь указывается, какую конфигурацию видеокарты и какой монитор следует использовать, а также параметры видеорежимов — разрешение и глубина цвета.

Настройка параметров монитора

Впрочем, вам, скорее всего, не понадобится ручное вмешательство в настройки X Window. Для настройки монитора, видеокарты, мыши, клавиатуры можно воспользоваться утилитами конфигурирования, например утилитой xf86config — простой консольной утилитой для конфигурирования X Window. Неудобна она тем, что если при выборе параметра вы ошиблись, отменить выбор невозможно. Программа Xconfigurator также представляет собой консольное приложение, но, в отличие от предыдущей утилиты, имеет более удобный интерфейс (рис. 34.1).

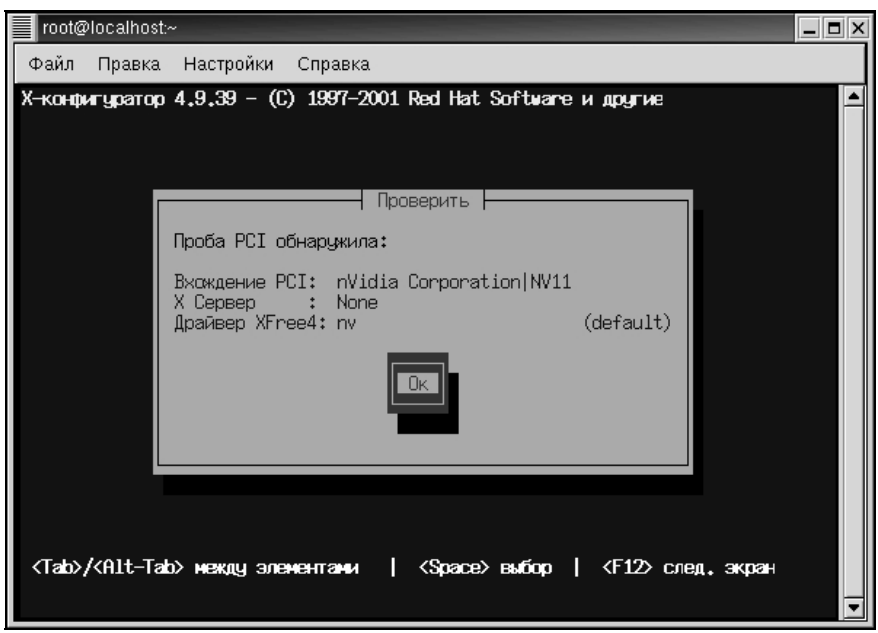

Рис. 34.1. Программа Xconfigurator

Однако с помощью утилиты xf86config каждый режим работы монитора может описываться самостоятельно, что позволяет выжать из вашего монитора все, на что он способен. Для этого необходимо в файле xf86config задать для нужного видеорежима строку в таком формате:

```
Modeline  "mode_name"  D  H1 H2 H3 H4  V1 V2 V3 V4  Flags
```

где:

- `Modeline` — ключевое слово, определяющее строку, содержащую описание видеорежима;
- `"mode_name"` — название нашего видеорежима. Написать можно что угодно, традиционно имя записывается в виде "разрешение_по_горизонтали × разрешение_по_вертикали", например "1024×768". `"mode_name"` используется в качестве ссылки на имя режима в `Section "Screen"`, `Subsection "Display"`, `Modes "mode_name"`. Режимы устанавливаются в порядке перечисления;
- `D` — частота тактового генератора. Это число задается в мегагерцах;
- `H1`, `H2`, `H3`, `H4` — числа, отвечающие за строчную синхронизацию;
- `V1`, `V2`, `V3`, `V4` — числа, отвечающие за кадровую синхронизацию;
- `Flags` — параметры для тонкой подстройки синхронизации.

Давайте разбираться с этими параметрами. У нас есть тактовая частота генератора, к которому привязываются все параметры видеокадра. Видеокадр состоит из видеострок. Строка имеет следующие параметры:

- `A` — количество пикселов в строке (временной интервал, затрачиваемый на вывод строки пикселов);
- `B` — время между окончанием вывода строки и появлением строчного синхроимпульса;
- `C` — время, за которое выводится синхроимпульс;
- `D` — время обратного хода развертки.

Таким образом, для строчной развертки получаем:

```
H1 = A
H2 = A+B
H3 = A+B+C
H4 = A+B+C+D
```

Для кадровой развертки в качестве единицы измерения используется частота строк. Поэтому:

`V1` — количество строк, отображаемых в одном кадре;

`V2` — количество строк от начала кадра до начала кадрового синхроимпульса;

V3 — количество строк от начала кадра до конца кадрового синхроимпульса;

V4 — общее количество строк в кадре.

На современном оборудовании при инсталляции операционной системы частоты монитора программа инсталляции выставляет по максимуму, поэтому ручное вмешательство в настройки монитора вам, скорее всего, не понадобится.

Последовательность запуска X Window

Давайте для лучшего понимания функционирования системы X Window рассмотрим процесс ее запуска.

Стандартный процесс запуска состоит из 5—6 уровней:

1. Запуск пользователем программы startx.
2. Запуск программой startx программы xinit.
3. Запуск X Window и обработка файлов /etc/X11/xinit/xinitrc или ~/.xinitrc.
4. Обработка файлов /etc/X11/xinit/Xclients или ~/.Xclients.
5. Запуск разных программ.
6. Запуск Window Manager.

Бóльшая часть из вышеперечисленного — скрипты, и при необходимости можно внести коррективы в процесс запуска.

Конфигурация Window Manager

Файлы конфигурации диспетчеров окон (Window Manager) располагаются в каталоге /etc/X11/ и находятся в подкаталоге, совпадающем с названием диспетчера окон. Все диспетчеры окон используют разный синтаксис в файлах конфигурации, так что наилучший вариант настройки — почитать документацию и посмотреть примеры файлов.

Большинство современных диспетчеров окон имеют программу конфигурации, с помощью которой можно полностью их сконфигурировать.

В современных дистрибутивах отказываются от непосредственного использования диспетчеров окон. В качестве альтернативы предлагается использование графических сред KDE или GNOME.

Графическая интегрированная среда

По большому счету — это операционная среда над операционной средой (кажется парадоксом), организующая единый стилистический интерфейс для приложений, написанных для графической среды, предоставляющая

приложениям стандартизированные методы взаимодействия процессов и стандартные библиотеки. Приложения, входящие в графическую среду, отлично друг с другом взаимодействуют и представляют практически законченный интерфейс для офисного и домашнего использования. Сегодня наибольшее распространение получили две таких интегрированных среды — KDE и GNOME. И кстати, графическая среда может использовать различные оконные менеджеры (по крайней мере, GNOME).

Достоинство этого решения — набор программного обеспечения и стандарты взаимодействия и интерфейса. Недостаток — некоторая тяжеловесность, не позволяющая комфортно работать на слабых компьютерах. Хотя более или менее современный компьютер — начиная от Pentium 200 с 64 Мбайт оперативной памяти — позволяет вполне нормально работать в графической интегрированной среде. А те, у кого слабый компьютер — могут установить оконный менеджер попроще — например twm.

Графическая среда GNOME

GNOME (GNU Network Object Model Environment, Среда GNU, основанная на модели сетевых объектов) базируется на библиотеке GTK+ и реализована для разных платформ, что позволяет запускать ее в операционных средах Linux, BSD и Solaris. Система очень гибкая, использует внешний менеджер окон, в качестве которых можно применять наиболее распространенные оконные менеджеры.

Взаимодействие различных приложений GNOME друг с другом осуществляется с помощью CORBA (Common Object Request Broker Architecture), что дает возможность взаимодействия приложений независимо от того, на каком языке они были написаны, или от того, на каком компьютере они работают.

Настройка GNOME (и не только) полностью осуществляется из нее самой. На рис. 34.2 представлено стартовое окно GNOME сразу после инсталляции операционной системы.

С этой панели мы можем произвести конфигурацию серверов, в частности Apache, произвести настройки собственно GNOME, настроить операционную систему в целом — загружаемые серверы, оборудование компьютера, пользователей и многое другое. До этих же приложений для конфигурирования системы можно добраться и через меню GNOME. После выбора пункта **Настройки** запускается Центр управления GNOME (рис. 34.3).

Или, если вызывать Центр управления GNOME из меню, — получим его в другом варианте (рис. 34.4).

Как видите, можно пойти разными путями, а в итоге прийти к одному результату — хотите — стиль "Панель управления Windows", хотите — "все в одном" с древовидным меню, выбирайте сами.

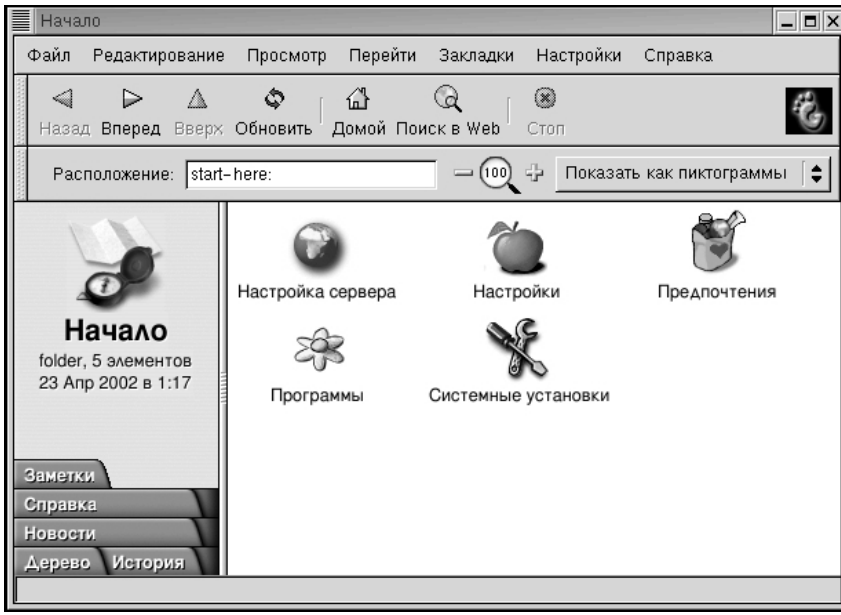

Рис. 34.2. Начало конфигурации GNOME

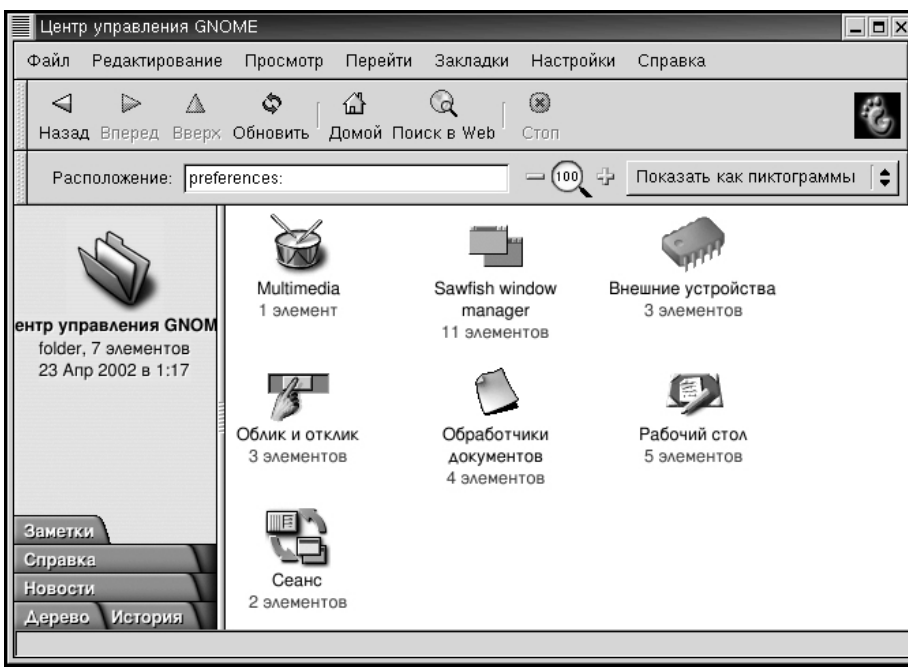

Рис. 34.3. Центр управления GNOME

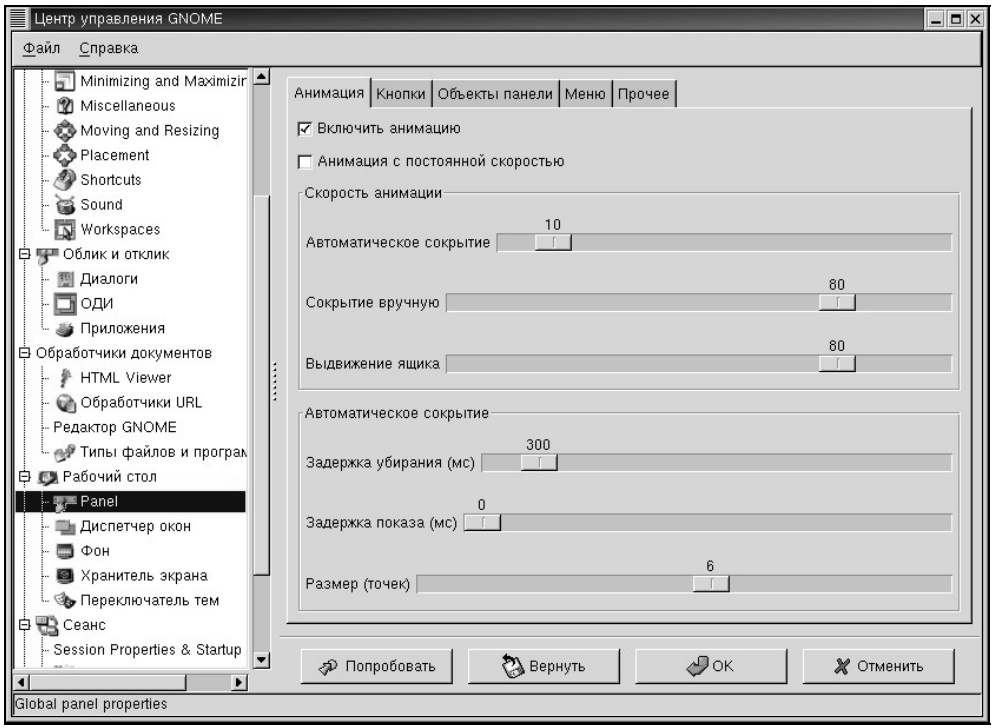

Рис. 34.4. Центр управления GNOME (вариант)

Что у нас здесь есть? Настройка мультимедиа — звуки системы, настройка менеджера окон Sawfish, настройка клавиатуры, CD-ROM, мыши, настройка приложений, обработчиков документов, настройка параметров рабочего стола и наконец — конфигурирование сеанса системы.

Начнем с мультимедиа. На рис. 34.5 изображено окно конфигурирования звуков системы.

Все прозрачно — список событий системы, соответствующий звуковой файл, возможность прослушать выбираемые звуки.

Следующий пункт — конфигурирование менеджера окон, который в данном случае — Sawfish (рис. 34.6).

Этот пункт может отличаться в зависимости от менеджера окон. В нашем случае мы можем настроить внешний вид окон в целом, указатели мыши, поведение окон, минимизацию и максимизацию окон, перемещение их, "горячие" клавиши, звуки и виртуальные рабочие столы. Выбор большой, в каждом пункте много дополнительных параметров. Настройки устройств очень прозрачны — скорость перемещения мыши и нажатия ее кнопок, приблизительно то же касается и клавиатуры.

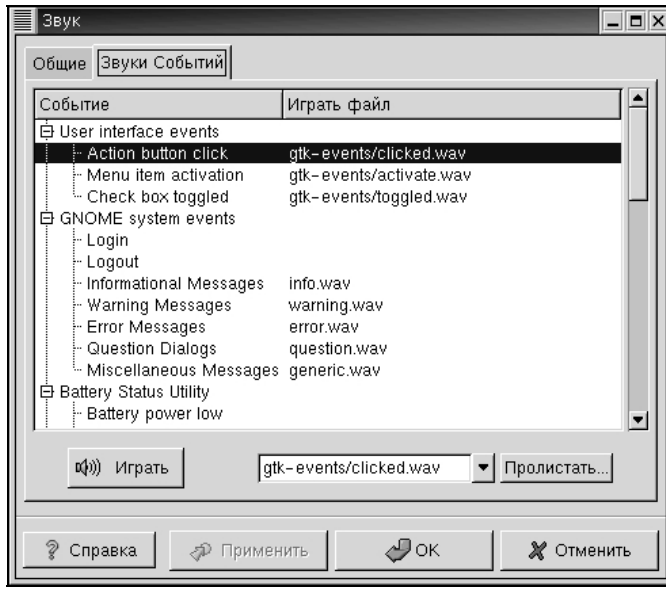

Рис. 34.5. Настройка звуков событий GNOME

Рис. 34.6. Конфигурирование оконного менеджера

Пункт **Облик и отклик** предназначен для определения внешнего вида и поведения приложений. В качестве примера позвольте привести иллюстрацию конфигурирования внешнего вида приложений (рис. 34.7).

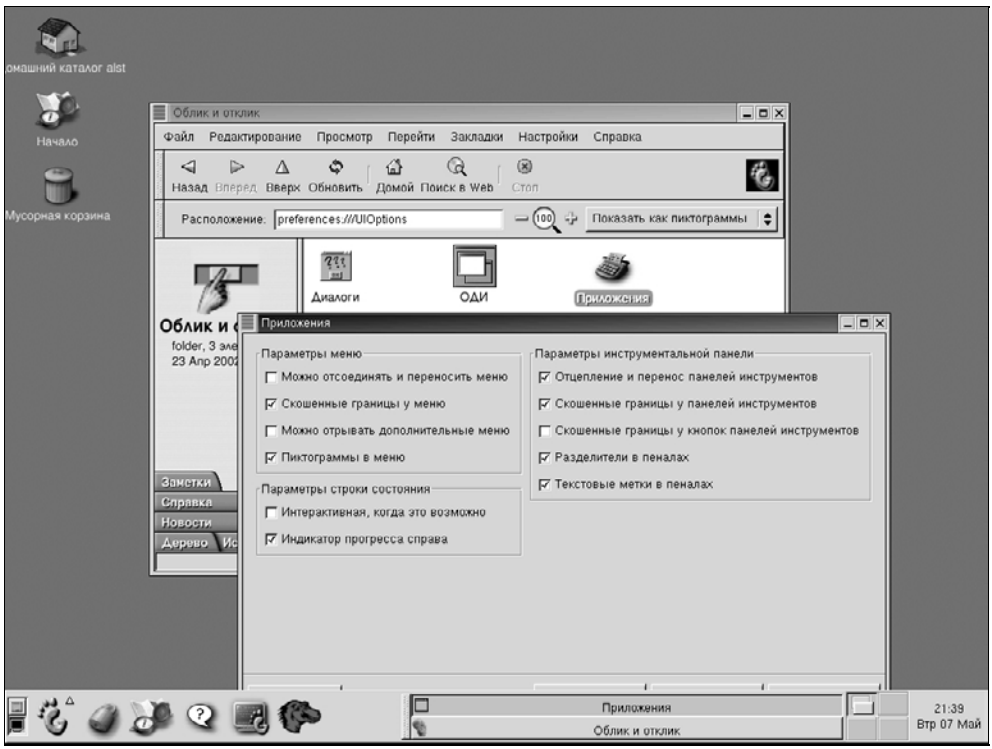

Рис. 34.7. Конфигурирование внешнего вида приложений

Затем можно перейти к пункту **Обработчики документов**. В этом пункте мы должны определить ассоциации приложений с типами документов (рис. 34.8).

Здесь мы задаем приложение для просмотра файлов HTML-формата, стандартный для GNOME текстовый редактор, обработчики URL-адресов и, наконец, — общий список типов файлов и приложения, которые вызываются для просмотра или обработки этих файлов. Очень похоже на Windows, только сделано более удобно.

Настройка рабочего стола. Все понятно из рис. 34.9.

Настройка сеансов позволяет вам запускать свои программы при старте GNOME (рис. 34.10).

Это используется для программ, которые не способны поддерживать управление сеансами.

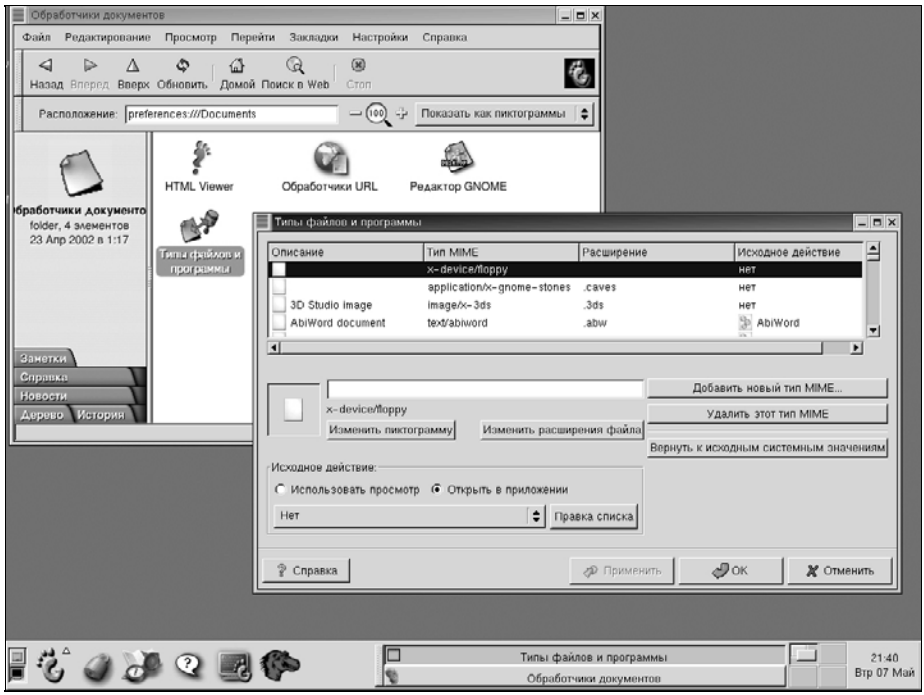

Рис. 34.8. Конфигурирование ассоциации приложений с типами документов

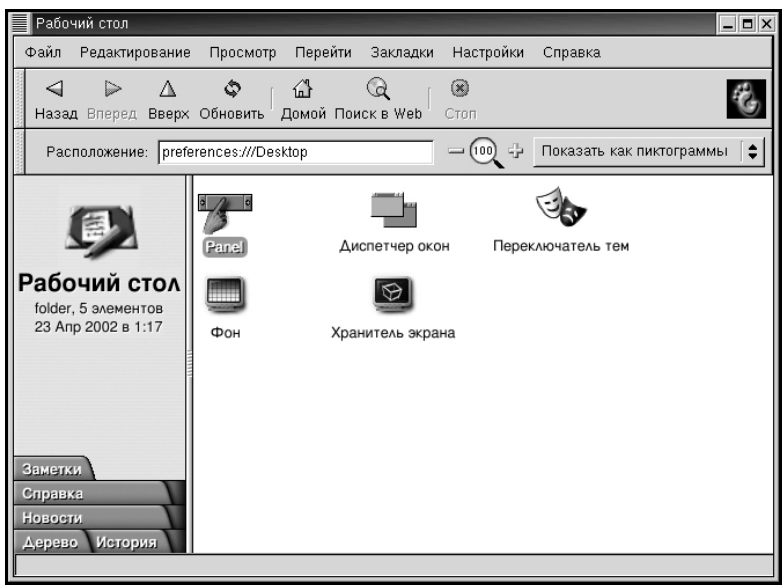

Рис. 34.9. Настройка рабочего стола

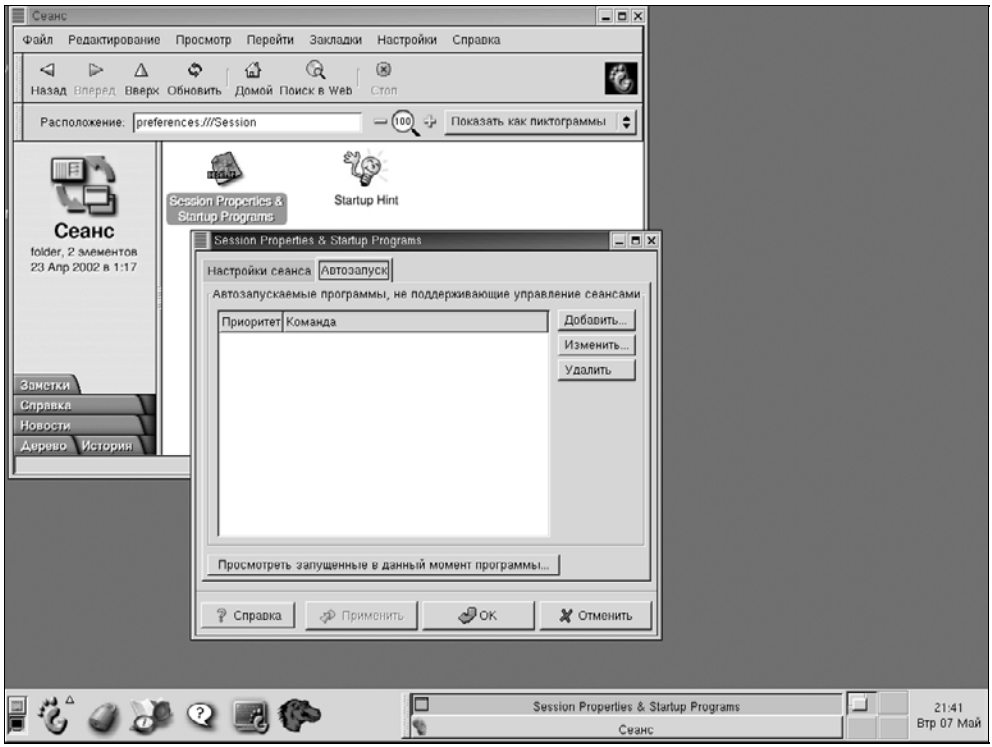

Рис. 34.10. Настройка сеансов GNOME

Вот вкратце и все, что касается конфигурирования собственно GNOME. А теперь перейдем к более административной части — пункту **Системные установки** (рис. 34.11).

Что мы здесь видим? Установки даты и времени, просмотр списка оборудования компьютера, мастер для конфигурирования интернет-соединения, приложение Lokkit, конфигурация сетевых параметров, конфигурирование принтеров, управление сервисами операционной системы и система для управления пользователями. Кратко рассмотрим некоторые утилиты конфигурирования.

Утилита просмотра списка оборудования представлена на рис. 34.12.

С помощью этой утилиты мы можем просмотреть список оборудования, установленного в компьютере, а также решить проблемы некорректной конфигурации драйверов, просмотреть структуру жестких дисков.

Еще одно интересное приложение — Lokkit. Эта утилита поможет обычным пользователям настроить на своем компьютере вполне функциональный брандмауэр, не вникая в тонкости программы iptable или конфигурирования сети и ядра операционной системы. Одно из окон этой утилиты представлено на рис. 34.13.

Рис. 34.11. Системные установки в GNOME

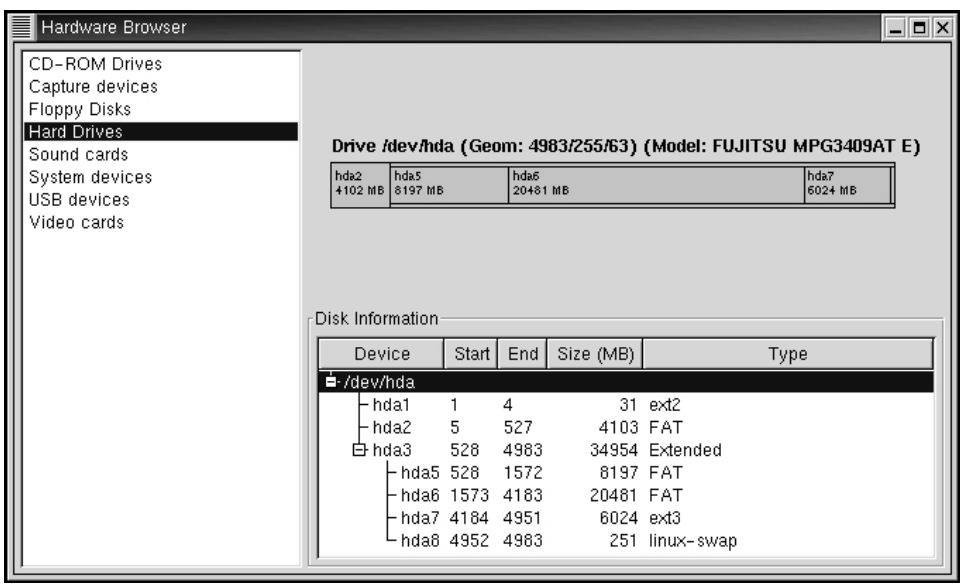

Рис. 34.12. Утилита просмотра списка оборудования

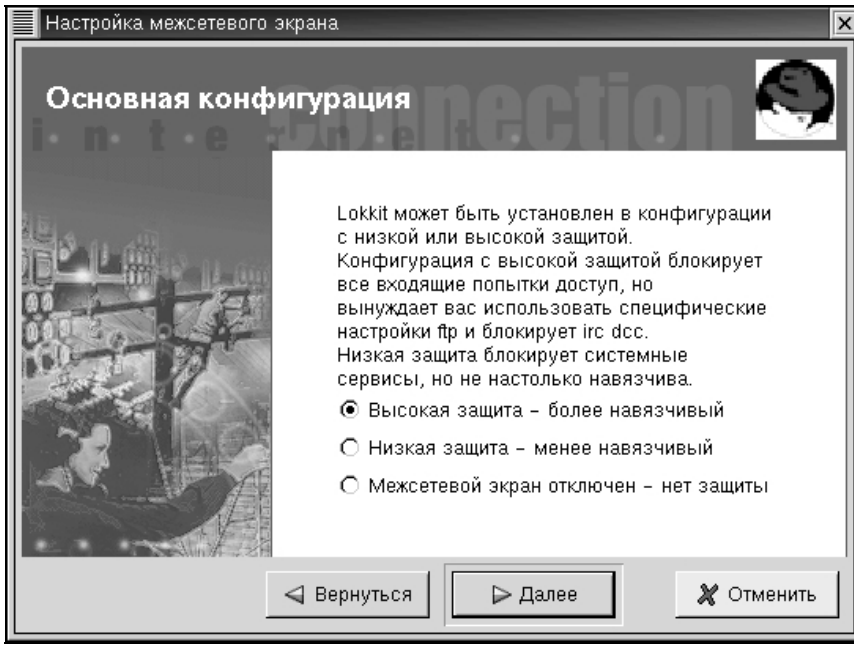

Рис. 34.13. Утилита Lokkit

Утилита Lokkit шаг за шагом проведет неопытного пользователя по процессу настройки сетевого экрана, задавая очень простые вопросы. Конечно, она не позволит вам филигранно его настроить, но быстро создать на рабочем компьютере достаточно надежный сетевой экран вполне способна.

Утилита для настройки служб вашего компьютера изображена на рис. 34.14.

Эта утилита избавляет пользователя от необходимости копаться в каталогах rc. Конечно, это не сложно, но разрешить или запретить запуск какого-либо сервиса на нужном уровне запуска можно намного быстрее, используя эту утилиту. Еще одна приятная ее особенность — она позволяет увидеть краткую характеристику сервиса, что сильно облегчит жизнь обычному пользователю.

И, наконец — менеджер пользователей (рис. 34.15).

Позволяет управлять пользователями и группами пользователей операционной системы — добавить или удалить, поменять описание пользователя или группу, к которой он принадлежит.

Как видите — в GNOME есть все, надо лишь только зайти в соответствующий пункт меню.

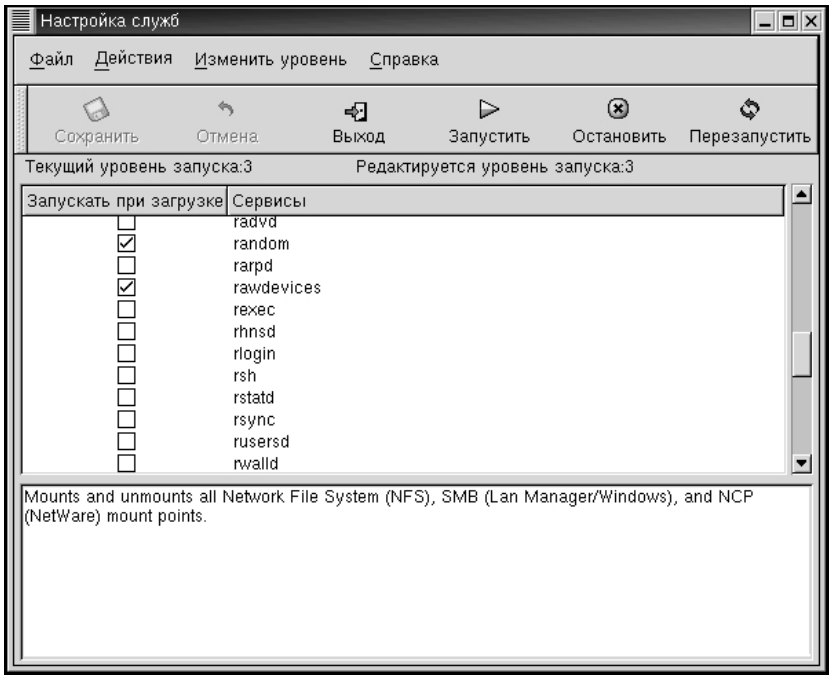

Рис. 34.14. Утилита для настройки служб операционной системы

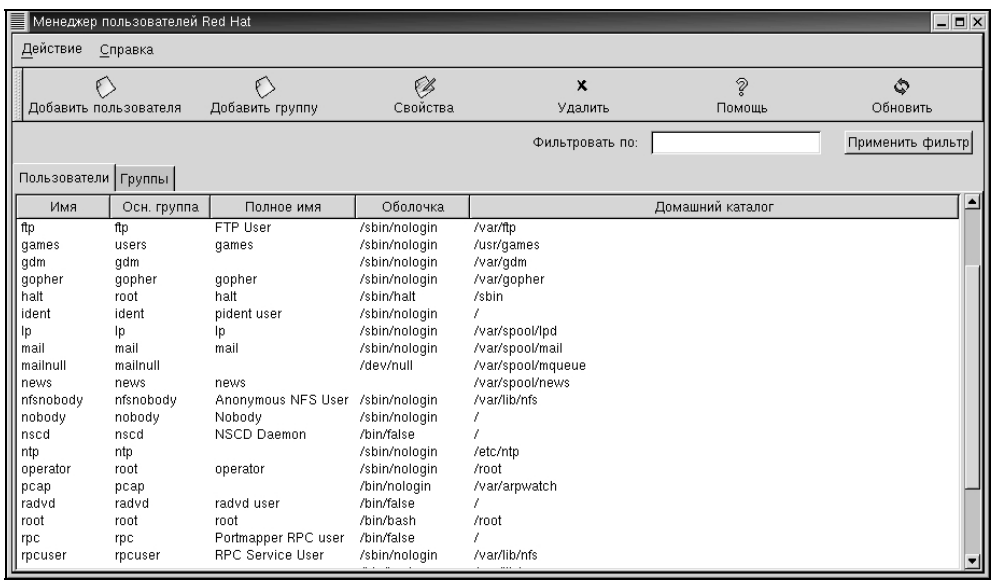

Рис. 34.15. Менеджер пользователей операционной системы

KDE — K Desktop Environment

Еще одна графическая интегрированная среда. По своим возможностям очень напоминает GNOME. Использует библиотеку Qt. Идеология KDE несколько отличается от GNOME — в ней невозможно применять независимые менеджеры окон, у нее несколько более строгий интерфейс. В отличие от GNOME, значительно централизовано управление разработкой как KDE, так и программ для нее. Одна из основных задач, которые ставили разработчики, — сделать систему, напоминающую Windows, но значительно лучше. Более эргономична, более привычна пользователям Windows. Благодаря координированной разработке дизайн практически всех программ решен в едином стиле, а из-за своей популярности KDE имеет большое количество программного обеспечения, написанного специально для нее. Как обычно, если система в чем-то хороша, у нее должны быть и недостатки. Недостатков, по большому счету, два:

- требовательность к ресурсам;
- утечки памяти.

К сожалению, разработчики настолько увлеклись интенсификацией разработки KDE и сопутствующих программ, что оптимизацию кода отложили "на потом". Как результат — неоптимальное использование ресурсов операционной системы — тяжеловесность и требовательность к оперативной памяти и процессору. Небрежность программирования приводит к "забывчивости" в освобождении занятой приложениями оперативной памяти, вследствие чего теряется контроль за ее рациональным использованием. Правда, в последнее время разработчики постепенно исправляют эту ситуацию, однако слава неповоротливой потребительницы ресурсов будет еще долго преследовать KDE.

Перейдем к конфигурированию среды. По существу конфигурирование KDE мало чем отличается от конфигурирования GNOME. Тот же центр управления, то же разделение на средства конфигурирования KDE и средства конфигурирования операционной системы. Правда, GNOME со своим набором утилит выглядит несколько аскетично на фоне KDE, но, в принципе, общий набор функций конфигурирования у этих систем приблизительно одинаков.

Как видите, несколько другой набор функций, но все очень похоже на GNOME. Исходя из похожести интерфейсов KDE и GNOME, мы не будем останавливаться на конфигурировании KDE, позволим себе обратить внимание лишь на редактор меню и менеджер тем. На рис. 34.16 изображен редактор меню KDE. Очень прост, функционален и интуитивно понятен. В левой части окна дерево меню KDE. В правой части окна — параметры пункта меню, такие как имя пункта меню, комментарий к пункту меню, команда, которая выполняется по нажатии на этот пункт меню, рабочий каталог программы.

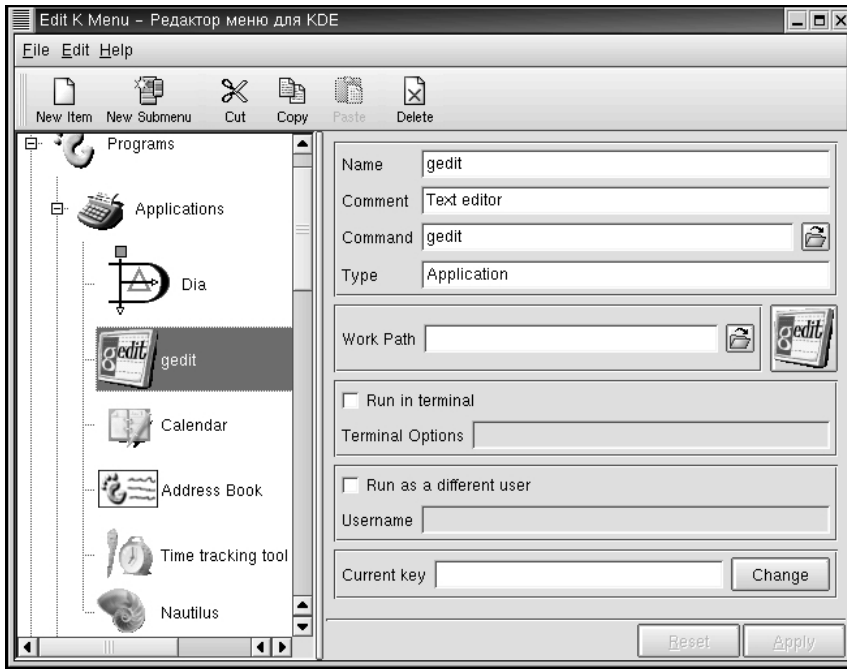

Рис. 34.16. Редактор меню KDE

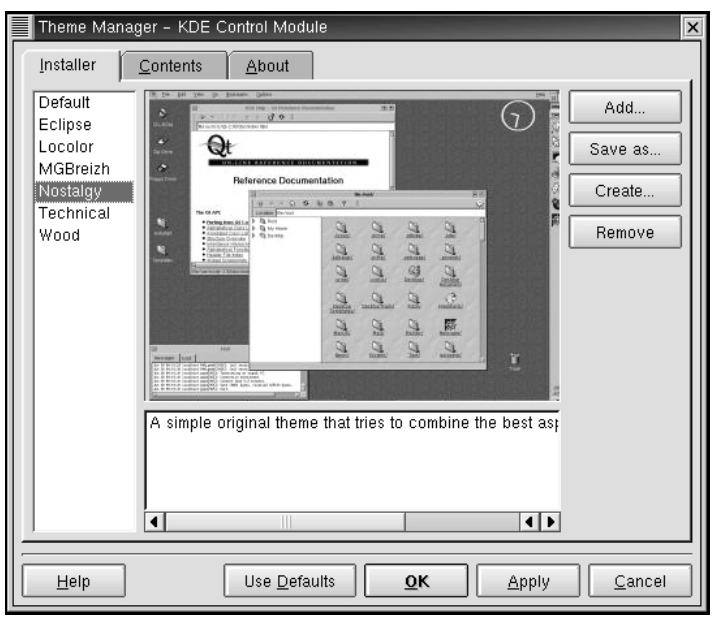

Рис. 34.17. Менеджер тем KDE

Можно выполнять программу в терминале и, что более интересно, — от другого пользователя (пароль перед запуском все равно попросят ввести).

Внешний вид менеджера тем приведен на рис. 34.17. Очень напоминает выбор тем от Microsoft Windows. Разнообразные темы можно скачать с сайта KDE или поискать в Интернете.

Конфигурирование программ — русификация

Конфигурирование программ, написанных для X Window, описать в принципе невозможно, поскольку настройки сильно зависят от типа приложения. Тем не менее, остановимся на одной небольшой особенности конфигурирования прикладных программ множества дистрибутивов операционной системы Linux, выпускаемых за границей. Это, конечно, проблемы русификации. И если с переводом интерфейса программ мы ничего сделать не можем (разве что взять и перевести все пункты меню и сообщения программы самостоятельно), то научить программу правильно воспринимать или хотя бы выводить кириллицу нам в большинстве случаев удается.

Большинство современных программ для KDE или GNOME разрабатываются с учетом требований дальнейшей их локализации, поэтому доведение до соответствующих кондиций нужных нам программ не вызывает особых сложностей. Рассмотрим это на примере текстового редактора gedit (рис. 34.18).

Если вы внимательно посмотрите на изображение, то увидите, что строка текста в gedit сначала набрана латинскими символами, а потом идут буквы с умляутами. Все очень просто — сначала эта строка набиралась в английской раскладке, а потом переключились на русскую. Сами видите, что получилось. Причина очень проста — неверно указана кодировка используемого в редакторе шрифта. Чтобы исправить это досадное недоразумение, необходимо зайти в пункт меню **Настройки**, в появившемся окне выбрать вкладку **Шрифты/Цвета** и щелкнуть на списке шрифтов. В результате откроется окно **Выбрать шрифт**, где можно выбрать используемый фильтр для перекодировки, получить информацию о шрифте, поменять собственно шрифт и его параметры. В нашем случае необходимо в списке **Стиль шрифта** найти кодировку шрифта, используемую вами в системе. Обычно это koi8-r (для пользователей UNIX/Linux) или cp1251 (для приверженцев кодировки Microsoft). В том случае, если в шрифте, используемом по умолчанию, отсутствует нужная вам кодировка, — выберите другой, в котором она есть. Установите стиль шрифта и размер символа в пунктах и сохраните настройки. После этого приложение будет корректно отображать кириллические символы.

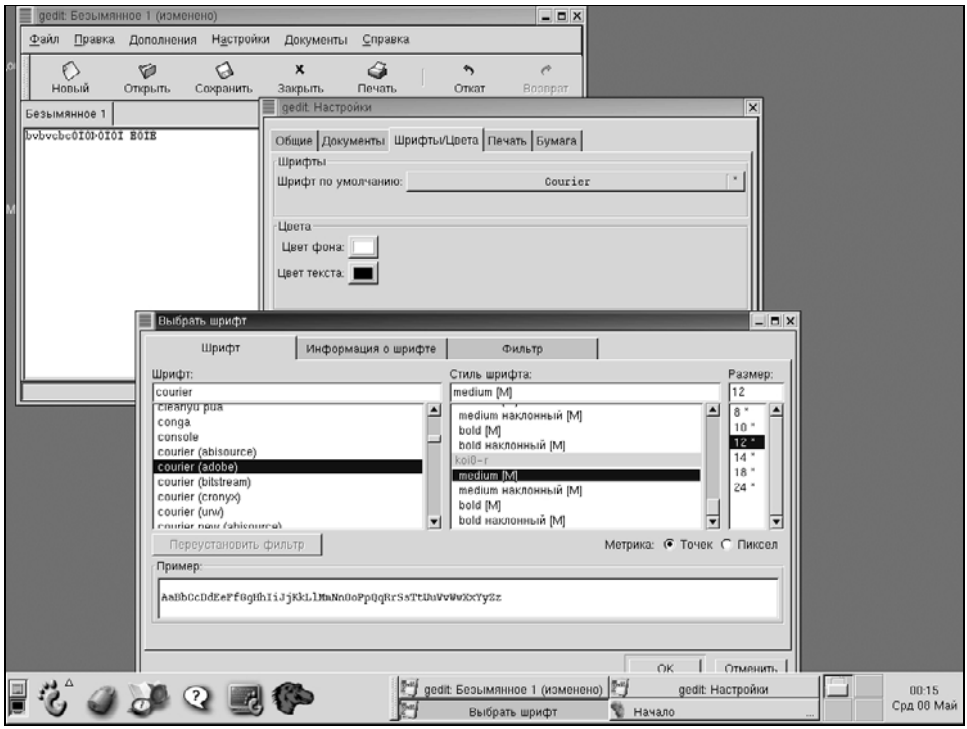

Рис. 34.18. Учим gedit разговаривать по-русски

Ссылки

- **knot.pu.ru/faq/xfaq** — XFAQ по настройке X Window.
- **www.linux.org.ru/books/gnome-ug/ug/** — руководство пользователя GNOME.
- **www.linux.org.ru/books/kde/general/userguide/index.html** — K Desktop Environment. Руководство пользователя.
- **sky.inp.nsk.su/~bolkhov/teach/inpunix/xsetup_simple.ru.html** — Дмитрий Болховитянов. Настройка X Window.
- **gazette.linux.ru.net/lg67/articles/rus-adam.html** — Thomas Adam. Колонка The Weekend Mechanic: настройка X Window. Перевод Владимира Меренкова.
- **gazette.linux.ru.net/lg64/articles/rus-sipos.html** — настройка режима монитора в XFree86.
- **linux.net.kg/articles/x.html** — настройка X Window.
- **www.gnome.org** — официальный сайт проекта GNOME.
- **www.kde.org** — официальный сайт проекта KDE.

Глава 35

Печать

Ну вот — система настроена, и кириллические шрифты есть, и X Window работает с заданными частотами и разрешениями, по Интернету путешествуем с приемлемой скоростью, документы разные находим, просматриваем. А печати у нас и нет, не настроена.

В данной главе мы постараемся исправить это упущение. Принтеры бывают разные — матричные, струйные, лазерные и сублимационные, цветные лазерные и даже объемные. Они могут использовать разные интерфейсы для подключения — последовательный, параллельный, USB и даже Ethernet. Производители принтеров продолжают увеличивать набор проблем — то протокол свой придумают, то с целью удешевления создадут Win-принтер, для которого драйвер не достать. И во всем этом приходится разбираться.

Первоначально рассмотрим конфигурацию принтера "правильным" способом, с применением стандартных UNIX-средств в консольном режиме. Затем рассмотрим конфигурацию с помощью графической утилиты, которая не идет ни в какое сравнение со стандартным способом — буквально за минуту можно сконфигурировать любой принтер.

Способы вывода на принтер

Как и для большинства задач, существует несколько способов добиться вывода данных на принтер. Конечно, в конце концов они замыкаются на простой вывод последовательности байтов в порт, к которому подключен принтер, однако с данными по пути следования от документа до распечатки могут производиться различные манипуляции.

Самый простой путь — прямой вывод информации без всякой предварительной обработки на порт принтера. Для этого достаточно выполнить всего лишь следующую команду:

```
cat mytext.txt > /dev/lp
```

Для DOS аналогичная команда будет выглядеть следующим образом:

```
copy mytext.txt > prn
```

Как обычно, это простота кажущаяся. Во-первых, для того, чтобы таким образом что-то отправить на печать, необходимо быть пользователем root — для остальных пользователей невозможно напрямую работать с файлами устройств. Во-вторых, зачастую вы получите на распечатке сплошную кашу из символов. Такое произойдет потому, что любой принтер имеет свой специальный язык управления, причем этих языков более десятка разновидностей. Так что выход для данного случая — использовать специальные утилиты, на вход которых подаются текстовые файлы, а на выходе — преобразованный с учетом языка управления принтера текст. Однако это крайне неудобно. Поэтому применяют специальные программные пакеты, предназначенные для управления печатью. Именно об этих программных пакетах и пойдет далее речь.

Система печати CUPS

CUPS (Common UNIX Printing System, общая система печати для UNIX), интересна своими богатыми возможностями. В ней реализован протокол печати, сходный с протоколом HTTP, заменяющий морально устаревший протокол LPD.

Поддерживает форматы Adobe PostScript, PDF, HP-GL/2, TIFF, JPEG, PNG, PBM, PGM, PPM, GIF, SGI, RGB, Sun Raster, Kodak Photo CDTM. Интересным моментом для администратора являются следующие особенности системы:

- правила управления доступом;
- наличие системы квот;
- авторизация пользователя;
- ведение log-журналов.

Программный пакет LPD

LPD (Line Printer Daemon, демон линейной печати) — пожалуй, старейший программный пакет для печати в мире UNIX. Идеология стандартна для UNIX — программы-утилиты для управления процессом печати и программа-демон, обеспечивающая печать на несколько принтеров. Благодаря такому построению программного пакета вы имеете возможность одновременно работать с несколькими принтерами и настроить сетевую печать. В пакет входят следующие программы:

- lpd — демон системы печати;
- lpr — пользовательская команда печати. lpr выдает новое задание печати в очередь печати lpd. Синтаксис lpr очень прост:

```
lpr [ опции ] [ имя_файла ... ]
```

Если имя_файла не задано, lpr ожидает ввод данных со стандартного ввода. Это позволяет пользователям перенаправлять вывод команд в очередь печати;

- lpq — утилита для просмотра очереди печати. Команда, запущенная без аргументов, возвращает содержимое очереди печати принтера по умолчанию;
- lpc — утилита контроля lpd. С ее помощью можно производить любые манипуляции с очередью печати — добавлять и удалять задания, останавливать печать, переупорядочивать задания в очереди печати и т. д. lpc чаще всего используется в системах, где несколько принтеров установлено на один компьютер.

  Команда `lpc` обычно используется в интерактивном режиме, однако никто вам не мешает запускать на выполнение эту команду с опциями. Некоторые из опций приведены далее:

  - `disable` — запрещает добавление любых новых заданий печати;
  - `down` — запрещает все задания на принтере;
  - `enable` — разрешает ввод новых заданий в очередь печати;
  - `quit` (or exit) — покинуть lpc;
  - `restart` — перезагрузить lpd для данного принтера;
  - `status` — статус печати принтера;
  - `up` — разрешить все и запустить новый демон lpd.
- lprm — утилита для удаления задания из очереди печати. Команда `lprm` удаляет из очереди все задания печати, владельцем которых является пользователь, выполнивший эту команду. Для того чтобы отменить одиночное задание печати, надо сначала получить номер задания с помощью команды `lpq`, а затем сообщить полученный номер команде `lprm`.

Функционирует система следующим образом. При старте операционной системы стартует демон lpd. Используя файл /etc/printcap, он узнает, какие принтеры будет обслуживать. При запуске (пользователь что-то выводит на печать) lpr взаимодействует с lpd через именованный сокет /dev/printer и передает lpd-файл для печати и некоторую информацию о том, кто печатает и как печатать файл. Затем lpd печатает файл на соответствующем принтере в порядке очереди.

Настройка LPD

Начнем с простого — настроим простой струйный принтер фирмы Hewlett-Packard — HP DeskJet 400. Будем считать, что LPD уже установлен в вашей операционной системе, поскольку этот пакет входит во множество дистрибутивов как стандартная система печати.

Для добавления очереди печати к lpd вы должны внести запись в файл /etc/printcap и создать новый буферный каталог в каталоге /var/spool/lpd. Запись в файле /etc/printcap выглядит следующим образом:

```
# ЛОКАЛЬНЫЙ deskjet400
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :sh:
```

Вышеприведенная запись определяет принтер с псевдонимами lp, dj или deskjet, его спул печати размещается в каталоге /var/spool/lpd/dj. Отсутствует ограничение максимального размера задания. Печать производится на устройство /dev/lp0 и не сопровождается выводом страницы с именем человека, который печатает, добавленной в начало задания печати. Как вы видите — все очень просто. Но, во-первых, извечная проблема текстовых файлов UNIX и Windows — для UNIX в конце текстовой строки достаточно символа перевода строки, для Windows — необходимо наличие символов возврата каретки и перевода строки. Большинство современных принтеров рассчитаны для использования совместно с Windows, и поэтому для нормальной печати текста им также необходимо в конце текстовой строки наличие символов возврата каретки и перевода строки. Если не учесть эту особенность, при распечатке текста на принтере получится приблизительно следующее:

```
Строка номер один
                Строка номер два
                                Строка номер три
                                                Строка номер четыре
```

Это называется лестничным эффектом, и с ним необходимо бороться. Существует много способов, самый простой — написать небольшой фильтр, через который перед печатью будет пропускаться наш текстовый файл, а результат — уходить на печать.

Поправим нашу запись в файле /etc/printcap следующим образом:

```
# ЛОКАЛЬНЫЙ deskjet400
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/dj/filter:\
        :sh:
```

В документации к printcap описаны атрибуты принтера `if` — входной фильтр и `of` — выходной фильтр. Как видите, мы определили входной фильтр, расположенный в каталоге /var/spool/lpd/dj/ и носящий имя filter. Этот файл представляет собой две строчки, написанные на Perl:

```
#!/usr/bin/perl
while(<STDIN>){chop $_; print "$_\r\n";};
print "\f";
```

В результате мы получаем принтер, на котором корректно можно распечатать текстовые файлы, используя встроенные шрифты принтера. Для современного мира это не актуально — практически всегда используется графическая печать. Обычно печатают документы PostScript или графические файлы. На первый взгляд — нетривиальная задача, на самом деле — все достаточно просто. Вспомните еще раз идеологию UNIX — сколь угодно сложные задачи решать применением последовательности небольших утилит.

Для решения этой проблемы опять используется свойство файла printcap — использование входных и выходных фильтров. Если мы будем использовать фильтр, который может воспринимать произвольные типы файлов как ввод, обрабатывать их в зависимости от формата файла и производить вывод на принтер, — мы решим нашу задачу.

Такой фильтр называется *магическим фильтром* (magic-filter). Существует большое количество магических фильтров, причем наверняка несколько такого типа фильтров находится в вашем дистрибутиве операционной системы. Ниже приведены некоторые магические фильтры печати:

- APSFILTER — фильтр печати для стандартного lpd;
- lprMagic — фильтр печати с неплохими возможностями. Автоматически определяет тип входного документа, есть поддержка печати через Samba.

Учет ресурсов

Обычно в больших фирмах принято хранить информацию о том, кто, когда и сколько печатал. Стандартный LPD предоставляет очень небольшую помощь для учета ресурсов. Вы можете указать имя файла для учета ресурсов, используя атрибут `af=` в printcap, но, по большому счету, это не решение проблемы. Пожалуй, лучший вариант — использовать магический фильтр, который может писать данные в файл учета ресурсов, а вы будете обрабатывать этот файл позже каким-нибудь скриптом обработки статистики.

Программа печати LPRng

Доработанная версия LPD, по всей видимости, скоро станет стандартной во всех дистрибутивах Linux. LPRng более легка для администрирования и имеет значительно лучшие возможности по сравнению с LPD для администри-

рования большого количества принтеров (в том числе и сетевых). Более безопасна с точки зрения администратора, поддерживает аутентификацию через PGP или Kerberos.

Программный пакет netcat

netcat — простой программный пакет для работы с принтерами. Удобен и прост в настройке, имеет проблемы с сетевой печатью, однако для домашнего пользователя, которому не нужна сеть, — очень неплохой вариант.

Система печати PDQ

PDQ (Print Don't Queue, печатать не буферизуя). Это система печати без центрального демона. Она включает возможность объявления настроек печати, а также графическую утилиту и утилиту командной строки для настройки и вывода на печать.

Для управления печатью используются следующие программы:

- Xpdq — приложение для X Windows, которое показывает список доступных принтеров и данные об очереди печати. Вы можете установить настройки вашего драйвера принтера, используя диалоговое окно **Driver Options**; обычно можно установить параметры двунаправленного соединения, плотность печати, размер и тип бумаги и т. д.;
- Pdq — утилита командной строки. Она может использоваться вместо команды `lpr` в большинстве случаев. Подобно lpr, она печатает либо перечисленные файлы, либо данные со стандартного ввода.

Функционирует PDQ следующим образом:

- запускается pdq или xpdq с указанием файла, который необходимо распечатать;
- выбирается принтер;
- определяются параметры печати — двухсторонняя печать, количество копий, качество печати и т. д.;
- программа анализирует содержимое файла, который вы печатаете, и следует инструкциям, записанным в файле драйвера PDQ, которые описывают как обрабатывать ваши данные для печати на данном принтере с заданными параметрами;
- программа посылает обработанные данные на принтер через указанный интерфейс — прямо на /dev/lp0, или сетевому демону LPD, или на факс-гейт (специальную программу, на вход которой поступают документы, предназначенные для отправки по факсу. Эта программа при помощи факс-модема дозванивается до нужного абонента и автоматически отправляет факс);

- если PDQ не может послать данные на принтер указанным способом, то она запускает в фоновом режиме процесс, который пытается произвести печать.

Настройка PDQ

PDQ может быть настроена либо администратором, либо обычным пользователем. Администратор для настройки PDQ редактирует файл /etc/printrc, а обычный пользователь может изменять только свой персональный файл .printrc.

PDQ позволяет пользователям выбрать принтер, на который будет производиться печать. Принтеры в PDQ определяются как комбинации драйвера и интерфейса и являются текстовыми описаниями в файле настройки PDQ.

Интерфейс PDQ описывает то, как данные посылаются на принтер. Ниже приведены некоторые параметры интерфейса:

- `local-port` — интерфейс локального порта работает с параллельным или последовательным портом на той машине, на которой запущен PDQ. Используя этот интерфейс, PDQ может печатать прямо в параллельный порт;
- `bsd-lpd` — интерфейс `bsd-lpd` общается по сети с демоном LPD или с работающим по протоколу LPD сетевым принтером. PDQ поддерживает постановку, отмену заданий и запросы к интерфейсу LPD.

Драйвер PDQ описывает, как перевести выводимые на принтер данные в формат, который понимает принтер. Для принтеров, понимающих PostScript, он будет включать преобразование из ASCII в PostScript; для не-PostScript-принтеров он будет описывать преобразования из PostScript в язык принтера, используя GhostScript.

Для того чтобы определить принтер в PDQ, необходимо запустить xpdq и выбрать команду меню **Printer | Add printer**. Этот мастер настройки проведет вас через выбор нужного драйвера и интерфейса.

Вот, собственно, и все по настройке PDQ. В том случае, если вашего принтера нет в списке поддерживаемых программой PDQ принтеров — почитайте документацию, там описано, как можно самостоятельно добавить ваш принтер в список драйверов.

Система буферизации печати PPR

PPR — система буферизации печати, ориентированная на PostScript. Она включает в себя хорошие возможности учета, поддержку клиентов Appletalk, SMB и LPD. Система PPR, как и другие перечисленные системы буферизации, может вызывать Ghostscript для работы с принтерами, не понимающими PostScript.

Печать на сетевой принтер

Одним из важных свойств пакетов PDQ и LPD является то, что они позволяют осуществлять печать по сети на принтер, физически подключенный к другому компьютеру, принт-серверу или просто сетевому принтеру.

Для того чтобы разрешить удаленным компьютерам, печатать на ваш принтер, используя протокол LPD, вы должны перечислить эти компьютеры в файле /etc/hosts.lpd. Помимо этого, вы можете разрешить только определенным пользователям с других компьютеров печатать на ваш принтер.

Для того чтобы печатать на другой компьютер вы должны в /etc/printcap сделать следующую запись:

```
# Удаленный deskjet400
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :rm=machine.out.there.com:\
        :rp=printername:\
        :lp=/dev/null:\
        :sh:
```

Как видно из вышеприведенного текста, на нашем компьютере существует каталог очереди печати, обслуживаемой lpd. Это позволяет сохранить и распечатать позднее задание, если удаленная машина занята или отключена. Также мы определяем имя компьютера, который предоставляет нам свой принтер (`machine.out.there.com`), имя принтера на удаленном компьютере (`printername`) и показываем, что сетевой принтер не подключен ни к какому ресурсу на нашем компьютере (`lp=/dev/null`).

Печать на Ethernet-принтер

Обычно высокоскоростные принтеры, позиционируемые производителем как устройства для совместной печати, имеют встроенный сетевой интерфейс, на который вы можете печатать, используя протокол LPD. Обычно в инструкции, идущей в комплекте с принтером, описывается, каким образом необходимо настроить клиентский компьютер для печати на сетевой принтер. Например, следующая запись в файле printcap используется для работы с сетевым принтером фирмы Hewlett-Packard:

```
lj-5|remote-hplj:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/lj-5:\
        :rm=printer.name.com:rp=raw:
```

Принтеры HP Laserjet с интерфейсами Jet Direct поддерживают две встроенных очереди LPD — "raw", которая принимает PCL (или PostScript) и "text", которая принимает "чистые" файлы ASCII и автоматически справляется с лестничным эффектом.

Графические утилиты конфигурирования принтера

Перейдем к визуализации информации — все-таки некоторые вещи проще делать в X Window.

В дистрибутиве Red Hat Linux есть много удобных утилит, и одна из них printconf-gui. С помощью этой простой утилиты установим принтер HP DeskJet 400.

На рис. 35.1 вы видите панель системных установок, среди которых присутствует **Printer Configuration**.

Рис. 35.1. Панель системных установок

После запуска утилиты конфигурации принтера мы увидим небольшое окно (рис. 35.2) всего лишь с тремя пунктами меню.

Перейдем к собственно конфигурации принтера. Процедура занимает мало времени и в большинстве случаев проходит без осложнений. Как видно из

рис. 35.2, в верхней части окна утилиты присутствует кнопка **Новая**. Это не описка, в отличие от Windows мы добавляем в операционную систему не *новый принтер*, а *новую очередь печати*. Нажимаем. И видим окно, изображенное на рис. 35.3.

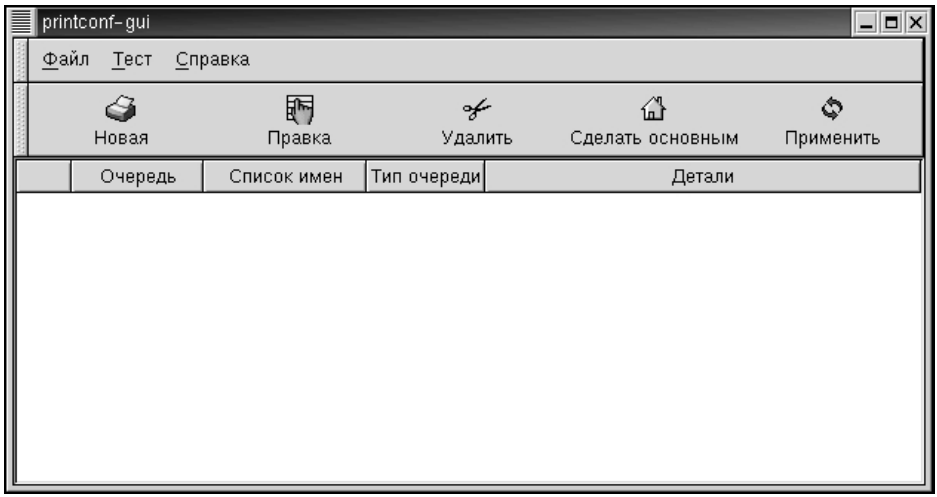

Рис. 35.2. Внешний вид утилиты printconf-gui

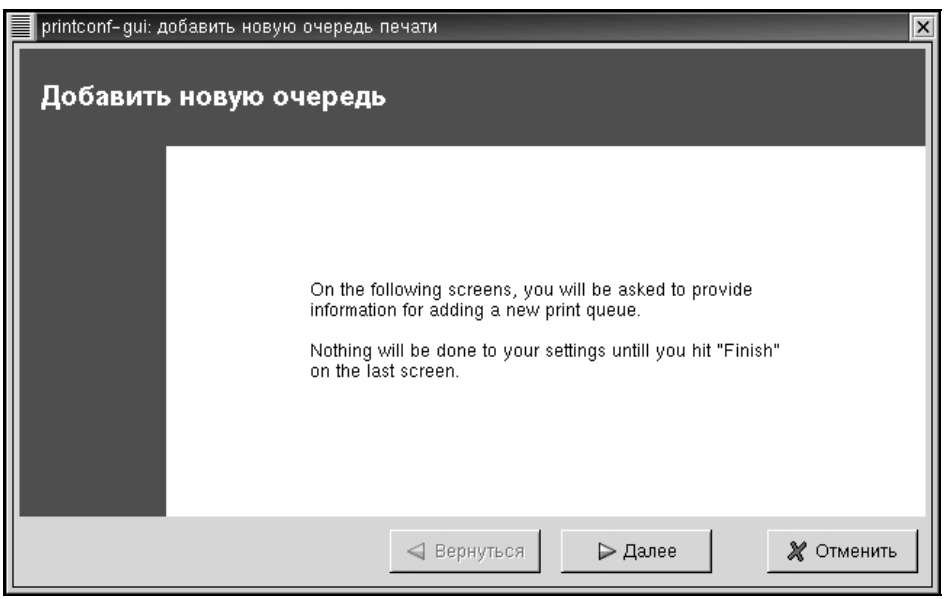

Рис. 35.3. Начало создания новой очереди печати

Переведем с английского обращение системы: "В последующих окнах вы должны ответить на предлагаемые вопросы для добавления новой очереди печати. Ничего с вашими настройками не произойдет до тех пор, пока вы не нажмете кнопку **Финиш** в последнем окне" (в смысле, все изменения в конфигурационные файлы будут внесены только по нажатии кнопки **Финиш**). Нажимаем кнопку **Далее**. Получаем следующее окно (рис. 35.4), где нам необходимо сделать следующее:

1. Написать имя для очереди печати (в нашем случае, как вы видите, — `hp400`).
2. Выбрать тип очереди.

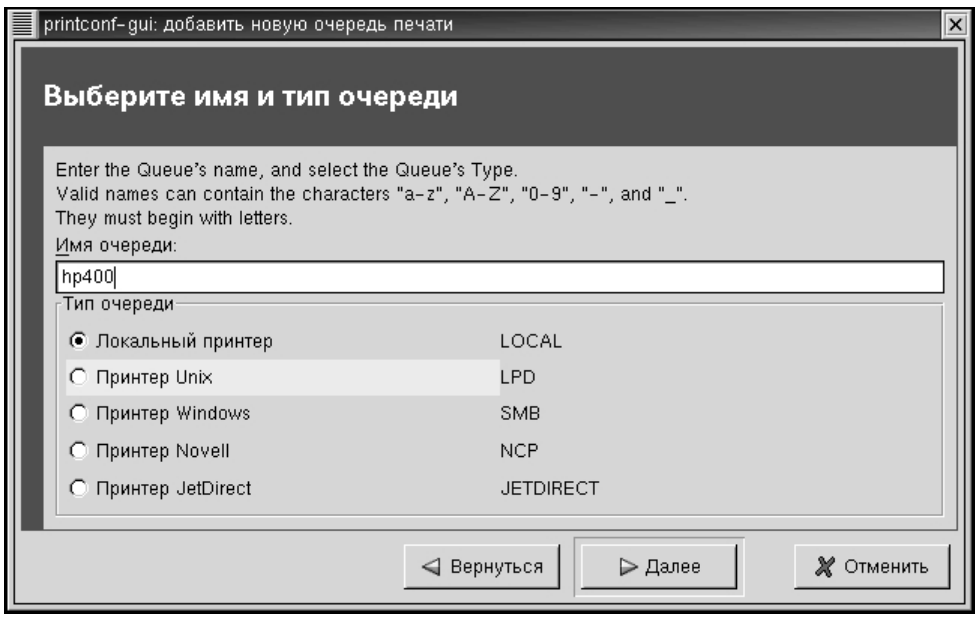

Рис. 35.4. Выбор имени и типа очереди печати

С первым понятно — можно использовать английские буквы A—Z, a—z, цифры от 0 до 9 и символы тире и подчеркивания, причем имя должно начинаться с буквы.

Со вторым пунктом несколько сложнее. Как видно из рисунка, нам предлагают на выбор пять пунктов:

- **Локальный принтер**;
- **Принтер Unix**;
- **Принтер Windows**;
- **Принтер Novell**;
- **Принтер JetDirect**.

С локальным принтером нет проблем — локальный принтер он и есть локальный.

Принтер UNIX — это сетевой принтер, подключенный к компьютеру под управлением UNIX (Linux) и предоставляющий к себе сетевой доступ. О настройке сетевого принтера мы уже упоминали выше.

Принтер Windows — сетевой принтер, установленный на компьютер под управлением Windows или на компьютер под управлением Linux, который является сервером Samba. В этом случае конфигурирование немного осложняется тем, что вам (или утилите конфигурирования, которая за вас это сделает) необходимо установить и настроить Samba-клиент.

Принтер Novell — сетевой принтер, установленный на сервер под управлением операционной системы Novell Netware или на компьютер под управлением Linux, который является сервером MARS_NWE.

Принтер JetDirect — сетевой принтер, работающий по протоколу JetDirect, либо принт-сервер. Обычно по этому протоколу работают принтеры фирмы Hewlett-Packard.

Как видно из рис. 35.5, ваша задача — выбрать, к какому интерфейсу подключен ваш принтер. Одновременно с появлением этого окна программа сканирует порты компьютера и предлагает наиболее подходящий (с ее точки зрения). Если ничего похожего нет, вы можете самостоятельно определить интерфейс, к которому подключен ваш принтер. Идем далее.

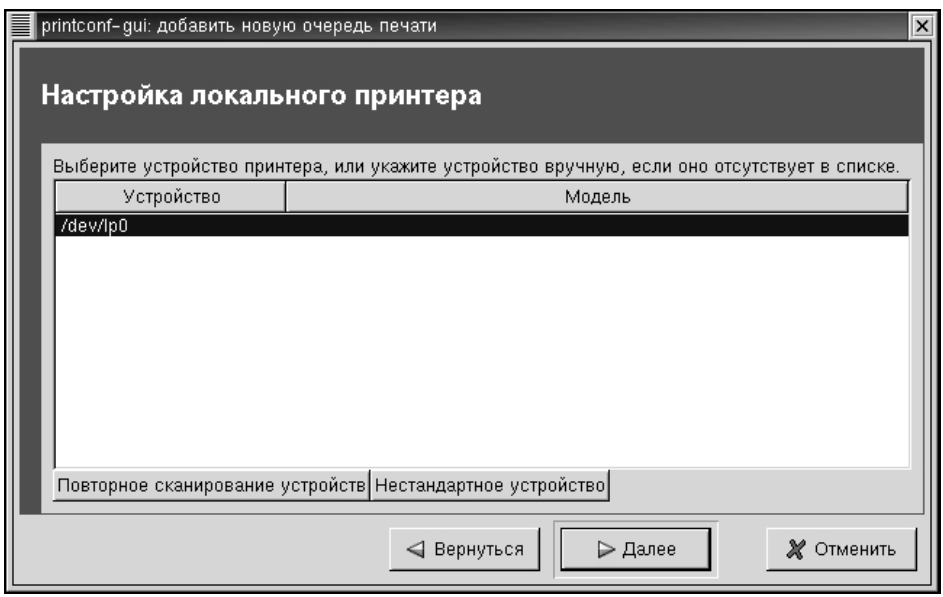

Рис. 35.5. Выбор порта подключения принтера

В следующем окне (рис. 35.6) мы должны выбрать наш принтер и драйвер для него. Если такого принтера нет в списке, посмотрите документацию на принтер, там должна быть информация, с какими моделями принтеров совместимо данное устройство.

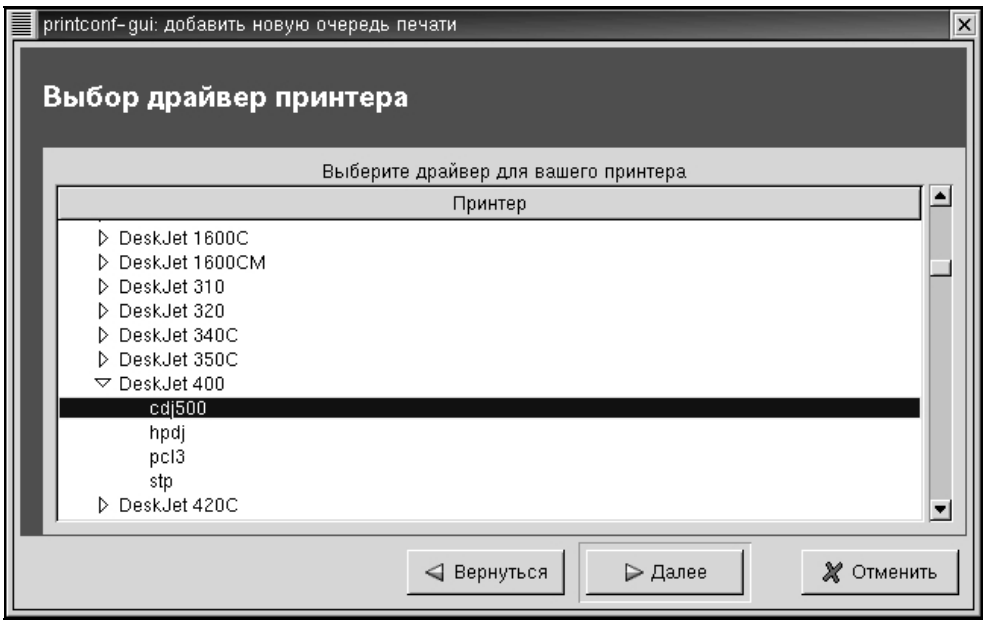

Рис. 35.6. Выбор драйвера принтера

Как видно из рисунка, для нашего принтера существует несколько драйверов, и нам ничего не остается, как выбрать его наугад. Мы выбрали cdj500 и ошиблись. Но об этом далее. Вот, наконец, получено окно (рис. 35.7) с заветной кнопкой **Завершить**.

В этом окне вам покажут, что́ вы выбрали и если вы где-то ошиблись — вернитесь назад и поправьте. Вот вы и добавили принтер в вашу операционную систему (рис. 35.8).

Теперь необходимо проверить, правильно ли установлен принтер. Для этого идем в пункт меню **Тест** и видим, какие тесты можно проделать — распечатать текстовую страницу, распечатать текст с использованием Ghostscript, распечатать картинки в разных форматах.

Попробовали. И не получилось... Как вы помните, в качестве драйвера принтера мы выбрали cdj500 — не обратили внимания на символ "c" — color. В результате наш принтер очень настойчиво попросил заменить чер-

ный картридж на цветной. Придется взять другой драйвер. Для того чтобы поправить параметры очереди, нажмите на кнопку **Правка**.

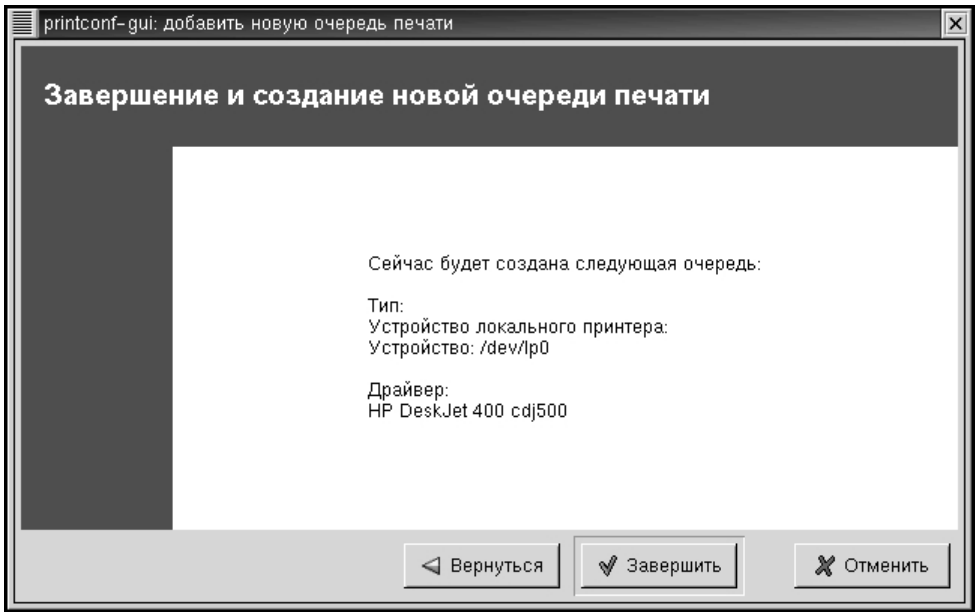

Рис. 35.7. Окно завершения создания очереди печати

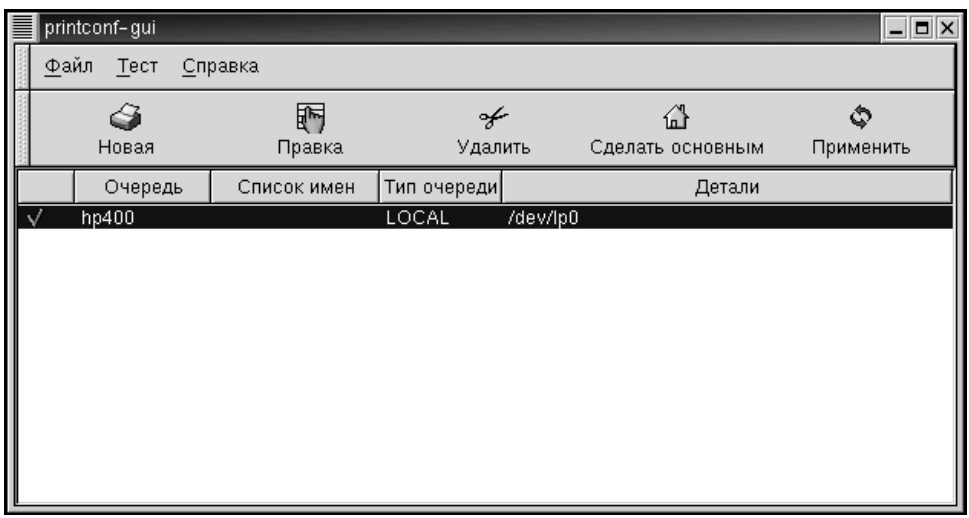

Рис. 35.8. Наша новая очередь печати

Как видите, получили окно (рис. 35.9) с четырьмя вкладками. В нашем случае изменять имя очереди нет надобности, поэтому идем далее.

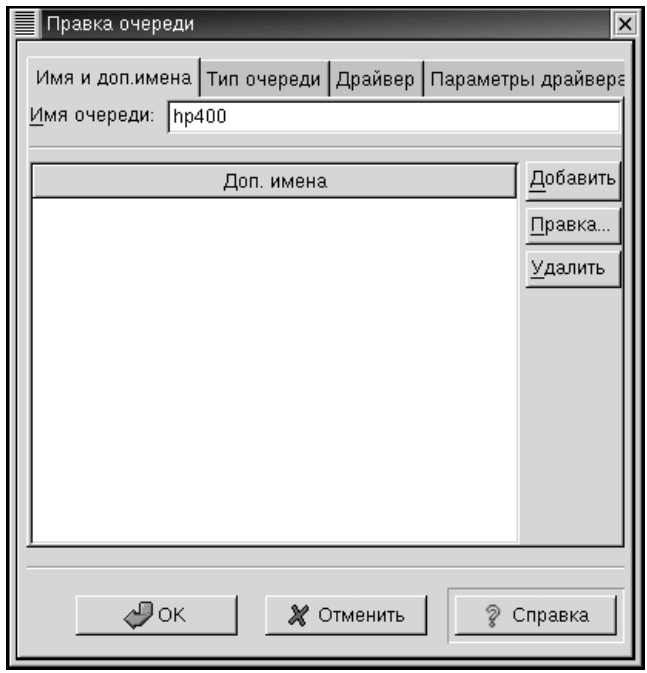

Рис. 35.9. Окно **Правка очереди**

На вкладке **Тип очереди** (рис. 35.10) можно поменять тип принтера и используемый порт. Поскольку у нас проблемы, заведомо не связанные с интерфейсом, — идем далее.

Вот то, что нам надо (рис. 35.11). Меняем тип драйвера с `cdj500` на `hpdj`. Забегая чуть вперед, скажем — помогло.

Затем перейдем на последнюю вкладку (рис. 35.12). Эта вкладка самая сложная — большое количество параметров, влияющих на печать. Тип используемой бумаги, качество печати, разрешение и т. п. Нажимаем кнопку **Сохранить** и проверяем печать. Все заработало.

Вот и все о настройке принтера.

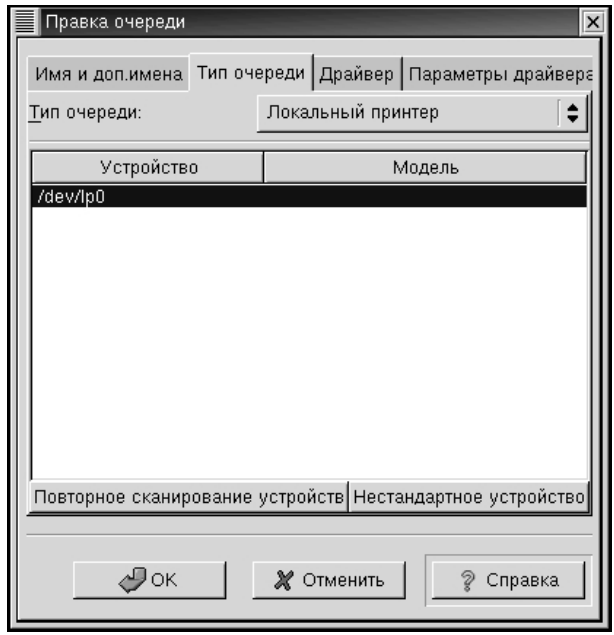

Рис. 35.10. Изменение типа принтера и используемого порта

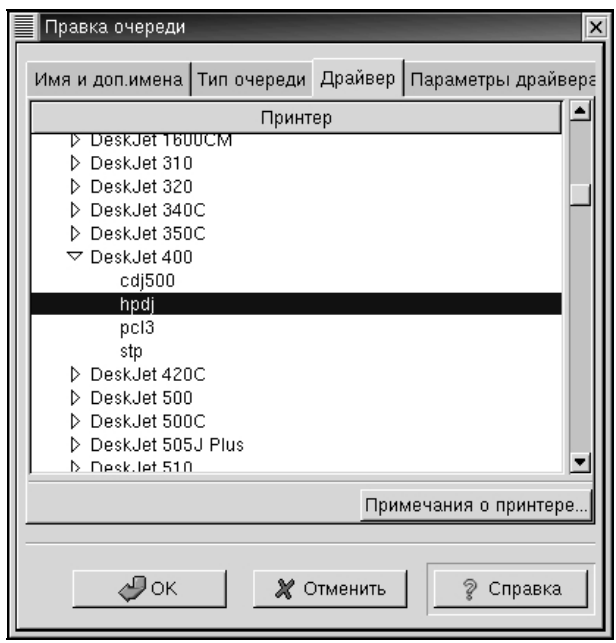

Рис. 35.11. Изменение типа драйвера

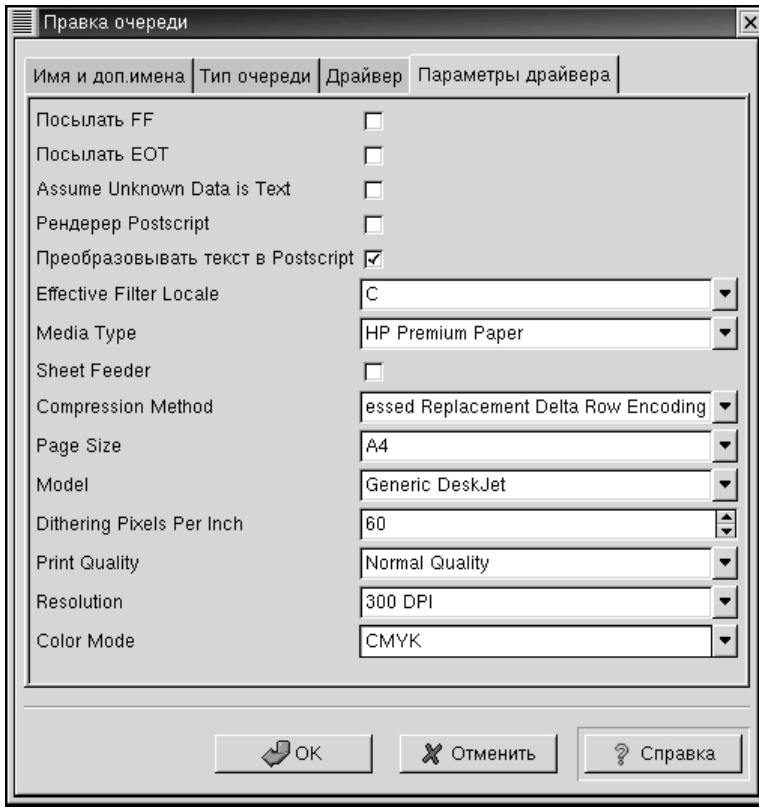

Рис. 35.12. Настройка параметров драйвера

Ссылки

- **hpinkjet.sourceforge.net** — драйверы для принтеров Hewlett-Packard непосредственно от фирмы-производителя (поддерживается более 60 моделей).
- **61.251.162.120:8080** — драйверы для принтеров Samsung от фирмы-производителя. Поддерживаются все принтеры серии ML.
- **http://www.linuxrsp.ru/artic/print_server.html** — Юрий Лушня. Печать в Linux с железными нервами.
- **linuxcenter.ru/lib/hardware/usbprinter.phtml** — Юрий Лушня. Настраиваем USB-принтер под Linux.
- **linux.yaroslavl.ru/Docum/Rus/print.html** — В. Толпекин. Настройка сетевого принтера для печати русского текста.
- **www.astart.com/lprng/LPRng.html** — страница проекта LPRng.

- **www.freebsd.org/~andreas/#apsfilter** — страница APSFILTER: Магический фильтр для печати.
- **metalab.unc.edu/pub/Linux/system/printing/** — lprMagic: Фильтр печати с неплохими возможностями.
- **feynman.tam.uiuc.edu/pdq/** — страница PDQ.
- **ftp://ppr-dist.trincoll.edu/pub/ppr/** — местонахождение PPR — системы буферизации печати, ориентированной на PostScript.
- **www.Linux-USB.org** — сайт, посвященный USB-устройствам и их применимости с точки зрения Linux.
- **http://www.linuxdoc.org/** — сайт, содержащий много интересной документации по Linux на английском языке.
- **www.citycat.ru/linux/docs/index.html** — сайт, содержащий много интересной документации по Linux на русском языке.
- **www.l0pht.com/~weld/netcat/** — страница netcat пакета для работы с принтером.
- **www.penguincomputing.com/prtools/npadmin.html** — страница npadmin — программы для управления сетевыми принтерами. Управление осуществляется через SNMP.
- **www.redhat.com** — Red Hat Linux 7.2. The Official Red Hat Linux Customization Guide.
- **www.redhat.com** — Red Hat Linux 7.2. The Official Red Hat Linux Getting Started Guide.
- **www.redhat.com** — Red Hat Linux/x86 7.2 Release Notes.
- Linux Printing HOWTO — Mark Komarinski. Использование печати в Linux. Перевод Alex Ott *(см. гл. 13)*.

Часть VI

Разное

Глава 36

Сканер

Поскольку в среде неискушенных пользователей бытует мнение, что операционная система Linux не предназначена ни для чего, кроме как для организации различных серверов, то эти пользователи и не помышляют об использовании компьютера с операционной системой Linux в качестве мультимедийного компьютера или графической станции. В следующих главах мы постараемся убедить вас, что графика, мультимедиа и Linux вполне совместимы.

Начнем, пожалуй, с рабочего места дизайнера. Для организации рабочего места дизайнера необходимо иметь следующее:

- ❒ большой, хороший монитор;
- ❒ современную видеокарту;
- ❒ сканер;
- ❒ принтер;
- ❒ графический редактор с мощными возможностями.

С монитором вообще проблем практически нет — как вы уже убедились, прочитав *гл. 34*, посвященную X Window, настроить монитор можно именно так, как вам хочется и как позволит ваша аппаратура. То же относится и к видеокарте.

О принтерах и их настройке мы уже тоже знаем из *гл. 35*.

Графический редактор Gimp — ничем не уступает Photoshop, а кое в чем и превосходит его, в том числе и по цене.

Остается один существенный компонент — сканер. Именно поддержке сканеров в Linux и посвящена данная глава.

До последнего времени производители аппаратного обеспечения, мягко говоря, не баловали наличием драйверов для своих устройств под Linux, поэтому приходилось выходить из положения своими силами. Если драйверы для сетевых карт, большинства видеокарт и принтеров энтузиасты всеми правдами и неправдами разрабатывали, портировали или приспосабливали

уже существующие, то с драйверами для "экзотической" периферии (с точки зрения пользователя офисного компьютера или разработчика программ) — сканеров, фотокамер, плат видеозахвата — дела обстояли совсем печально.

Отголоски этих времен и до сих пор чувствительно отзываются для обычного домашнего пользователя — для многих периферийных устройств, особенно выпущенных два-три года назад, не существует драйверов или программ, способных полностью реализовать их возможности. К большому сожалению, это касается и сканеров. Для того чтобы заставить работать сканер в операционной системе Linux в настоящее время, по большому счету, существует только один программный пакет — SANE. И, как уже упоминалось ранее, — далеко не для всех сканеров существуют драйверы. Помимо того, что производители не озаботились написанием драйверов, тяжелое положение с драйверами сложилось также из-за разнообразия типов интерфейсов, применяемых в сканерах.

Как известно, большая часть современных сканеров имеет один из четырех (а иногда два из четырех) интерфейсов:

- SCSI;
- параллельный (подключаемый к принтерному порту);
- USB;
- IEEE-1394.

Помимо этого, существуют сканеры, которые имеют свой оригинальный интерфейс и, соответственно, специальную интерфейсную плату, устанавливаемую в компьютер, а так же сканеры, подключаемые к последовательному порту.

Не удивительно, что в условиях отсутствия спецификаций (а в бизнес-мире "хорошим тоном" является объявление спецификаций коммерческой тайной) Linux-сообщество не смогло в полной мере самостоятельно создать необходимые драйверы. Еще одним тормозом в расширении применения сканеров для Linux явилось то, что еще года два назад наиболее массовым на рынке был сканер с SCSI-интерфейсом, причем с целью его удешевления производитель обычно комплектовал сканер SCSI-контроллером с урезанными функциями, либо не совсем отвечающий SCSI-стандарту.

Впрочем, с приходом параллельного и USB-интерфейса, а также из-за того, что электроника сканеров сейчас производится пятью-семью фирмами, положение со сканерами в операционной системе Linux постепенно выравнивается.

Начинать необходимо с выбора сканера. К сожалению, в отличие от Windows, где работает практически любой сканер, существует не так уж много моделей сканеров, поддержка которых реализована в Linux и пакетом SANE *полностью*. Значительно больше моделей сканеров, поддержка которых системой реализована лишь частично. Списки поддерживаемых Linux сканеров вы можете посмотреть на сайтах, перечень которых находится в конце главы.

В табл. 36.1 приведен список некоторых полностью поддерживаемых Linux сканеров, причем только тех фирм, сканеры которых реально могут быть приобретены нашими пользователями.

Таблица 36.1. *Список сканеров, полностью поддерживаемых Linux*

Фирма-производитель	Модель сканера	Интерфейс
Acer/Benq	Prisa 620U	USB
	Prisa 640U	
	Prisa 640BU	
	AcerScan 1240	
	AcerScan 3300	
	AcerScan 4300	
	AcerScan 5300	
Agfa	Snapscan 1212U	USB
	Snapscan 1236U	
	Snapscan e20	
	Snapscan e40	
	Snapscan e50	
Epson	GT-7000	USB
	Perfection 610U	
	Perfection 636U	
	Perfection 640U	
	Perfection 1200U/Photo	
	Perfection 1240U/Photo	
	Perfection 1640SU	
	Perfection 1650/Photo	
	Actionscanner II	Параллельный
	GT-5000	
	GT-6500	
	ES-300C	
	ES-600C	
	ES-1200C	
	GT-5500	SCSI
	Perfection 636S	
	ES-8500	
	GT-8000	
	GT-7000	

Таблица 36.1 *(продолжение)*

Фирма-производитель	Модель сканера	Интерфейс
Hewlett-Packard	ScanJet 4100C	USB
	ScanJet 5200C	
	ScanJet 6200C	
	ScanJet 6250C	
	ScanJet 6300C	
	ScanJet 6350C	
	ScanJet 6390C	
Microtek	Scanmaker X6	USB
	Scanmaker 3600	
	Scanmaker V6 USB	
	Scanmaker X12 USB	
Minolta	Scan Dual II	USB
	Plug-a-Scan 600CU	
	Plug-a-Scan 1200UB	
	Plug-a-Scan 1200CU	
	Plug-a-Scan 1200CU Plus	
Mustek	600 IIIEP Plus	Параллельный
Umax	Paragon 600 II N	
	Paragon MFS-6000CX	SCSI
	Paragon MFS-12000CX	
	Paragon MFC-600S	
	Paragon 600 II CD	
	ScanMagic 600 II SP	
	Paragon MFC-800S	
	Paragon 800 II SP	
	Paragon MFS-6000SP	
	Paragon MFS-8000SP	
	Paragon MFS-1200SP	
	Paragon MFS-12000SP	
	ScanExpress 6000SP	
	ScanExpress 12000SP	
	ScanExpress 12000SP Plus	

Таблица 36.1 *(окончание)*

Фирма-производитель	Модель сканера	Интерфейс
	Vista S6	SCSI
	Vista S6E	
	UMAX S-6E	
	UMAX S-6EG	
	Vista-S8	
	Supervista S-12	
	UMAX S-12	
	UMAX S-12G	
	Astra 600S	
	Astra 610S	
	Astra 1200S	
	Astra 1220S	
	Astra 2200 (SU)	
	Astra 2400S	
	Astra MX3	
	Mirage D-16L	
	Mirage II	
	Mirage Iise	
	PowerLook	
	PowerLook II	
	PowerLook III	
	PowerLook 270	
	PowerLook 270plus	
	PowerLook 2000	
	Astra 6400	IEEE-1394
	Astra 6450	
	PowerLook 1100	

Настройка Linux для подключения сканера

Зачастую новое USB-устройство ядро операционной системы Linux не опознает и что с ним делать, соответственно, не представляет. Поэтому необходимо самостоятельно определить наше устройство. Для этого в файл /etc/modules.conf следует добавить строку:

```
options scanner vendor=0x04b0 product=0x100 read_timeout=8000
```

Конкретно для имеющегося USB-сканера, вполне вероятно, необходимо будет подставить свои значения.

Может возникнуть проблема со слишком маленьким временем ожидания подтверждения в драйвере. Для решения этой проблемы придется поэкспериментировать с параметром `read_timeout` в вышеприведенной строке, где `read_timeout` задается в сотых долях секунды.

Параметры вашего USB-сканера можно посмотреть в log-файлах операционной системы:

```
hub.c: USB new device connect on bus1/1, assigned device number 5
usb.c: USB device 5 (vend/prod 0x4b0/0x100) is not claimed by any active
driver.
/etc/hotplug/usb.agent: ... no drivers for USB product 4b8/110/110
```

Как видно из сообщения — ядро операционной системы ничего не знает о данном сканере. Чтобы решить эту проблему, в файле /etc/hotplug/usb.distmap надо взять подходящую строчку от другого сканера этого же производителя:

```
scanner 0x0003 0x04b0 0x0107 0x000 0x000 0x00 0x00 0x00 0x00 0x00 0x00
0x00000000
```

И скопировать ее в файл /etc/hotplug/usb.handmap, заменив идентификатор устройства на `0x100`. После этого надо заново подключить сканер, и в log-файлах системы вы увидите тогда приблизительно следующее:

```
usb.c: USB disconnect on device 5
hub.c: USB new device connect on bus1/1, assigned device number 6
usb.c: USB device 6 (vend/prod 0x4b0/0x100) is not claimed by any active
driver.
usb.c: registered new driver usbscanner
scanner.c: probe_scanner: User specified USB scanner -- Vendor:Product -
4b0:100
scanner.c: USB Scanner support registered.
```

Есть еще один небольшой нюанс — если сканер долго не использовать, то он отключается, а модуль выгружается из памяти. В результате автоматический поиск устройства не работает. Для решения этой проблемы необходимо отключить и заново включить сканер.

Программный пакет SANE

Установленный нами для сканера драйвер ядра Linux обеспечивает только транспортный уровень протокола — он умеет передавать/принимать байты, но не более того. Для работы со сканером необходима программа, умеющая

общаться именно с данной моделью сканера. Наиболее популярным комплектом таких программ является пакет SANE.

SANE представляет собой интерфейс, который обеспечивает доступ к сканирующему оборудованию стандартным образом, а также библиотеку модулей для многих моделей сканеров. Поддерживаются USB- и SCSI-сканеры, сканеры, подключаемые к параллельному порту, и даже сканеры, подключаемые по интерфейсу FireWire (IEE-1394), а также некоторые цифровые камеры.

В дополнение к библиотеке модулей, в состав пакета входят программы для сканирования (frontends), а также и программы от других разработчиков. Более подробно об этих программах мы поговорим ниже.

Замечание

Есть такие понятия – frontend и backend. Frontend — программа, с которой непосредственно "общается" пользователь, обычно она имеет графический интерфейс. Никогда не взаимодействует напрямую с аппаратными средствами. Backend — программа, с которой пользователь обычно не работает. С этой программой взаимодействует программа frontend — она передает какую-то информацию, а backend управляет аппаратурой.

Обычно практически любой дистрибутив содержит пакет SANE, однако лучше всего взять его на сайте разработчиков, поскольку пакет динамично развивается и дополняется. После установки пакета желательно отредактировать список устройств в файле /etc/sane.d/dll.conf — все лишние устройства "закомментировать".

Добавим наше устройство в файл /etc/sane.d/scanner.conf:

```
usb /dev/usb/scanner0
```

После этого протестируем список доступных устройств командой:

```
scanimage -L -v
```

Среди распознанных устройств должно быть и наше устройство. Теперь можно посмотреть, на что способно наше устройство:

```
scanimage --help -v --device scanner:/dev/usb/scanner0
```

Вы должны увидеть нечто, подобное приведенному ниже:

```
--mode Binary|Gray|Color
--depth 8|16
--halftoning
--dropout None|Red|Green|Blue
--brightness -4..3
--sharpness -2..2
--gamma-correction
```

```
--color-correction --resolution
50|60|72|75|80|90|100|120|133|144|150|160|175|180|200|216|240|266|300|320
|350|360|400|480|600|720|800|900|1200|1600|1800|2400|
--threshold 0..255
--mirror[=(yes|no)]
--speed[=(yes|no)]
--auto-area-segmentation[=(yes|no)]
--zoom 50..250
--preview[=(yes|no)]
--preview-speed[=(yes|no)]
--source Flatbed|Transparency Unit
--film-type Positive Film|Negative Film
--focus-position Focus on glass|Focus 2.5mm above glass
```

Программное обеспечение (frontend) для пакета SANE

На сайте SANE заявлено о наличии в данный момент ряда программ для сканирования с помощью SANE. Рассмотрим их поподробнее.

Xsane

Графическая программа под X Window для сканирования. Поддерживает следующие возможности:

- сканирование и просмотр изображения в формате JPEG, PNG, PNM, PS, RAW, TIFF;
- отправку отсканированного изображения по факсу с помощью специальной утилиты;
- отправку отсканированного изображения по электронной почте с помощью специальной утилиты;
- управление гамма-коррекцией;
- встраивается в качестве plug-in в GIMP;
- работу в следующих операционных системах:
  - UNIX (Linux);
  - OS/2 с X11;
  - Windows 9*x*/NT/2000.

xscanimage

Программа для сканирования в среде X Window. По сравнению с Xsane имеет слишком мало возможностей:

- сохраняет сканированное изображение в файл в формате PNM;
- встраивается в качестве plug-in в GIMP.

QuiteInsane

Программа работает в среде X Window и позволяет сканировать и сохранять изображения. Базируется на библиотеке Qt.

FlScan

Программа работает в среде X Window и позволяет сканировать и сохранять изображения. Базируется на библиотеке FLTK.

scanimage

Утилита командной строки для сканирования изображений. Неудобна в использовании, зато работает в текстовом режиме.

TkScan

Как написано на сайте SANE — очень приятный графический интерфейс, который поддерживает сканеры Mustek, используя утилиту scanimage, входящую в состав SANE.

saned

Сетевой демон для удаленного сканирования. Существуют же сканеры с автоподачей оригиналов.

scanadf

Утилита командной строки, позволяющая задействовать дополнительные возможности сканеров с автоподачей оригиналов.

scanlite

Утилита для сканирования изображений, написанная на Java. В настоящее время находится в стадии beta-тестирования.

xcam

Графическая утилита для фотокамер. Немного не по теме данной главы, но эта программа входит в SANE.

Staroffice v5.2

Этот офисный пакет содержит простой интерфейс для сканирования, который использует SANE.

NSane

Графическая программа по взаимодействию с SANE в NeXTStep.

Программа VueScan

VueScan позиционируется разработчиками как альтернатива SANE. Включает библиотеку драйверов сканеров и графическую оболочку в одной программе. Исходные тексты программы не публикуются. Распространяется как Shareware — без оплаты не сохраняет сканированные изображения. Ориентирована на слайд-сканеры: поддерживает инфракрасный канал, фокусировку, установку времени экспозиции, пакетную обработку, многократное сканирование. Содержит специальные фильтры обработки изображений для пленки: удаление зерна, восстановление блеклых цветов.

Ссылки

- **www.bog.pp.ru** — Сергей Богомолов. Hardware: Использование USB-сканера в Linux.
- **www.digitalware.ru/static/dwscanners/** — обзор сайтов, посвященных сканерам и сканированию.
- **www.hamrick.com/vsm.html** — официальный сайт VueScan — программы для сканирования, содержащей набор драйверов для сканеров.
- **www.scaner.ru** — сайт, посвященный сканерам.
- **www.scaners.ru** — сайт, посвященный сканерам.
- **www.buzzard.org.uk/jonathan/scanners-usb.html** — список USB-сканеров, поддерживаемых SANE.
- **www.mostang.com/sane** — официальная страница пакета SANE.
- **panda.mostang.com/sane/sane-backends.html** — поддерживаемые сканеры.
- **www.qbik.ch/usb/devices/devices.php** — список USB-устройств, более или менее поддерживаемых Linux, с отзывами владельцев.
- **www.epsondevelopers.com/lscan.jsp** — страница на сайте Epson о драйверах сканеров для Linux.
- **www.xsane.org** — официальный сайт Xsane.
- **www.hamrick.com/vsm.html** — сайт программы VueScan.
- **sunsite.unc.edu/pub/Linux/apps/graphics/capture/** — месторасположение программы TkScan.

Глава 37

Различная "экзотическая" периферия и внешние устройства

В этой главе пойдет речь о таких устройствах, с которыми большинству пользователей, вероятно, в обычной жизни еще не довелось сталкиваться. Например — карманный персональный компьютер (КПК, PDA) или мобильный телефон с инфракрасным портом. Или цифровой фотоаппарат. Одним словом, экзотика, которая медленно становится нормой жизни. И основная проблема — каким образом компьютеру обмениваться информацией с этими приборами? Как обычно, производители всевозможных электронных устройств позаботились о программном обеспечении для Windows, а для альтернативных операционных систем практически ничего нет. Попробуем устранить этот недостаток и рассказать о программном обеспечении для синхронизации информации между Linux и вашими электронными новинками.

Linux и телефоны Nokia

Пожалуй, добрая треть мобильных телефонов, находящихся в эксплуатации у нашего населения, — это аппараты финской фирмы Nokia. Вы не замечали, что обыкновенная записная книжка, по крайней мере, по части записи телефонов, для вас — уже прошедшее время? Что все телефоны находятся либо в памяти вашего мобильного телефона, либо на его же SIM-карте? А не задумывались ли вы о перспективе потери мобильного телефона или выходе его из строя? Ведь в таком случае вы потеряете все телефонные номера, которые собирали на протяжении, наверное, целого года. Перспектива не радужная... Руками переписывать всю информацию с дисплея телефона на бумажку? Многие, наверное, уже забыли, как авторучку держать, все время на компьютере да на компьютере. Надо бы для этого компьютер и приспособить.

А в этом нам поможет замечательная программа Gnokii (рис. 37.1). Уже из логотипа понятно ее назначение.

Рис. 37.1. Логотип программы Gnokii

Программа предназначена для работы с мобильными телефонами фирмы Nokia. Ниже приведен список полностью поддерживаемых мобильных телефонов:

- ❒ Nokia 6130;
- ❒ Nokia 6150;
- ❒ Nokia 6190;
- ❒ Nokia 5110;
- ❒ Nokia 5130;
- ❒ Nokia 5190;
- ❒ Nokia 3210;
- ❒ Nokia 3110;
- ❒ Nokia 3810;
- ❒ Nokia 8110;
- ❒ Nokia 8110i;
- ❒ Nokia 2110;
- ❒ Nokia 2140;
- ❒ Nokia 6080;
- ❒ Nokia 640;
- ❒ Nokia 5160;
- ❒ Nokia 6160;
- ❒ Nokia 6185.

Как видите, список полностью поддерживаемых телефонов не очень велик, однако частично поддерживаемых моделей телефонов не меньше. Причем среди них попадаются как GSM- так и NMT-модели. Соединение с телефоном может производиться через специальный кабель, подключаемый к последовательному порту компьютера и телефону, или через инфракрасный порт.

На рис. 37.2 изображен внешний вид программы Gnokii.

Рис. 37.2. Внешний вид программы Gnokii

Для конфигурирования телефона следует зайти в соответствующее меню (рис. 37.3).

Для резервного копирования данных необходимо выполнить простую операцию (рис. 37.4).

Если ваш телефон подключен к компьютеру и запущена программа Gnokii, вы можете использовать программу для набора телефонного номера и звонка, отсылки SMS-сообщений (рис. 37.5), редактирования телефонной книги и т. п.

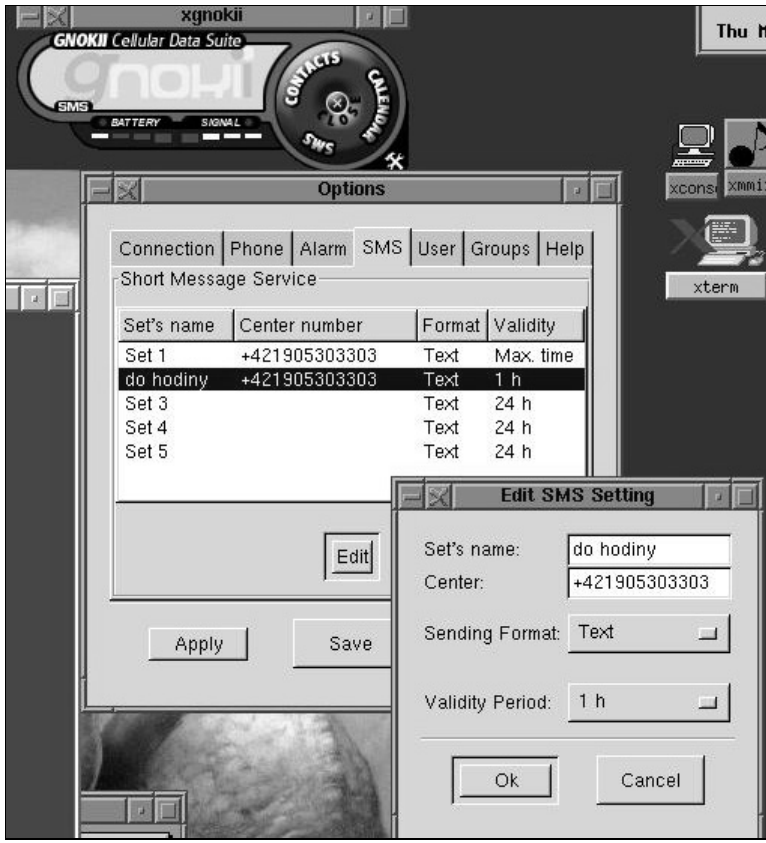

Рис. 37.3. Конфигурация номеров центра SMS-сообщений в программе Gnokii

Рис. 37.4. Резервное копирование телефонных номеров из телефона программой Gnokii

Рис. 37.5. Отсылка SMS-сообщения при помощи программы Gnokii

В общем, нужная программа. К сожалению, автору неизвестно о существовании подобного программного обеспечения для мобильных телефонов других производителей. Правда, создатели программы Gnokii обещают в следующих версиях программы поддержку телефонов Ericson, но пока этого еще нет.

Linux и КПК

Существует еще один класс устройств, которому не менее, а, пожалуй, и в большей степени необходима синхронизация с компьютером — карманные персональные компьютеры (КПК). Эти устройства можно условно разделить на четыре ветви:

- устройства, работающие под управлением операционной системы Palm OS;
- устройства, работающие под управлением операционной системы Epoc OS;
- устройства, работающие под управлением операционной системы Windows CE;
- устройства, работающие под управлением других операционных систем.

Синхронизации КПК с операционными средами первых двух типов и компьютером под управлением операционной системы Linux мы и рассмотрим далее.

Linux и Palm

КПК под управлением операционной системы Palm OS великое множество — это и собственно КПК производства фирм Palm, Sony и Handspring и множество КПК менее именитых производителей.

Для того чтобы соединить КПК под управлением операционной системы Palm OS и компьютер под управлением Linux, ничего сверхординарного ненужно — два устройства, так называемый *кредл* (от англ. *cradle*, колыбель — специальная подставка с разъемом для подключения к компьютеру и подзарядки) для синхронизации или инфракрасный порт на компьютере (в КПК он уже присутствует) и программа для синхронизации компьютера и КПК.

Для комфортной работы с КПК под управлением операционной системы Palm OS есть множество программ, но все эти программы используют в своей работе программный пакет, называемый Pilot-Link. В этом пакете есть все необходимое для работы с КПК под управлением операционной системы Palm OS. Однако в большинстве случаев вы не будете использовать этот пакет в полной мере, поскольку значительная часть утилит с успехом заменяется более удобной и красивой программой, работающей в X Window.

После установки программы Pilot-Link необходимо указать, к какому последовательному порту и на какой скорости подключен ваш КПК. Проще всего добавить следующие строки в файл /etc/profile:

```
export PILOTRATE=115200
export PILOTPORT=/dev/ttyS1
```

Здесь:

- `PILOTRATE` — скорость передачи данных от КПК к компьютеру. Эту скорость желательно установить как можно больше, в идеале — 115 200 бит/с. Однако, если вы для связи пользуетесь инфракрасным портом — могут возникнуть проблемы, особенно если ваш стол с компьютером стоит возле окна, и на улице вовсю светит солнце. Тут, как обычно, выхода два: или зашторить окно или понизить скорость передачи информации;
- `PILOTPORT` — эта переменная указывает, к какому порту подключен кредл синхронизации с КПК.

pilot-xfer

Утилита для синхронизации КПК и компьютера в консольном режиме. Ниже приведены основные опции командной строки этой программы:

- `-b [`*`каталог`*`]` — делает полную копию содержимого памяти КПК в указанный каталог;
- `-u [`*`каталог`*`]` — производит обновление копии памяти КПК в каталоге;
- `-s [`*`каталог`*`]` — синхронизирует каталог и память КПК;
- `-r [`*`каталог`*`]` — переносит содержимое каталога в память КПК;

- `-i файлы` — инсталлирует в КПК указанные файлы;
- `-m файлы` — инсталлирует в КПК те файлы, которых в нем нет;
- `-f база` — забирает соответствующую базу из КПК;
- `-d база` — удаляет из памяти КПК соответствующую базу.

Программы под X Window

Консольный режим хорош своим минимализмом, но иногда хочется красоты и удобства. Пойдем за ними на сайт **www.freshmeat.net**. В поле ввода поисковой системы сайта введем слово `pilot` и получим достаточно длинный список, в котором найдется десятка полтора программ, предназначенных для работы с КПК. Рассмотрим несколько из них.

gnome-pilot

Программа, являющаяся частью проекта GNOME, позволяет синхронизировать КПК с компьютером, устанавливать и удалять приложения, править записную книгу и т. п.

J-Pilot

Все, что написано о предыдущей программе, можно смело сказать и о программе J-Pilot. Внешний вид программы изображен на рис. 37.6.

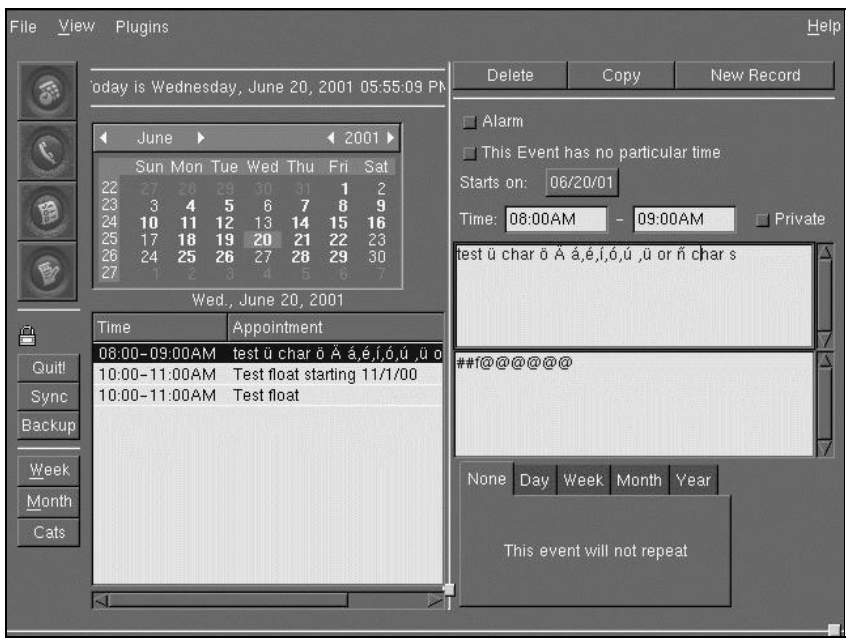

Рис. 37.6. Программа J-Pilot

KPilot

Программа для синхронизации КПК и компьютера, является частью проекта KDE. Внешний вид программы представлен на рис. 37.7.

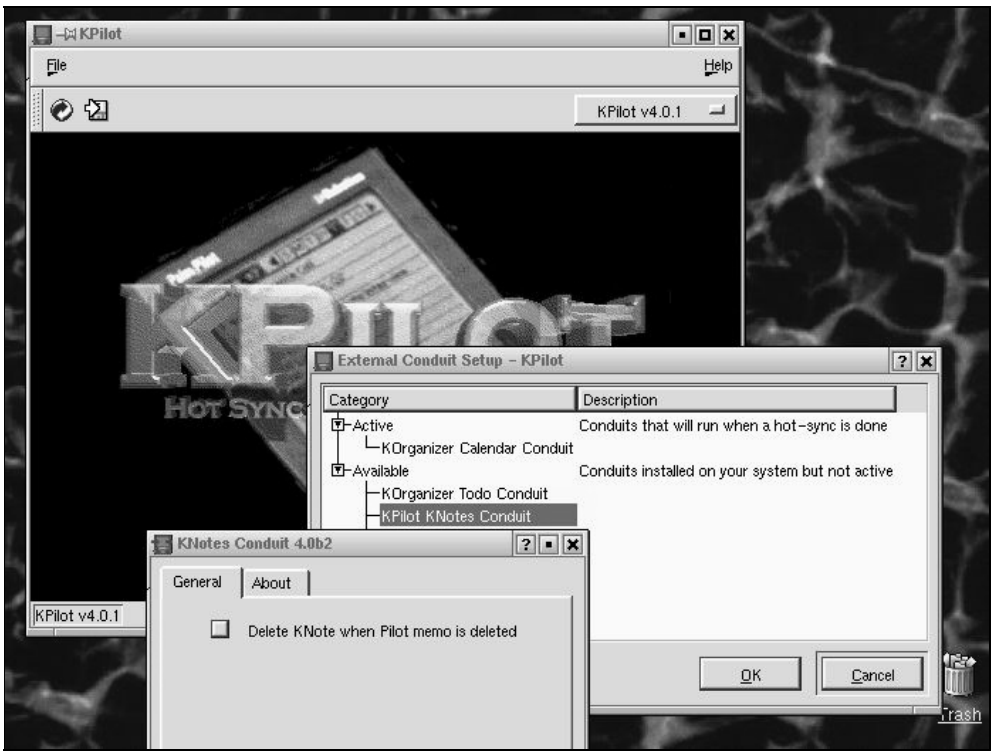

Рис. 37.7. Программа KPilot

Linux и Psion

Еще одним большим классом КПК являются устройства под управлением операционной системы Epoc производства английской фирмы Psion. Как правило, это более мощные устройства, со значительно бóльшим экраном и объемом памяти, чем устройства на базе операционной системы Palm OS. Еще одним немаловажным достоинством КПК Psion является наличие клавиатуры. Как и предыдущий класс КПК, Psion может соединяться с компьютером по последовательному интерфейсу или через инфракрасный порт. К сожалению, из-за политики фирмы Psion большого количества программного обеспечения для синхронизации Linux и Psion не появилось. Рассмотрим то, что есть.

PsiLin

Функциональный аналог программы PsiWin, написанной для Windows. Позволяет синхронизировать КПК и компьютер, производить резервное копирование, загрузку файлов на КПК.

plptools

Программное обеспечение для связи КПК и компьютера. В частности, позволяет подмонтировать память КПК или работать с КПК как с сервером FTP.

kpsion

Программное обеспечение, разработанное для использования в среде KDE. Предназначено для связи с КПК, в состав входит утилита для конвертирования документов формата Psion Word в форматы HTML или TXT. К сожалению, поддержка данного пакета прекращена.

p3nfs/p5nfs

Утилита для монтирования памяти КПК с использованием NFS.

Psiconv

Утилита для конвертирования документов формата Psion Word в форматы HTML или TXT.

Linux и TV Tuner

Помимо различных устройств, предназначенных для работы, к компьютеру можно подключить и устройства для развлечений. Одним из таких устройств является плата телевизионного приемника (TV-тюнер), зачастую совмещенная с радиоприемником. Для нормального функционирования такого тапа устройств необходимы две вещи — драйверы (и корректная их настройка) и соответствующие программы для просмотра телевизионных передач. Начнем с настройки драйверов.

Ключевое слово для нас при поиске информации — video4linux или video4linux2 — набор драйверов и документации для обеспечения поддержки видео под Linux.

Как правило, практически все платы TV-тюнеров используют в качестве центрального элемента одну из следующих микросхем:

- Bt848;
- Bt848a;
- Bt849/Bt878;
- Bt879.

Для обеспечения работы TV-тюнера необходимо выполнить несколько условий.

1. Иметь настроенную звуковую карту, поскольку звук с TV-тюнера передается по кабелю на вход звуковой карты.
2. Скомпилировать ядро операционной системы Linux с поддержкой следующих функций:
   - установить поддержку ядром операционной системы драйвера bttv;
   - установить поддержку I2C-интерфейса;
   - установить поддержку I2C bit-banging интерфейса;
   - в секции Multimedia Devices включить поддержку Video For Linux и установить BT8XX Video For Linux как модуль.
3. Проверить наличие устройства /dev/video* и если таковое отсутствует — создать его следующими командами:
   - `cd /dev`;
   - `./MAKEDEV video`.
4. После компиляции и установки ядра операционной системы Linux и модулей перегрузить компьютер и выполнить команду:

   ```
   /sbin/insmod bttv
   ```

5. В документации на вашу плату найдите точное ее название, а в документации на драйвер bttv выясните, поддерживает ли драйвер эту карту.

После выполнения этих условий можно переходить к настройке платы TV-тюнера.

Для проверки работоспособности платы TV-тюнера, а так же для подборки параметров для драйвера необходимо использовать какое-то программное обеспечение, способное работать с video4linux. Пожалуй, самый оптимальный вариант — прриенить программу xawtv, которая примечательна тем, что использует библиотеку libXaw и не требует никаких дополнительных специфических библиотек для компиляции.

Самый простой вариант заставить вашу плату TV-тюнера функционировать — попытаться, чтобы программное обеспечение самостоятельно определило ее тип. К сожалению, такая удача бывает не часто, поэтому нам ничего не остается, как внимательно изучить сопровождающую плату TV-тюнера документацию. Находим точное название платы и фирму-производителя и ищем это устройство в списке поддерживаемых драйвером bttv TV-тюнеров, где и определяется необходимый нам для успешного использования драйвера номер карты TV-тюнера. Если же ваше устройство в указанном списке отсутствует, остается только одно — настройка устройства методом подбора.

Для этих целей воспользуемся программой xawtv. После установки программы нам необходимо подправить конфигурационный файл .xawtv.

Небольшое отступление — у нас используется стандарт телевизионного вещания SECAM D/K, поэтому при редактировании конфигурационного файла .xawtv выставлен тип кодировки `SECAM`. Помимо этого, при настройке драйвера bttv необходимо выставить переменную `tuner type`. Для большинства плат TV-тюнеров и стандарта SECAM подходит `tuner type=3`. Однако для некоторых разновидностей плат переменной `tuner type` надо присвоить значение 1 или 5.

Конфигурационный файл .xawtv приведен ниже:

```
[global]
 fullscreen = 800 x 600
 freqtab = europe-east
 pixsize = 128 x 96
 pixcols = 1
 jpeg-quality = 75

 [defaults]
 norm = SECAM
 capture = over
 source = Television
```

В этом файле мы определили размер изображения, частотную таблицу каналов, качество jpeg-сжатия, стандарт телевизионного изображения и источник сигнала.

Теперь необходимо подобрать для драйвера bttv номер типа TV-тюнера, при котором наша плата будет нормально функционировать.

Алгоритм подбора следующий:

1. Устанавливаем модуль ядра операционной системы, поддерживающий i2c:

   ```
   modprobe i2c
   ```

2. Устанавливаем модуль ядра операционной системы, поддерживающий стандарт SECAM:

   ```
   modprobe tuner type=3
   ```

3. Устанавливаем модуль ядра операционной системы с драйвером bttv и типом карты TV-тюнера, равным 1:

   ```
   modprobe bttv card=1
   ```

4. Затем запускаем программу xawtv:

   ```
   xawtv &
   ```

5. Далее, с помощью клавиш <^> и <v> находим телевизионный канал, а с помощью клавиш <<> и <>> производим точную подстройку.

   Проверяем, как выводятся изображение и звук. Если телепередача не выводится нормально — черно-белое изображение, нет звука или вообще ничего не видно, не слышно — выполняем команду `q` в окне xawtv — и производим следующие действия:

   5.1. Выгружаем драйвер bttv:

   ```
   rmmod bttv
   ```

   5.2. Меняем тип карты TV-тюнера:

   ```
   modprobe bttv card=2
   ```

6. Повторяем пп. 4—5 до тех пор, пока не добьемся результата.

Однако в этот простой алгоритм могут добавиться еще кое-какие действия. Некоторые платы TV-тюнеров имеют в своем составе *отдельный* декодер звука (обычно микросхемы msp34xx, tda8425, tea6300). В этом случае необходимо дополнительно загружать соответствующие модули (предварительно их нужно скомпилировать).

Предположим, все прошло успешно, и вы определили параметры, с которыми надо загружать модули ядра, относящиеся к плате TV-тюнера. Теперь нам необходимо сделать так, чтобы эти модули автоматически загружались при старте операционной системы. Для этого в файл /etc/conf.modules следует добавить такие строки:

```
alias char-major-81-0 bttv
alias char-major-81 videodev
options tuner type=3
options bttv card=8
pre-install bttv modprobe -k tuner
```

Перезагружаем компьютер, запускаем опять программу xawtv и проверяем функционирование платы TV-тюнера. В случае успеха можно переходить к программам, функционирующим под управлением X Window.

wmtv

Программа интересна тем, что может в минимизированном виде выводить изображение. Много места не занимает, а в тот момент, когда идет что-то интересное — просто делаем на минимизированной программе двойной щелчок мышью и получаем увеличенное изображение. Программу можно настроить таким образом, что по двойному щелчку она будет вызывать внешнее приложение, например тот же xawtv.

kWinTV

Программа для просмотра телепередач под KDE. Удобна, красива, функциональна.

На рис. 37.8 вы можете увидеть внешний вид программы.

Рис. 37.8. Программа kWinTV

LIRC

LIRC (Linux Infrared Remote Control, программное обеспечение для управления устройством с помощью пульта дистанционного управления). Поскольку проект развивающийся, рекомендуется перед установкой скачать самую последнюю версию программного обеспечения с сайта разработчиков. Перед установкой ознакомьтесь со списком поддерживаемых устройств и документацией, поскольку вполне вероятно, что вам придется вносить изменения в драйвер bttv. В дистрибутиве LIRC содержатся примеры конфигурационных файлов для поддерживаемых устройств.

Поддержка управления с пульта дистанционного управления есть, например, в том же kWinTV.

Создание Real Video под Linux

Имея в составе компьютера плату TV-тюнера, можно получить с нее видеоизображение и закодировать его в формате Real Video, а при желании даже организовать видеовещание через локальную сеть или в Интернете.

Для организации видеозахвата и кодирования видеоинформации в формате Real Video необходимо выполнить следующие действия:

1. Получить Real Producer Basic с сайта **www.real.com**.
2. После процесса инсталляции зайти в систему как пользователь root, перейти в каталог, где установлен real producer, и выполнить:

```
realproducer -o /tmp/testing.rm -t 7 -a 3 -v 0 -f 0 -b "Testing Video"
-h "localhost" -c "Personal" -vc video -l 2:1,8:1
```

Таким образом вы захватили видеопоток с TV-тюнера, перекодировали его в Real Player 8 и записали в /tmp directory как testing.rm.

Этот простой пример показывает, как несложно получить телепередачу, записанную в формате Real Video. Чтобы узнать о всех возможностях Real Producer Basic — ознакомьтесь с документацией, идущей в комплекте с этим программным обеспечением.

Для организации трансляции видеопотока по сети необходимо на том же сайте получить Real Server и установить его в системе. К сожалению, Real Server — программа не бесплатная, бесплатно ей можно пользоваться только ограниченное время.

Как альтернативу можно использовать программный пакет ffmpeg — очень быстрый audio/video кодировщик/преобразователь, а так же потоковый сервер (видео, аудио).

Пакет SANE

В состав пакета SANE, который предназначен для работы со сканерами, входит модуль для захвата изображений с video4linux, который работает с платами TV-тюнера.

Видеокарта с TV-out

Теперь давайте разберемся с выводом видеоизображения, например на видеомагнитофон. В последние год-полтора стандартной видеокартой для домашнего компьютера стали видеокарты на базе чипов от nVidia. Существует много моделей видеокарт, которые, помимо своих прямых обязанностей по выводу изображения на монитор, так же осуществляют вывод изображения на телевизор, а иногда и захват видеоизображения. Поэтому дальнейший рассказ будет касаться видеокарт, основанных на чипах nVidia. Всю нужную информацию по настройке видеокарты можно получить из документации, идущей в комплекте с фирменными драйверами от nVidia.

Для настройки TV-out необходимо произвести следующие действия:

1. С сайта nVidia берем последние драйверы и устанавливаем их.

2. В файле /etc/X11/XF86Config-4 приводим данные к следующему виду:

```
Section "Module"
    Load  "dbe"
    Load  "glx"
    Load  "extmod"
    Load  "type1"
    Load  "freetype"
EndSection

Section "Device"
    Identifier  "NVIDIA GeForce2 DDR"
    VendorName  "nvidia"
    BoardName   "ABIT"
    Driver      "nvidia"
    VideoRam    32768
    Option      "DPMS"

    # запустите 'lspci' чтобы узнать BusID
    BusID       "PCI:1:0:0"

    # Если при переключении из консоли в X Window случаются
    # падения X'ов,
    # измените 3 на 1.

    Option "NvAGP" 3
    Option "ConnectedMonitor" "TV"

    # SVIDEO или COMPOSITE — в зависимости от того, каким образом
    # подключен TV к видеокарте,через svideo-разъем
    # или разъемом типа "тюльпан" (COMPOSITE)
    Option  "TVOutFormat" "COMPOSITE"

    # Описываем частотные характеристики телевизора.
    Option  "SecondMonitorHorizSync" "30-50"
    Option  "SecondMonitorVertRefresh" "60"

    # Какой телевизионный стандарт использовать для вывода
    # изображения, как следует выбирать либо PAL-I, либо NTSC-J
    Option  "TVStandard" "PAL-I"
```

```
    # Включаем режим TwinView
    Option  "TwinView"

    # Clone — дублирование на TV изображения с экрана монитора
    # можно использовать также "RightOf" "LeftOf" "Above" "Below",
    Option  "TwinViewOrientation" "Clone"
    # Сопоставляем частоту монитора и телевизора.
    Option "MetaModes" "1024x768,640x480; 1024x769,640x480; 800x600,
640x480; 640x480,640x480"

    # Показываем, что подключили TV, а не второй монитор.
    Option "ConnectedMonitor"  "crt,tv"
EndSection
```

3. Перезапускаем X Window.

Вот, собственно и все, можно смотреть фильмы в формате AVI или MPEG4 прямо на экране телевизора.

Цифровые фотокамеры

В последнее время расширяется использование цифровых фотоаппаратов для фотосъемок. Несомненным плюсом этой технологии является быстрое перемещение фотографий на компьютер, минуя стадии печати фотоизображений и их сканирования. Для работы с цифровыми фотоаппаратами существует программа xcam, входящая в комплект пакета SANE.

Спутниковый Интернет

Года два назад энтузиасты FIDO и интернет-сообщества бурно обсуждали животрепещущую тему — Интернет через спутниковый канал, без ограничений, за баснословно низкую цену — что-то около 15 долларов. Все это можно было получить от фирмы Europe Online. Потом с аналогичным проектом возникло небезызвестное NTV. А всего-то надо купить тарелку, конвертор, немного кабеля и карту типа SkyStar1. И, конечно, установить драйверы для этой карты. Давайте посмотрим, реально ли это? Оказывается — вполне реально. Чтобы ознакомиться с этой темой — зайдите на сайт **www.gs.ru** в раздел спутникового Интернета и походите по ссылкам. Там же есть несколько статей, поясняющих, каким образом установить и настроить карту типа SkyStar1.

Ссылки

- **fero.koli.kando.hu/rivatv/** — описание настройки TV-out для видеокарт на чипах nVidia.
- **ftp://ftp.cs.unm.edu/mirrors/kde/unstable/apps/utils/** — утилита kpsion для связи с КПК Psion.
- **ftp://ftp.to.com/pub/psion/** — утилиты plptools для связи с КПК Psion.
- **ftp://ryeham.ee.ryerson.ca/pub/PalmOS/** — местонахождение утилиты Pilot-Link.
- **gazette.linux.ru.net/lg62/articles/rus-silva.html** — Anderson Silva. Видеоприложения на вашем Linux. Перевод Дмитрия Попкова.
- **huizen.dds.nl/~frodol/psiconv/** — официальная страница утилиты Psiconv.
- **jpilot.org** — сайт проекта J-Pilot.
- **linuxtv.org** — сайт, посвященный телевидению и Linux.
- **palm.opennet.ru/base/X/tv_out.txt.html** — пример настройки видеокарты nVidia с TV-out ((linux tv video).
- **www.cadsoft.de/people/kls/vdr/index.htm** — организация Video Disk Recorder на базе компьютера, платы SkyStar1 и Linux.
- **www.gnokii.org** — официальная страница проекта Gnokii.
- **www.gnome.org/projects/gnome-pilot/** — официальная страница пакета gnome-pilot.
- **www.in-berlin.de/User/kraxel/xawtv.html** — официальная страница программы xawtv.
- **www.linux.opennet.ru/base/X/video_out.txt.html** — пример TwinView для nVidia-карт, редактирование и запись видео (linux tv video).
- **www.linuxdvb.tv** — сайт, посвященный драйверам для карт спутникового телевидения.
- **www.lirc.org** — страничка проекта LIRC (Linux Infrared Remote Control).
- **www.mainconcept.com** — сайт программного обеспечения для редактирования видео.
- **www.mathematik.uni-kl.de/~wenk/kwintv** — официальная страница программы kWinTV.
- **www.medsyn.fr/perso/g.delafond/psilin/psiolinu.htm** — страница пакета PsiLin — программы для связи с КПК Psion.
- **www.nvidia.com** — сайт фирмы nVidia.
- **www.real.com** — Real Producer Basic.

- **www.slac.com/pilone/kpilot_home/** — официальная страница пакета KPilot.
- **www.strusel007.de/Linux/bttv/** — драйверы для чипов BT8XX.
- **www.stud.uni-hamburg.de/users/lennart/projects/atitvout/** — описание настройки TV-out и программного обеспечения для видеокарт на чипах ATI.
- **www.student.uwa.edu.au/~wliang** — официальная страница программы wmtv.
- **www.thp.uni-koeln.de/~rjkm/linux/bttv.html** — драйверы bttv.
- **http://linux.webclub.ru/adm/palm_pilot.html** — Вячеслав Калошин. Линукс и PalmPilot.
- Соответствующие HOWTO *(см. гл. 13)*:
  - bttv mini-HOWTO — Владимир Бормотов, Алексей Дец;
  - Linux and Psion HOWTO.

Глава 38

Сосуществование операционных систем

Как бы мы ни старались, а полностью жить в операционной системе Linux в современном мире не получается. Так сложилась жизнь, что множество программ написаны под операционные системы MS Windows или DOS. И зачастую по тем или иным причинам эти программы незаменимы. Поэтому данную проблему необходимо решать — вариант "или-или" нам не подходит.

Существует тривиальное решение этого вопроса — установить на компьютер две или три операционные системы и перезапускать компьютер, когда необходимо поработать в Windows NT. Но наверняка такое решение не понравится никому — неоптимальное расходование дискового пространства, постоянные перезагрузки компьютера и, как следствие, непроизводительные затраты времени. Поэтому желательно решить эту проблему по-другому.

Сосуществование двух или более операционных систем необходимо рассматривать комплексно. Проблемы, возникающие при взаимодействии разных операционных систем, можно разбить на следующие виды.

1. *Передача файлов с одной операционной системы в другую.* Это условие решается созданием протоколов передачи файлов и информации, которыми на данном этапе являются протоколы Интернета.
2. *Возможность работать с дисками и данными другой операционной системы.* Это условие неплохо решено для операционной системы Linux — она поддерживает файловые системы FAT, VFAT, NTFS и т. п. Со стороны операционных систем производства Microsoft все выглядит намного хуже — поддерживаются только файловые системы, разработанные Microsoft.
3. *Возможность выполнять программы, созданные для другой операционной системы.* Для решения этой задачи есть два подхода — разработка виртуальных машин и создание эмуляторов операционной системы:
   - *виртуальные машины* позволяют создать внутри операционной системы "виртуальный" компьютер, на котором и выполняется альтернативная операционная система и ее приложения. Для виртуальной машины все равно, какая операционная система будет установлена внутри нее, по-

скольку она обеспечивает псевдокомпьютер, на который и устанавливается операционная система. У этого подхода есть недостатки — мы вынуждены устанавливать на виртуальную машину альтернативную операционную систему и программное обеспечение, что не всегда возможно с точки зрения ресурсов и финансов. Несомненным достоинством виртуальной машины является полная эмуляция компьютера, и как следствие — возможность нормально запускать альтернативную операционную систему;

- *эмуляторы* призваны создать для программ альтернативной операционной системы "нормальную среду обитания". Для сложной операционной системы практически невозможно создать нормально функционирующий эмулятор, особенно для закрытых коммерческих операционных систем.

4. *Возможность работать с файлами, созданными в форматах программ другой операционной системы.* По большому счету это условие выполняется либо тогда, когда выпускается эквивалентное программное обеспечение для другой операционной системы, либо тогда, когда существует доступная документация на формат файлов, либо когда данные очень нужны. В принципе, четвертое условие можно решить, решив третье.

Как видите — список небольшой, но охватывающий множество проблем сосуществования операционных систем.

Эмуляторы

Начнем описание с эмуляторов, поскольку исторически в операционной системе Linux они появились раньше, чем виртуальные машины. Так что, пойдем от простого к сложному.

DOSEmu

Эмулятор однозадачной, однопользовательской операционной системы MS-DOS. Вы скажете, что в эпоху развитой Windows эмулятор MS-DOS не актуален — и будете неправы. Еще много программ, написанных под MS-DOS, находится в эксплуатации. Различные учетные, складские программы, программы отделов кадров и тому подобные АРМ, спокойно трудятся на своих рабочих местах. Достаточно много есть и хороших игр, написанных под MS-DOS, к примеру WarCraft II, Doom и Dune II. В свое время много специфических аппаратно-программных комплексов было разработано под MS-DOS, устройства эти эксплуатируются и по сегодняшний день.

Установка пакета DOSEmu не представляет сложности, поскольку этот пакет обычно входит в состав дистрибутива операционной ситемы. Поэтому переходим сразу к конфигурированию этого эмулятора DOS.

Конфигурирование DOSEmu

Пакет DOSEmu не отличается особой оригинальностью — конфигурационный файл называется dosemu.conf и находится в каталоге /etc. Помимо этого, каждый пользователь может создать в своем домашнем каталоге файл .dosrc, в котором можно откорректировать некоторые настройки DOSEmu для данного пользователя. Плюс к этим возможностям, поведение эмулятора можно изменить, используя параметры запуска.

На самом деле все записи в файле в dosemu.conf — это просто переменные, которые в последующем используются в /var/lib/dosemu/global.conf и имеют вид:

```
$_xxx = (n)
```

или

```
$_zzz = "s"
```

Описание параметров конфигурации сгруппировано по исполняемым функциям.

Управление отладочной информацией

Для включения вывода отладочной информации DOSEmu необходимо в конфигурационный файл добавить следующую строку:

```
$_debug = "-a"
```

где строка содержит то, что обычно передается через ключ командной строки `'-D'`.

Отладочная информация будет выводиться в файл, определенный опциями `'-o file'` либо `'-O'` (в последнем случае выводит в stderr).

Основные параметры

- Разрешает или запрещает использование прерывания таймера INT08:

  ```
  $_timint = (on|off)
  ```

- Позволяет либо запрещает задачам DOS использовать математический сопроцессор:

  ```
  $_mathco = (on|off)
  ```

- Параметр определяет, какой тип процессора эмулировать:

  ```
  $_cpu = (80386)
  ```

  Можно установить тип процессора не выше существующего в компьютере. Разрешенные значения:

  - 80386;
  - 80486;
  - 80586.

- Параметр разрешает или запрещает DOSEmu использовать счетчик циклов Pentium для лучшей обработки временных интервалов:

```
$_rdtsc = (on)
```

- Для использования 'rdtsc' DOSEmu необходимо выставить точную тактовую частоту процессора. Обычно она определяется автоматически, но в случае ошибок можно задать ее явно.

```
$_cpuspeed = (166.666)
```

- Разрешает DOSEmu доступ к конфигурированию PCI устройств:

```
$_pci = (on)
```

- Следующие параметры позволяют задать распределение оперативной памяти, которая доступна для DOS:

```
$_xms = (1024)
$_ems = (1024)
$_ems_frame = (0xe000)
$_dpmi = (off)
$_dosmem = (640)
```

- Следующий параметр определяет стиль поведения DOSEmu по отношению к процессорному времени, используемому DOSEmu:

```
$_hogthreshold = (1) # 0 — максимум процессорного времени для DOSEMU
                     # 1 — максимум процессорного времени для Linux
                     # >1 чем больше, тем меньше процессорного времени
                     #    для DOSEMU
```

- В том случае, если на вашем компьютере установлено нестандартное оборудование, для которого отсутствует Linux-драйвер, но существует DOS-драйвер, часто необходимо разрешить использование соответствующего IRQ в DOS:

```
$_irqpassing = ""  # список номеров IRQ (2-15) для передачи DOS
```

- Следующий параметр определяет, каким образом будет использоваться встроенный динамик:

```
$_speaker = ""     # or "native" or "emulated"
```

- При помощи следующих параметров можно получить управление реальными портами компьютера, но с точки зрения безопасности этого делать ни в коем случае нельзя:

```
$_ports = ""  # список портов, например "0x1ce 0x1cf 0x238"
```

Терминалы

Этот раздел предназначен для DOSEmu, выполняемой на удаленном компьютере или в графическом терминале xterm.

- Определяет набор используемых шрифтов:

```
$_term_char_set = ""
```

- Разрешает использование цвета:

```
$_term_color = (on)
```

- Задает интервал между обновлениями экрана в 1/20 секунды:

```
$_term_updfreq = (4)
```

- Определяет символ ESC:

```
$_escchar = (30)
```

Установки клавиатуры

При запуске DOSEmu из консоли или X Window может понадобиться задать подходящую раскладку клавиатуры. Это делается либо выбором одной из внутренних таблиц клавиатуры, либо загрузкой внешней таблицы.

- Внутренняя таблица клавиатуры определяется параметром:

```
$_layout = "name"
```

- Используется для сосуществования с X Window, поскольку по умолчанию устанавливается нейтральная (US) клавиатура:

```
$_X_keycode = (on)
```

- Следующий параметр позволяет получить прямой доступ к клавиатуре для DOS-программ. Обычно это необходимо для игр.

```
$_rawkeyboard = (1)
```

- Следующая переменная используется для улучшенной обработки прерывания клавиатуры:

```
$_keybint = (on)
```

Поддержка X Window

Для запуска DOSEmu в собственном окне X Window необходимо установить некоторые переменные, приведенные ниже.

- Задает интервал обновления изображения в 1/20 секунды:

```
$_X_updfreq = (5)
```

- Определяет заголовок окна программы:

```
$_X_title = "DOS in a BOX"
```

- Определяет текст значка:

```
$_X_icon_name = "xdos"
```

- Разрешение трансляции клавиатурных кодов через таблицы DOSEmu:

```
$_X_keycode = (off)
```

- Параметр задает частоту мерцания курсора:

```
$_X_blinkrate = (8)
```

- Задает тип шрифта для DOS-программы:

```
$_X_font = ""
```

- Параметр разрешает использование разделяемой памяти:

```
$_X_mitshm = (on)
```

- Использование системной палитры:

```
$_X_sharecmap = (off)
```

- Параметр разрешает пропорциональное изменение размеров окна:

```
$_X_fixed_aspect = (on)
```

- Разрешает использовать отношение сторон окна 4:3 в графике:

```
$_X_aspect_43 = (on)
```

- Параметр задает начальные размеры окна:

```
$_X_winsize = ""
```

- Параметр задает коэффициент гамма-коррекции:

```
$_X_gamma = (1.0)
```

- Задает размер фрейм-буфера для эмуляции VGA в килобайтах:

```
$_X_vgaemu_memsize = (1024)
```

- Параметр разрешает использовать линейный фрейм-буфер для VESA-режимов:

```
$_X_lfb = (on)
```

Видеоустановки для консоли

За конфигурирование DOSEmu для работы в консольном режиме отвечают следующие параметры.

- Этот параметр позволяет выбрать тип видеокарты:

```
$_video = "vga"
```

- Разрешает или запрещает использование видео на консоли:

```
$_console = (0)
```

- Параметр разрешает использовать BIOS-карты для установки видеорежима:

```
$_graphics = (0)
```

- Параметр разрешает доступ к видеопорту в графических режимах:

```
$_videoportaccess = (1)
```

- С помощью этого параметра задается адрес видео-BIOS:

```
$_vbios_seg = (0xc000)
```

- С помощью этого параметра указывается размер видео-BIOS:

```
$_vbios_size = (0x10000)
```

- С помощью этого параметра задается размер буфера регенерации:

```
$_vmemsize = (1024)
```

- С помощью этого параметра можно указать чипсет видеокарты для лучшего взаимодействия с видеокартой:

```
$_chipset = ""
```

Диски и дискеты

Следующие переменные определяют наличие дисковода, а так же параметры жесткого диска.

- Параметр используется для задания имени файла виртуальной дискеты, с которой будет производиться загрузка:

```
$_vbootfloppy = ""
```

- Параметр определяет тип и наличие дисковода A:

```
$_floppy_a ="threeinch"
```

- Параметр определяет тип и наличие дисковода B:

```
$_floppy_b = ""
```

- Этот параметр задает имя файла, содержащего список образов жесткого диска в /var/lib/dosemu:

```
$_hdimage = "hdimage.first"
```

При установке DOSEmu в файл /var/lib/dosemu/hdimage.first записывается образ загрузочного диска. Это файл, содержащий виртуальный образ файловой системы DOS — FAT.

Альтернативой загрузки с виртуального диска может служить загрузка с виртуальной дискеты, которая создается командой

```
'dd if=/dev/fd0 of=floppy_image'
```

Если это загрузочная дискета DOS, то при установке следующего параметра

```
$_vbootfloppy = "floppy_image"
```

будет загружаться с этой виртуальной дискеты.

COM-порты

Нижеприведенные параметры используются DOSEmu для задания параметров COM-портов и устройств, которые их используют.

- Параметр определяет, какое устройство Linux соответствует порту COM1:

```
$_com1 = "/dev/mouse"
```

- Параметр определяет, какое устройство Linux соответствует порту COM2:

```
$_com2 = "/dev/modem"
```

- Параметр определяет тип используемой мыши:

```
$_mouse = "microsoft"
```

- Параметр задает драйвер мыши:

```
$_mouse_dev = "/dev/mouse"
```

- С помощью этого параметра можно установить специальные управляющие флаги:

```
$_mouse_flags = ""
```

- Параметр задает скорость обмена информацией с мышью, 0 — не устанавливать:

```
$_mouse_baud = (0)
```

Принтеры

Принтер эмулируется передачей печатаемых данных на обычный Linux-принтер. С помощью следующих параметров указывают DOSEmu, какой из принтеров использовать.

- Параметр определяет имя Linux-принтера, который будет называться LPT1:

```
$_printer = "lp"
```

- Параметр задает задержку перед началом печати:

```
$_printer_timeout = (20)
```

Работа с сетью IPX/SPX

Следующие параметры используются для поддержки сетевого протокола IPX/SPX, при этом ядро операционной системы должно быть сконфигурировано с поддержкой протокола IPX.

- Параметр разрешает использование протокола IPX/SPX:

```
$_ipxsupport = (on)
```

- Параметр используется в том случае, если вы примените драйвер dosnet:

```
$_vnet = (on)
```

Звук

Для поддержки звуковой карты DOSEmu средствами звуковой подсистемы Linux необходимо установить следующие параметры.

- Параметр разрешает или запрещает поддержку звука:

```
$_sound = (off)
```

- Параметр определяет базовый адрес портов ввода/вывода звуковой карты:

```
$_sb_base = (0x220)
```

- Параметр определяет прерывание, используемое звуковой картой:

```
$_sb_irq = (5)
```

- Параметр определяет канал DMA, используемый звуковой картой:

```
$_sb_dma = (1)
```

- Параметр определяет используемое звуковое устройство:

```
$_sb_dsp = "/dev/dsp"
```

- Параметр определяет используемый микшер:

```
$_sb_mixer = "/dev/mixer"
```

- Параметр определяет базовый адрес MPU-401:

```
$_mpu_base = "0x330"
```

Приложения DEXE

Непосредственно исполняемые DOS-приложения DOSEmu (DEXE) — достаточно оригинальная концепция. На самом деле — это загружаемый образ диска, содержащий одно DOS-приложение. Достоинства такого типа приложений — они имеют доступ только к образу диска, и как следствие — порождают меньше проблем с безопасностью. Помимо этого — вам не надо делать инсталляцию DOS-приложения и настраивать его.

Для создания приложения формата DEXE нужно:

- пакет mtools;
- скомпилированный DOSEmu;
- zip-архив, содержащий все файлы, относящиеся к DOS-приложению;
- подготовить следующую информацию перед запуском mkdexe:
  - размер раздела для образа диска;
  - версию DOS, которую следует поместить на этот образ;
  - содержимое файлов Config.sys и Autoexec.bat.

После этого можно приступать к созданию приложения. Для этого необходимо зайти в систему как пользователь root и выполнить следующее:

```
mkdexe myapp.zip -x myapp.exe -o confirm
```

Если все прошло нормально, то у вас появится файл myapp.exe, который можно запустить на выполнение командой

```
dos -L myapp.exe [ dosemu-options ]
```

либо

```
dosexec myapp.exe [ dosemu-options ]
```

Wine

Wine (Wine Is Not an Emulator) — эмулятор операционной системы Windows разных версий. Позволяет запускать некоторые Windows-приложения под X Window.

К сожалению, больших успехов в запуске больших приложений типа игр или графических редакторов разработчики Wine пока не добились, однако запустить небольшие приложения можно. Программа интенсивно развивается, поэтому рекомендуется перед установкой получить самую свежую версию Wine с сайта разработчиков. Процесс установки подробно описан в документации и не представляет особого труда.

Для запуска приложения Windows необходимо в Xterm запустить Wine с параметрами командной строки. После простого запуска программы Wine без параметров появится строка формата запуска.

Самый простой вариант запуска программы, написанной для Windows, — набрать следующую строку:

```
wine имя_программы.exe
```

Можно указать при запуске, для какой версии Windows написана запускаемая программа. Например:

```
wine winver win98 имя_программы.exe
```

Если программа требует использования каких-либо библиотек, их подключение также можно задать в строке запуска, например:

```
wine winver win95 dll a.dll b.dll c.dll имя_программы.exe
```

WineX

WineX — проект, основанный на коде Wine. Коммерческая попытка довести до ума проект Wine, причем основной целью разработчиков является запуск игр, написанных для Windows. Как заявляют разработчики — на сегодняш-

ний день под WineX запускается более 80 наиболее популярных игр. Проект коммерческий, но для домашнего использования его можно загрузить бесплатно. В инсталляции и использовании мало чем отличается от Wine.

Виртуальные машины

Те, кто в компьютерной индустрии давно, наверняка помнят Систему Виртуальных Машин (СВМ), которая была очень распространена на больших ЭВМ серии ЕС (ЕС 1033/1066 — советский аналог IBM 360/370). Идеи живучи, и для Linux также была создана СВМ, которая с успехом эксплуатируется и получила достаточно широкое распространение.

VMWare

VMWare — это коммерческий продукт, позволяющий запускать на одной машине одновременно несколько операционных систем. Программу можно скачать с сайта производителя и пользоваться ей в тестовых целях в течение месяца.

Установка

Для установки VMWare необходимо скачать rpm-пакет для вашего дистрибутива с сайта разработчика. Установить VMWare можно только от пользователя root. После установки надо запустить vmware-config.pl — скрипт, помогающий настроить VMWare.

Для каждой операционной системы, запускаемой под VMWare, следует создавать свою конфигурацию. Для этого необходимо запустить на выполнение файл /usr/bin/vmware. После проверки видеорежима возникнет окно выбора конфигурации VMWare.

Режим **Run Configuration Wizard** предназначен для создания и быстрой и простой настройки новой виртуальной машины. Режим **Run Configuration Editor** предназначен для создания и детальной настройки новой виртуальной машины. Режим **Open An Existing Configuration** предназначен для выбора уже созданной виртуальной машины.

При создании виртуальной машины необходимо выбрать тип устанавливаемой на виртуальной машине операционной системы и каталог, где будут располагаться все файлы новой виртуальной машины. После этого нужно выбрать тип жесткого диска — виртуальный или физический диск, установленный на вашем компьютере.

Далее производится разрешение использования CD-ROM для виртуальной машины и дисковода.

После этого нужно настроить поддержку сети для виртуальной машины — полное ее отсутствие, использование настроек реальной сети или эмуляция сети средствами VMWare.

С помощью Configuration Editor можно произвести тонкую настройку уже созданной виртуальной машины.

Установка Windows 98 с помощью WMWare

Запустите WMWare, выберите созданную вами ранее виртуальную машину. После того как виртуальная машина выбрана, ее необходимо включить. Для этого нажмите кнопку **Power On** на панели инструментов VMWare или выберите команду меню **Power | Power On**. После включения виртуальной машины вы увидите эмуляцию настоящего BIOS, в настройки которого можно зайти с помощью клавиши <F2>.

В настройках BIOS необходимо в разделе **Boot** установить порядок просмотра устройств в поисках загрузчика. Поставьте первым **ATAPI CD-ROM Drive** — для установки Windows с загрузочного CD-ROM. Для выхода из BIOS с сохранением изменений необходимо нажать клавишу <F10>.

После успешной загрузки необходимо разбить виртуальный жесткий диск с помощью программы fdisk и отформатировать его. После этого приступайте к установке Windows 98.

После установки Windows 98 необходимо сконфигурировать ее. Единственным сложным местом в настройке Windows 98 под VMWare является конфигурирование видеокарты.

Для этого надо установить VMWare Tools. VMWare Tools состоит из двух компонентов — драйвера видеоадаптера, работающего под VMWare, и программы, которая позволяет менять параметры виртуальной машины прямо из загруженной под ней операционной системы.

Для установки VMWare Tools необходимо в VMWare выбрать команду меню **Settings | VMWare Tools Install...**. При этом в вашем дисководе должна быть установлена псевдодискета с VMWare Tools. Далее, в Windows 98 открываем диск A: и видим на нем запускаемый файл VMWare Tools, который и надо установить. После установки будет открыто диалоговое окно настройки экрана и инструкция по установке драйвера видеоадаптера. Далее действуйте по инструкции.

Пожалуй, это все о VMWare — система очень надежна, позволяет устанавливать множество операционных систем на одном компьютере и, что самое интересное, — эти операционные системы могут быть одновременно запущены и даже обмениваться информацией.

Win4Lin

Еще один эмулятор виртуального компьютера, но, в отличие от VMWare, он создан и оптимизирован специально для запуска Windows в Linux. Для своей работы требует внесения изменений в код операционной системе Linux. Благодаря этому он быстрее и более надежен, чем VMWare. Кроме того, Win4Lin позволяет также организовать полнофункциональную DOS-сессию. Единственный недостаток — отсутствие нормальной поддержки DirectX.

Сама Windows запускается из-под X Window в окне. Также вы имеете возможность предоставить доступ к любому разделу на винчестере, даже к каталогам операционной системы Linux.

Получить Win4Lin следует с сайта производителя, находящегося по адресу **www.netraverse.com**. Для этого необходимо зайти в раздел Members, где надо бесплатно зарегистрироваться, после чего на ваш электронный адрес будет выслано письмо с вашим логином и паролем. Только после получения пароля вы сможете скачать с сайта нужную программу. Программа-инсталлятор определяет версию дистрибутива, библиотек, установленного ядра операционной системы и предлагает загрузить нужные для вашей системы подправленное ядро Linux и, собственно, сам пакет Win4Lin.

На том же сайте надо получить пробную лицензию на Win4Lin сроком на 30 дней. Далее, устанавливаем новое ядро операционной системы. После переустановки ядра устанавливаем пакет Win4Lin. После его установки необходимо произвести инсталляцию Windows 98. В каталоге /var/win4lin/publicbin есть утилита installwindows, которую следует запустить и указать ей, где брать инсталляцию Windows. После инсталляции необходимо воспользоваться программой winsetup для того, чтобы настроить устройства и разделы жесткого диска для использования Windows. Помимо этого, можно указать каталоги, которые будут видны в Windows как диски.

Все! После установки Windows 98 набираем в командной строке `win` и получаем окно, в котором выполняется Windows 98.

Ссылки

- **linuxbegin.by.ru/articles/article17.shtml** — запуск Windows-программ в Linux.
- **www.linux-ve.chat.ru** — виртуальная библиотека Linux.
- **linux.yaroslavl.ru/Docum/Other/dosemu/README.html** — документация по DOSEmu v. 0.97 pl. 3.0. Перевод Валерия Груздева.
- **www.suse.com/~dosemu/** — домашняя страница DOSEmu.

- **www.osp.ru/os/2001/07-08/023.htm** — Виктор Костромин. Две системы на одном компьютере.
- **www.winehq.org** — официальный сайт проекта Wine.
- **www.vmware.org** — официальный сайт проекта VMWare.
- **dhls.agava.ru/vmware.html** — Ерижоков А. А. Использование VMWare.
- **www.softerra.ru/freeos/16294/print.html** — Александр Куприн. VMWare Workstation 3.0 — песочница для взрослых.
- **www.netraverse.com** — сайт производителя Win4Lin.
- **www.linux.hitech.by X-Stranger** — Win4Lin — Windows из-под Linux.
- **t37.nevod.perm.su/linux/tune/dosemu.html** — В. Вислобоков. Как установить и настроить DOSEmu.
- **www.mgul.ac.ru/~t-alex/Linux/howto.mine/howto.mine.2.htm** — эмуляция других сред. MINI-NOWTO.

Глава 39

Мультимедиа

О программах и устройствах, необходимых для работы в операционной системе Linux, мы говорили на протяжении всей книги, теперь пришла пора немного развлечься. Для комфортной работы никогда не мешает немного отдохнуть. А современный отдых при помощи компьютера можно обозначить одним емким словом — мультимедиа.

Точного определения мультимедиа так никто и не сформулировал, мы же под этим подразумеваем звук и видео во всех их проявлениях.

Настройка звуковой карты

Начнем со звука. Современные дистрибутивы знают о большинстве звуковых карт и при инсталляции дистрибутива практически всегда корректно их устанавливают. Если у вас, все же, возникли проблемы со звуковой картой, не огорчайтесь — эти проблемы решаемы. Во-первых, сходите на сайт производителя дистрибутива — вдруг о проблеме с вашим типом звуковой карты известно производителю дистрибутива, и он описал решение проблемы или выложил обновленные драйверы. Во-вторых, можно сходить на сайт **www.alsa-project.org** — сайт разработчиков драйверов для звуковых карт. Почти наверняка для вашей звуковой карты там есть свежий драйвер. В документации на драйвер есть описание процесса компиляции, установки и настройки драйвера.

В том случае, если при установке операционной системы в вашем компьютере отсутствовала звуковая карта, и вы ее установили позже, процедура конфигурации будет следующая.

Если вы самостоятельно перекомпилировали ядро операционной системы Linux, вам необходимо убедиться в том, что при компиляции была включена поддержка звуковых карт и, в частности, вашей звуковой карты. Если это не так — вам необходимо пересобрать ядро операционной системы Linux.

Если же у вас ядро операционной системы после установки операционной системы осталось нетронутым — можете не беспокоиться — в ядре, идущем в дистрибутиве, включена поддержка всех звуковых карт.

Затем мы должны в консольном режиме запустить утилиту sndconfig от пользователя root. Эта утилита входит в дистрибутивы Red Hat, Mandrake, Altlinux, ASPLinux. Она выдаст сообщение об обнаруженной в системе звуковой карте (рис. 39.1), затем задаст несколько простых вопросов и в конце настройки выдаст тестовый звук.

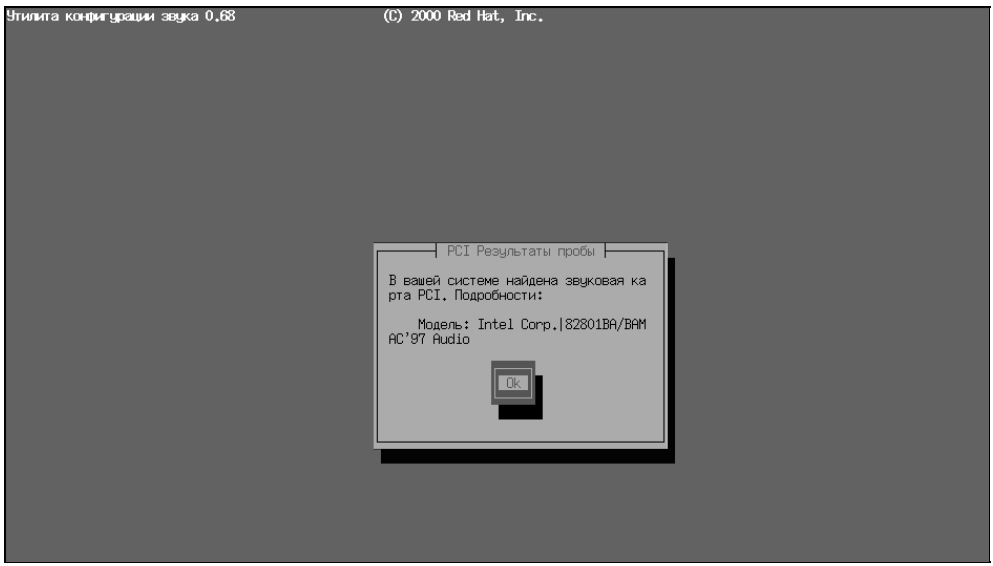

Рис. 39.1. Утилита конфигурирования звуковой карты sndconfig

Консольные утилиты для работы со звуком

Чтобы не обижать тех, кто привык работать с консольным режимом, начнем с них. Первое, что нам необходимо сделать — добраться до утилит регулирования громкости звука. По традиции эти утилиты называются *микшерами* и зачастую в своем названии содержат это слово. К примеру — aumix (рис. 39.2).

Помимо этой утилиты можно использовать еще с ряд других микшеров, к примеру alsamixer, xmix.

Управлять громкостью звука мы уже умеем. Как правило, первое, что приходит в голову — попытаться воспроизвести Audio CD. Нет ничего легче. Если на вашем CD-ROM есть кнопка воспроизведения — вставляем диск и нажимаем ее. Если такой кнопки не существует — воспользуемся какой-нибудь подходящей программой. Первое, что приходит в голову — cdplay.

Простая утилита, минимум функциональности, но для прослушивания компакт-дисков в консольном режиме вполне подходит. Если же вас не удовлетворяют ее возможности — можно воспользоваться другой программой — cdp. В отличие от предыдущей утилиты — вы можете более комфортно прослушивать музыку — управлять громкостью, последовательностью воспроизведения музыкальных треков, произвести смену компакт-диска.

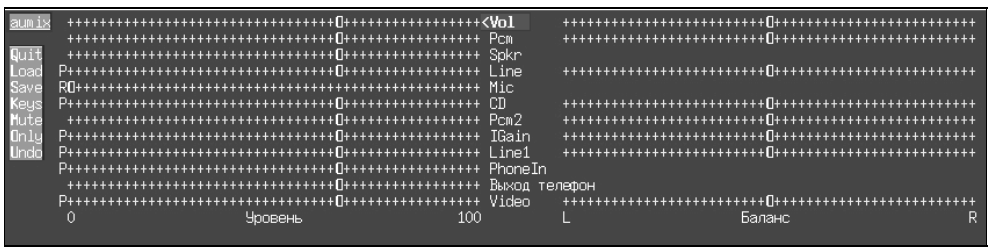

Рис. 39.2. Утилита управления громкостью звука aumix

Следующее, что приходит в голову — каким-то образом воспроизвести четырехгигабайтный архив музыки в формате MP3. Для этого существует отличная программа — mpg123. За полным описанием возможностей программы mpg123 мы рекомендуем обратиться к документации на нее. Вкратце поясним, что в командной строке необходимо задать путь к воспроизводимому файлу или к каталогу с маской файла *, и все, что находится в этом файле, будет воспроизведено. Можно также создать свой список воспроизведения (play-list) или включить режим воспроизведения музыкальных файлов в произвольном порядке. Для воспроизведения файлов MP3 служат также и программы blaster и splay.

Помимо воспроизведения файлов формата MP3 вам может понадобиться воспроизвести музыкальные файлы и других форматов. Утилита wavplay предназначена для воспроизведения музыкальных файлов формата WAV, программа playmidi — для воспроизведения музыкальных файлов MIDI, а утилита tracker — для воспроизведения файлов формата MOD. В том случае, если вам захочется преобразовать музыкальный файл из одного формата в другой — воспользуйтесь программой Sox.

Все, о чем писалось выше, касалось готовых музыкальных файлов. Но вот вам принесли новый музыкальный компакт-диск, и вы хотите сохранить его содержимое в файлах формата MP3. Для этого сначала воспользуемся программой, носящей имя cdparanoia. Программа проста в использовании — вставляем компакт-диск и запускаем cdparanoia, причем в качестве первого аргумента указываем номер музыкального трека, а в качестве второго — имя выходного файла формата WAV.

Программа cdparanoia имеет большое количество опций командной строки, о которых можно узнать из документации к программе. Самая интересная для нас опция командной строки `-B`, позволяющая оцифровать сразу все дорожки компакт-диска.

Однако после работы программы cdparanoia мы получаем просто wav-файл. А нам необходимо было получить файлы формата MP3. Различных программ преобразования полученных wav-файлов в формат MP3 достаточно много, поэтому обратим ваше внимание только на одну — lame. Эта утилита очень популярна среди пользователей, позволяет выбирать алгоритм сжатия файла, частоту дискретизации и многое другое. Однако в самом простом случае достаточно только указания двух аргументов — имени исходного wav-файла и целевого файла формата MP3. Процесс оцифровки компакт-диска и создания файла формата MP3 можно совместить, для этого достаточно выполнить следующую команду:

```
cdparanoia 1 | lame – my_music.mp3
```

В том случае, если вас не устраивает такой процесс получения mp3-файлов, можно воспользоваться программой mp3c (рис. 39.3).

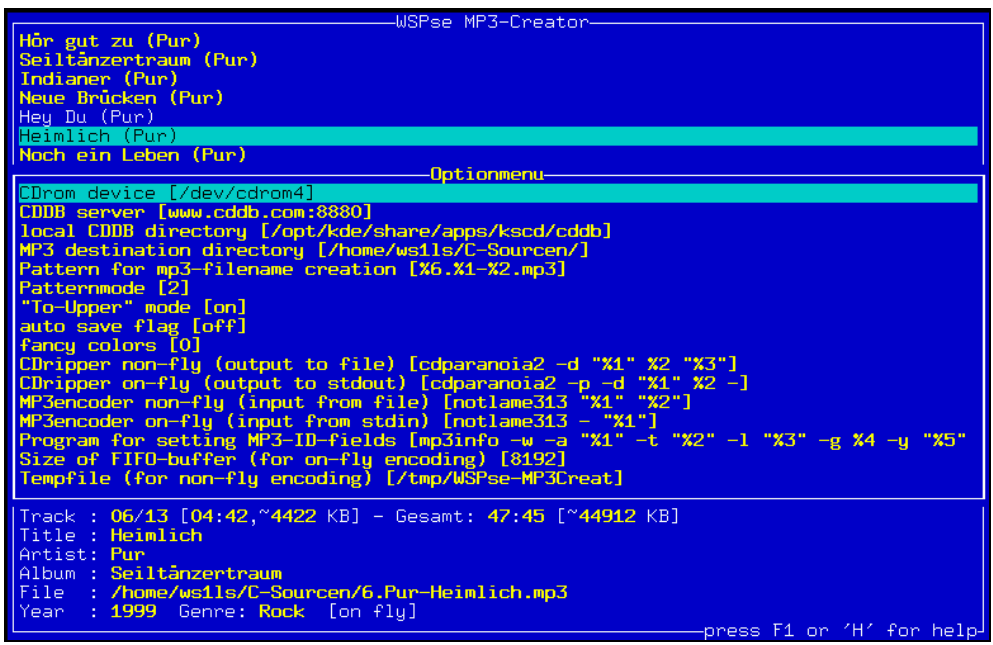

Рис. 39.3. Программа оцифровки компакт-диска в формат MP3

Это просто программа-оболочка, которая для выполнения своих функций вызывает различные программы оцифровки компакт-дисков и утилиты-конверторы.

Программа mp3c имеет ряд дополнительных возможностей — заполнение полей автора, названия песни, имени альбома и другой информации, которая может содержаться в файле формата MP3.

Звук в X Window

Рассмотрев вопросы консольного режима, перейдем в графический, в X Window. Здесь еще больше разнообразия. Возьмем для примера KDE и GNOME. В этих оболочках программы мультимедиа сгруппированы, соответственно, в пункте меню мультимедиа. Посмотрим, что же есть в этом разделе в среде GNOME.

Выбор достаточный. На рис. 39.4 вы видите программу, очень похожую на программу Звукозапись в Windows. То же назначение — запись с микрофона. Простой интерфейс — начать запись, остановить, продолжить и сохранить файл.

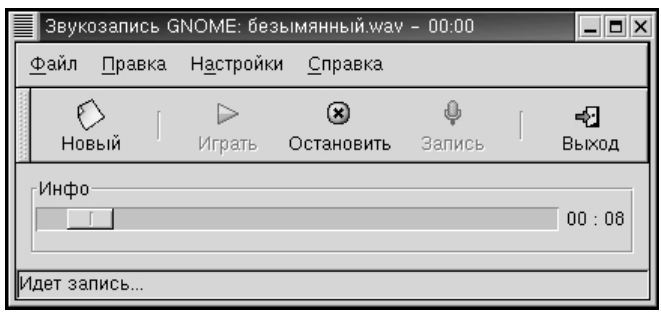

Рис. 39.4. Программа звукозаписи с микрофона

Пойдем далее. На рис. 39.5 вы видите микшер звуковой платы и проигрыватель компакт-дисков.

Программа xmms (рис. 39.6). Практически полный аналог winamp. Великолепно воспроизводит mp3-файлы, позволяет создавать play-list. Помимо всего прочего, существует возможность самостоятельно изменять внешний вид программы в широких пределах.

А теперь перейдем к мультимедиа-программам, входящим в стандартную поставку KDE. Набор программ напоминает по составу набор программ из GNOME, однако он несколько больше. Бегло просмотрим состав мультимедиа-программ.

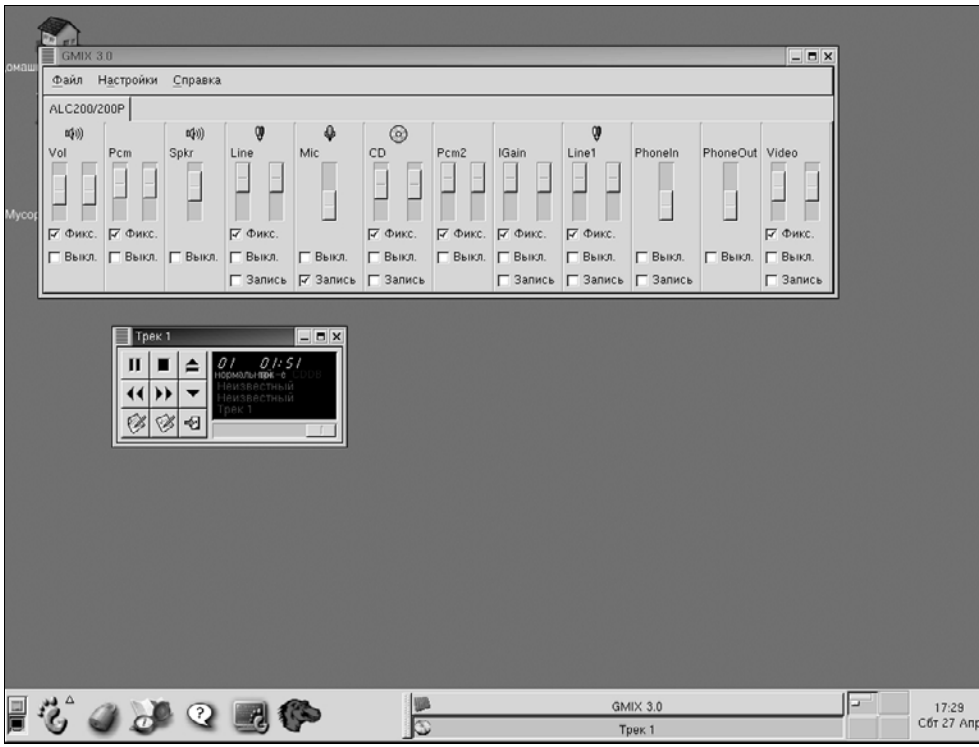

Рис. 39.5. Микшер звуковой платы и программа для воспроизведения компакт-дисков

Рис. 39.6. Программа xmms

Программа KMid (рис. 39.7) представляет собой проигрыватель midi-файлов. Позволяет выбирать инструменты, редактировать файл, менять темп воспроизведения музыки и некоторые другие параметры.

Аналогичная по назначению программа, только имеющая больше возможностей, называется KMidi. Внешний вид программы представлен на рис. 39.8.

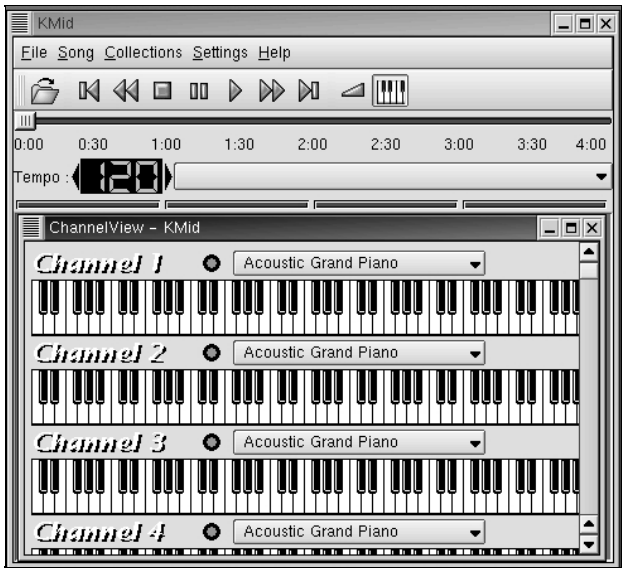

Рис. 39.7. Программа KMid

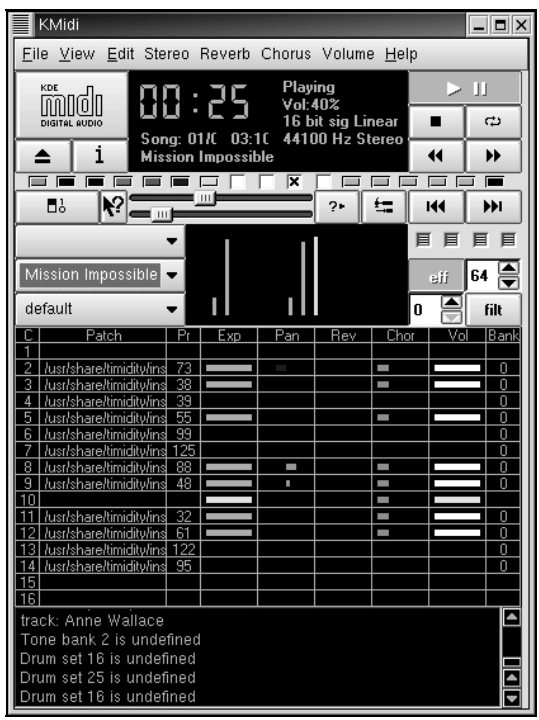

Рис. 39.8. Программа KMidi

Программа для воспроизведения компакт-дисков также входит в состав программ мультимедиа KDE (рис. 39.9).

Рис. 39.9. Программа CD-проигрыватель

И, наконец микшер, регулирование громкости воспроизведения звука (рис. 39.10).

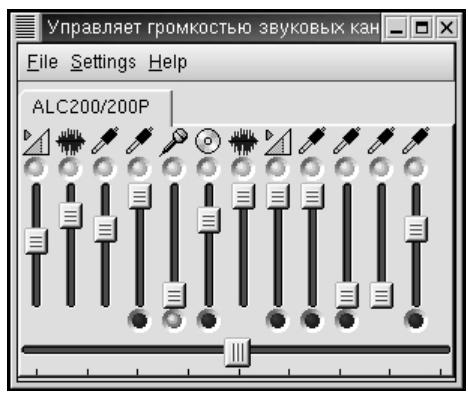

Рис. 39.10. Микшер звуковой платы

Помимо вышеприведенных программ, можно найти и установить большое количество дополнительного программного обеспечения для работы со звуком на все случаи жизни. Для примера можно взять программный пакет festival. Если у вас бессонница, а самому читать себе сказки на ночь не хочется, воспользуйтесь этой программой — она вам вслух почитает сказки. Правда, пока еще нет полноценной поддержки русского языка, но в качестве вполне приемлемой альтернативы можно воспользоваться при воспроизведении текстов языком испанским — звучание получается вполне приемлемого качества. А если вам вдруг захотелось попытаться заставить свой компьютер понимать человеческую речь — фирма IBM распространяет бесплатную систему распознавания речи ViaVoice.

Видео в Linux

На компьютере под управлением операционной системы Linux возможно и кино посмотреть. Для этой цели существует несколько различных программ, позволяющих воспроизводить видеофайлы различных форматов. На некоторых из них мы остановимся подробней.

Пожалуй, одним из старейших форматов хранения видео на компьютере является формат MPEG. Хорошая программа для воспроизведения такого типа файлов — smpeg. Понятный интерфейс, проста в управлении, не требовательна к ресурсам компьютера. На базе программы smpeg созданы следующие программные пакеты:

- Enjoympeg — MPEG-проигрыватель;
- dumpmpeg — простая программа для захвата кадров из mpeg-фильмов;
- XMPS — полнофункциональный MPEG-проигрыватель с поддержкой play-list и изменением внешнего вида;
- ZZPlayer — MPEG-проигрыватель для KDE;
- Xtheater — программа для воспроизведения Video CD.

Но главное не это. Почти все диски с видеофильмами для воспроизведения на компьютере, которые сейчас доступны у нас, — закодированы с использованием формата MPEG4 (DivX). Соответственно, нам нужна программа, которая могла бы воспроизводить эти файлы. Давайте посмотрим, что нам могут предложить разработчики программного обеспечения.

К сожалению, большинство программ используют кодек сжатия, предназначенный для Windows, поэтому желательно иметь самый свежий кодек. Получить его можно на сайте DivX (**www.divx.com**). Помимо этого, возьмите библиотеку avifile (**avifile.sourceforge.net**), которая позволяет использовать Windows AVI-кодеки (Indeo, Video, DivX) в операционной системе Linux.

Программа XMPS

Программа XMPS — полнофункциональный MPEG-проигрыватель с поддержкой play-list и изменением внешнего вида (рис. 39.11).

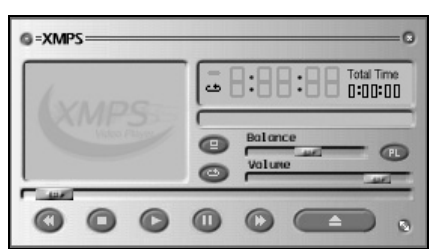

Рис. 39.11. Программа XMPS

Программа avifile-player

Программа avifile-player (рис. 39.12). Поскольку программа использует Win32-библиотеки, то можно смотреть не только DivX, но и avi-файлы, кодированные другими Windows-кодеками.

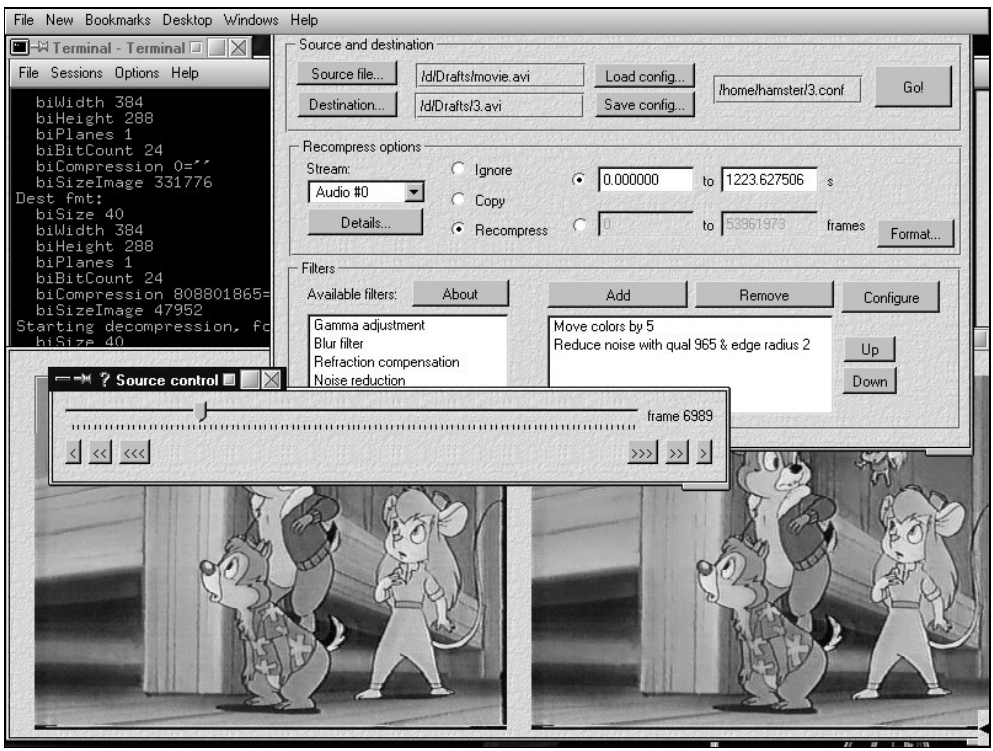

Рис. 39.12. Проигрыватель avifile-player

Программа xmms

Об этой программе мы упоминали выше, когда рассматривали программное обеспечение для воспроизведения mp3-файлов. Однако при установке соответствующих plug-ins программа может воспроизводить и видеофайлы. Для этого необходимо загрузить xmms-avi и установить avifile-библиотеку и avi-кодеки Win32. В результате получается хорошая программа для воспроизведения видеофильмов, правда достаточно сильно загружающая центральный процессор.

Программа XMMP — LinuX MultiMedia Player

Программа XMMP — LinuX MultiMedia Player. Использует avifile-библиотеку и avi-кодеки Win32. Проект новый, очень динамично развивающийся, поэтому достаточно много возможностей заявлено, но до конца не реализовано.

По замыслу автора проекта, XMMP (рис. 39.13) должен представлять собой центр по обработке видео. Это программное обеспечение должно проигрывать видео, создавать, редактировать и конвертировать файлы мультимедиа.

Хорошее качество воспроизведения, но сильно загружает центральный процессор.

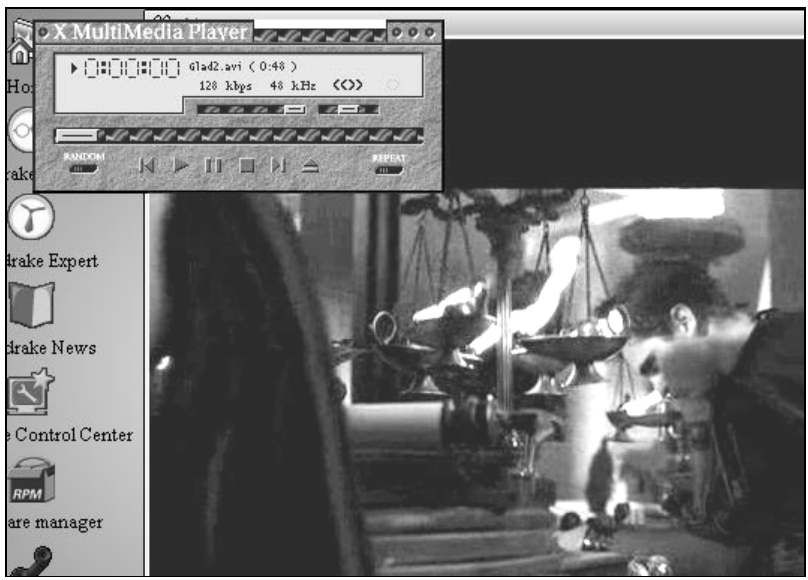

Рис. 39.13. Проигрыватель XMMP

Программа MPlayer

Программа MPlayer — Movie Player for Linux (рис. 39.14).

Эта программа на сегодняшний день является безусловным лидером в мире Linux. Очень удобна в использовании, многофункциональна, большое количество настраиваемых параметров, очень качественное воспроизведение. Можно изменять внешний вид программы на свое усмотрение (рис. 39.15). И при всем этом — центральный процессор при воспроизведении используется всего лишь на 15—20%. Но главное, программу можно скомпилировать для исполь-

зования в графическом интерфейсе и для использования в консольном режиме. Помимо всго этого, если у вас есть соответствующее оборудование, можно управлять проигрывателем при помощи дистанционного пульта управления. Программа очень динамично развивается, поэтому ходите на сайт разработчика хоть раз в месяц — наверняка найдете что-то новое.

Рис. 39.14. Проигрыватель MPlayer в одном обличье

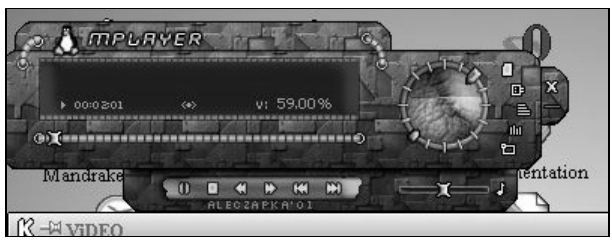

Рис. 39.15. Проигрыватель MPlayer в другом обличье

Программа XINE

Еще один проигрыватель видеофайлов (рис. 39.16).

Рис. 39.16. Проигрыватель XINE

Проигрыватель поддерживает большое количество видеокодеков:

- MPEG1;
- MPEG2;
- MPEG4;
- DivX;
- motion JPEG;
- AVI (использует win32-кодеки: Indeo 3.1-5.0, cinepak, Window Media 7/8).

Помимо воспроизведения видео, XINE умеет работать и с аудиофайлами:

- MPEG audio layer 1;
- MPEG audio layer 2;
- MPEG audio layer 3;
- a/52 (ac3, dolby digital);
- dts;
- vorbis;
- pcm;
- DivX audio.

Несколько нестандартное управление, умеет изменять внешний вид, не сильно загружает центральный процессор.

Вот, пожалуй, и все о мультимедиа.

Ссылки

- **avifile.sourceforge.net** — сайт avifile, позволяет использовать avi-кодеки Windows (Indeo, Video, DivX).
- **divx.euro.ru** — сайт видеопроигрывателя avifile-player.
- **mplayer.sourceforge.net** — сайт программы Mplayer (Movie Player for Linux).
- **people.freenet.de/for_Ki/** — сайт проигрывателя Enjoympeg.
- **sourceforge.net/projects/dumpmpeg** — сайт программы захвата видеокадров dumpmpeg.
- **www.chez.com/tsc/zzplayer/zzplayer.html** — сайт KDE MPEG-проигрывателя ZZPlayer.
- **www.divx.com** — сайт оригинального DivX-кодека.
- **www.frozenproductions.com/xmmp** — сайт проекта XMMP (LinuX MultiMedia Player).

- **www.linuxjournal.com/article.php?sid=4382** — Adam Williams. Issue 81: Movie Making on a Linux Box? No Way!
- **www.linuxoid.ru/how_to/DivX.html** — Гвоздев Андрей. DivX в Linux.
- **www.lokigames.com/development/smpeg.php3** — официальная страница проигрывателя smpeg.
- **www.opendivx.org** — сайт кодека DivX с открытым исходным кодом.
- **www.softerra.ru/freeos/13036/** — Алексей Федорчук. Консольное мультимедиа.
- **www.softerra.ru/freeos/14906/** — Алексей Федорчук. Как граббить награбленное.
- **xine.sourceforge.net** — сайт программы XINE.
- **xmms.org** — сайт программы XMMS (воспроизведение аудио и видео).
- **xmps.sourceforge.net** — сайт видеопроигрывателя XMPS.
- **xtheater.sourceforge.net** — сайт программы для воспроизведения Video CD Xtheater.
- **http://gazette.linux.ru.net/lg63/articles/rus-andreiana.html** — Marius Andreiana. Линукс на вашем рабочем столе: Мультимедиа. Перевод Александра Михайлова.

Глава 40

Действия при нештатных ситуациях

Эта глава посвящена возможным решениям проблем, возникающих с операционной системой либо с жестким диском. Большая часть таких ситуаций должна учитываться при составлении планов восстановления. Как правило, эти проблемы возникают неожиданно, и иметь соответствующие навыки по их решению необходимо, поскольку по известным законам неприятность возникает именно тогда, когда "упавшая" система нужна всем и немедленно. Большая часть материала этой главы основывается на книге "Системное администрирование Linux" *(см. приложение 5)*.

Утрата пароля root

Обычно эта ситуация встречается при смене администратора либо когда в фирме работают несколько администраторов, а сопровождение системы поставлено плохо. Иногда такая неприятность случается, когда кто-то редактирует файл /etc/passwd вручную, а в это время отключают электроэнергию. Пароль, наконец, можно просто забыть.

Для решения этой проблемы существуют несколько вариантов, которые мы сейчас и рассмотрим.

Восстановление без перезагрузки

Если на одной из виртуальных консолей сохранился открытым сеанс пользователя root (работать под пользователем root, вообще-то, вопиющее нарушение трудовой дисциплины и безопасности системы), восстановить пароль root проще простого — можно воспользоваться командой `passwd`. Как вариант — отредактировать файлы passwd и shadow таким образом, чтобы пользователя root пускали без пароля и опять-таки воспользоваться командой `passwd`.

Перезагрузка в однопользовательском режиме

Для доступа к системным файлам можно перегрузить систему в режиме одиночного пользователя.

Для безопасного выполнения перезагрузки с системной консоли достаточно нажать комбинацию клавиш <Ctrl>+<Alt>+<Delete>. Однако не везде операционная система настроена таким образом, что позволяет любому пользователю перегрузить операционную систему. Также можно попробовать команды `reboot`, `poweroff`, `halt`, однако на серверах обычно операционная система настраивается таким образом, что перегрузить ее может либо пользователь root, либо специально выделенный пользователь.

При компиляции ядра в него иногда включают средства поддержки функции Magic SysRq, предназначенной для принудительного выполнения перезагрузки компьютера или ввода команд `sync` и `unmount`.

Ниже перечислены комбинации клавиш, которые удобно использовать для обращения к данной функции.

- <Alt>+<SysRq>+<s> — предпринимается попытка выполнить команду `sync` для всех смонтированных файловых систем;
- <Alt>+<SysRq>+<u> — последовательно выполняются команды `unmount` и `remount` в режиме "только для чтения";
- <Alt>+<SysRq>+<b> — выполняется перезагрузка системы.

Если перезагрузить систему с помощью упомянутых корректных методов все же не удастся, постарайтесь минимизировать разрушения файловых систем, которые могут возникнуть в связи с аварийным выключением системы. Лучше всего такие действия производить тогда, когда для восстановления операционной системы у вас будет достаточно много времени, например по окончании рабочего дня или на выходные. При аварийном выключении компьютера файловые системы не будут корректно демонтированы, и после перезагрузки системы автоматически запустится программа fsck. На сервере с большим объемом дискового пространства выполнение программы может занять несколько часов.

После перезагрузки компьютера нам необходимо загрузить операционную систему в однопользовательском режиме. Как правило, для этого достаточно указать параметр `single`. При использовании в качестве загрузчика операционной системы программы LILO в командную строку достаточно ввести следующую команду:

```
LILO: linux single
```

После этого операционная система должна загрузиться в однопользовательском режиме. Однако иногда операционная система настраивается так, что для перевода операционной системы в однопользовательский режим необ-

ходимо указывать пароль пользователя root. Если операционная система настроена именно так, то при выполнении команды `init` она потребует ввести соответствующий пароль. Для решения этой проблемы можно воспользоваться следующим вариантом запуска операционной системы Linux:

```
LILO: linux init=/bin/sh
```

В результате нормальное выполнение загрузки операционной системы будет отменено, лишь только запустится командная оболочка.

В вышеприведенных примерах подразумевается, что используется пункт меню программы LILO, имеющий название linux. Узнать, какие, собственно, варианты загрузки существуют на вашем компьютере можно в тот момент, когда программа LILO ожидает ввода команды. Список доступных для загрузки ядер можно вывести с помощью клавиши <Tab>.

Помимо вышеописанных сложностей, вполне может оказаться, что ваш загрузчик операционной системы (LILO в данном случае) настроен на использование паролей, значения которых указываются в файле /etc/lilo.conf. Этот режим применяется для запроса специальных паролей в случае загрузки определенных конфигураций системы.

В этом случае без так называемой rescue-дискеты делать нам нечего. В дистрибутиве Red Hat Linux инсталляционный диск содержит вариант использования компакт-диска в качестве rescue-дискеты.

Восстановление пароля root после перезагрузки

После успешной загрузки выполните команду `mount` для корневой файловой системы (в том случае, если вы производили загрузку операционной системы с дискеты или CD-ROM) или заново смонтируйте ее в режиме доступа "для чтения/записи" (в случае обычной загрузки в режиме single или init=/bin/sh).

Для монтирования корневой файловой системы в режиме доступа "для чтения/записи" можно воспользоваться приведенной ниже командой:

```
mount -o remount,rw /
```

Если схема размещения файловых систем вам незнакома, можно с помощью команды `fdisk -l` вывести список существующих дисковых разделов. Монтируя по очереди дисковые разделы, выясните, на каком именно разделе находится каталог /etc.

Теперь можно удалить прежний пароль пользователя root путем редактирования файлов /etc/passwd или /etc/shadow. Однако иногда удобней изменить пароль, просто скопировав его из записи пользователя, пароль которого достоверно известен.

Если в системе применяется PAM или подобный ему пакет, настроенный на предупреждение использования пустых паролей, скопируйте хэшированное значение пароля, соответствующее некоторому известному вам паролю.

Если корневая файловая система жесткого диска была смонтирована в каталоге /mnt (при загрузке системы с аварийной дискеты), то с помощью команды `chroot` можно указать, с какой именно копией файла passwd должны работать такие команды, как `passwd`.

После того как корневая файловая система будет смонтирована в режиме доступа "для чтения/записи", с помощью команды `chroot` можно сделать ее корневой для всех процессов оболочки. В этом случае для установки нового значения пароля можно просто воспользоваться командой `passwd`. Данный вариант обеспечивает автоматическое корректное обновление любых файлов, используемых средствами синхронизированной аутентификации.

Если файл /etc/passwd был разрушен, скопируйте его с любой машины, имеющей сходные параметры настройки, или с аварийной дискеты, а еще лучше — извлеките этот файл из резервной копии системы.

Устранение последствий атак хакеров

Это большая и сложная тема. Действия, которые следует предпринимать после различных нарушений защиты системы, описывались в главе, посвященной безопасности системы *(см. гл. 7)*.

Вкратце алгоритм решения проблемы, вызванной атакой, приведен ниже.

1. Отключить компьютер от сети как Интернет, так и от локальной, поскольку зачастую целью атаки является либо получение информации, находящейся на атакованном компьютере, либо использование взломанного компьютера в качестве плацдарма для организации атак на другие компьютеры сети.
2. Переустановить с нуля операционную систему, безжалостно уничтожив всю информацию на жестком диске. В принципе, если вам позволяют время и возможности — желательно установить, каким образом был произведен взлом операционной системы. Для этого необходимо действовать двумя путями:
   - на основании log-файлов операционной системы попытаться выяснить, когда и каким образом произошел взлом;
   - просмотреть на сайте производителя дистрибутива (в разделе bagtraq или в рассылках по безопасности) информацию о всех найденных со времени выхода дистрибутива (или обновления вами соответствующих пакетов операционной системы) брешах в безопасности используемых вами пакетов. При обнаружении в пакетах различных "дыр", как пра-

вило, патчи этих пакетов выходят достаточно оперативно — обычно это занимает день-два.

Замечание

Как правило, если взломщик достаточно квалифицированный, самое большее, что вы сможете понять на основании log-файлов (да и то приблизительно) — время взлома системы, поскольку "нормальные" взломщики подчищают log-файлы или вообще их удаляют. Если вы хотите все-таки докопаться до сути — необходимо настроить ведение log-файлов таким образом, чтобы их копии передавались по сети на отдельный компьютер, на котором будут храниться все log-файлы вашей сети.

3. Восстановить файлы с данными и параметрами настройки взломанной машины. Крайне не рекомендуется восстанавливать систему либо исполняемые файлы из резервной копии, поскольку вполне вероятно, что в последних резервных копиях находится уже взломанная система, а взломщики обычно "подсаживают" во взломанную операционную систему модифицированные специальным образом исполняемые программы.

   Файлы настройки, содержимое которых должно отличаться от полученных во время установки операционной системы, необходимо тщательно проверить, поскольку многие файлы настройки содержат команды сценариев и различных программ, которые могут использоваться хакерами для организации атак с целью получения доступа к системе. Лучший вариант — делать копии всех файлов настройки сразу же после их создания, а также после внесения в них любых изменений, причем копировать их как на носитель информации, так и распечатывать на бумаге.

4. После переустановки и восстановления файлов настройки обязательно установить все обновления, касающиеся критически важных пакетов операционной системы.

Проблемы с загрузкой операционной системы

В процессе загрузки операционной системы принимают участие несколько различных программных компонентов, поэтому существует несколько этапов, на каждом их которых могут возникать разнообразные проблемы. Будем решать проблемы по мере их возникновения.

Останов загрузки в процессе выполнения LILO

Во избежание появления большинства проблем, связанных с программой LILO, после изменения конфигурационного файла lilo.conf, а также после установки нового ядра операционной системы или изменения конфигура-

ции жестких дисков, всегда выполняйте команду `lilo`. Обязательно обращайте внимание на сообщения об ошибках или предупреждениях, которые программа LILO генерирует в процессе обновления главной загрузочной записи или карты расположения используемых при загрузке файлов.

Создайте загрузочную дискету, которая будет использоваться в случаях разрушения загрузочных карт LILO.

Сохраняйте копию файла /etc/lilo.conf, копии главных загрузочных записей (MBR) на специальной аварийной дискете, а лучше — на компакт-диске. Постоянно поддерживайте эти копии в актуальном состоянии, отражающем все вносимые в систему изменения.

Далее описаны ситуации, которые могут возникнуть при ненормальном функционировании программы LILO.

Программа LILO выводит последовательность 01010101010

Ядро и карты загрузки находятся на устройстве, которое не поддерживается средствами BIOS.

Программа LILO останавливается, выдав L

Первичный загрузчик LILO не может найти вторичный загрузчик этой программы. Кроме того, могут возвращаться коды ошибок жесткого диска, описанные в документе /usr/doc/packages/lilo. Обычно эта ситуация означает, что программы BIOS и LILO по-разному определяют геометрию жесткого диска.

В случае возникновения такого рода проблемы проверьте, чтобы в BIOS был установлен режим LBA (если у вас жесткий диск достаточно больших размеров), и поместите директиву `linear` в файл /etc/lilo.conf (либо удалите ее, если она там уже была), после чего выполните команду `lilo` до повторной перезагрузки системы.

Проверьте, не превосходят ли размеры дисковых разделов максимальный размер раздела, поддерживаемый программами BIOS.

Обычно такого рода проблемы возникали на материнских платах для процессоров i486, Pentium и ранних материнских платах с поддержкой Pentium II.

Программа LILO останавливается, выдав LI

Вторичный загрузчик программы LILO был найден, но его не удалось корректно загрузить. Такое обычно случается при наличии проблемы с геометрией диска. Если реальный файл boot и его описание в карте загрузки не соответствуют друг другу, выполните команду `lilo` и перезагрузите компьютер.

Программа LILO останавливается, выдав LIL?

Вторичный загрузчик программы LILO не смог получить доступ к требуемому ему адресу. Проблема устраняется так же, как и в предыдущем случае.

Программа LILO останавливается, выдав LIL

Вторичный загрузчик программы LILO не смог прочитать системную карту. Выполните команду `lilo` и перезагрузите компьютер.

Программа LILO останавливается, выдав LIL-

Некорректная таблица дескрипторов системной карты. Обычно это означает, что файл /boot/map был разрушен или перемещен.

Проблемы с выполнением программы LILO

В процессе выполнения программы LILO могут встречаться самые различные проблемы. Рассмотрим наиболее распространенные из них.

Неверная сигнатура LILO

```
First boot sector doesn't have a valid LILO signature
```

Приведенное сообщение об ошибке обычно означает, что разрушен файл /boot/boot. Другой вариант — в файле /etc/lilo.conf директива `install` указывает на объект, который программа LILO не воспринимает как программу первоначальной загрузки.

```
Chain loader doesn't have a valid LILO signature
```

Приведенное сообщение об ошибке означает, что разрушен файл /boot/chain. Другой вариант — в файле /etc/lilo.conf директива `loader=` указывает на объект, который программа LILO не воспринимает как загрузчик цепочки.

Загрузка ядра Linux выполняется программой вторичного загрузчика boot.b, которая загружается в память программой первичного загрузчика, расположенного в главной загрузочной записи жесткого диска. Все имеющиеся в файле lilo.conf директивы `image=` обрабатываются исключительно вторичным загрузчиком.

Программа загрузки chain используется для загрузки MS-DOS и других операционных систем подобного типа. Она вызывается при загрузке версий операционных систем, задаваемых в файле lilo.conf директивами `other=`. Кроме того, для специфических вариантов загрузки машины могут использоваться особые программы первоначальной загрузки, описываемые директивой программы LILO `loader=`.

BIOS не имеет доступа к жесткому диску

Иногда ваш компьютер может выдать следующее сообщение:

```
Warning: BIOS drive 0x82 may not be accessible
```

Оно появляется в тех случаях, когда некоторые из указанных в файле lilo.conf вариантов загрузки системы ссылаются на программы и операционные системы, расположенные на жестком диске, отличном от первых двух устройств, подключенных к первичному контроллеру. То есть параметры в файле lilo.conf требуют выполнить загрузку с третьего устройства первичного контроллера или с устройства, подключенного к вторичному контроллеру.

Приведенное выше сообщение программы LILO является просто предупреждением. В случае его появления никакого вреда устанавливаемой системе нанесено не будет.

Повреждение главной загрузочной записи (MBR)

Если главная загрузочная запись жесткого диска или таблица разделов повреждена, ее можно восстановить. Как правило, это не вызывает никаких повреждений в размещенных на этом жестком диске файловых системах или данных.

Для этого необходимо предварительно сохранить копию главной загрузочной записи и таблицы разделов на резервную дискету. Копию главной загрузочной записи и таблицы разделов можно сделать следующей командой:

```
dd if=/dev/hda of= hda-mbr.bin bs=512 count=l
```

Здесь:

- `/dev/hda` — представляет собой ссылку на первый жесткий диск с интерфейсом IDE;
- файл `hda-mbr.bin` — это тот самый файл, содержимое которого и будет являться главной загрузочной записью жесткого диска;
- размер блока устанавливается равным 512 байтам;
- параметру `count` присвоено значение 1, поскольку требуется скопировать только один сектор данных.

Программа первичного загрузчика содержится в главной загрузочной записи только первичного жесткого диска (/dev/hda или /dev/sda). На всех остальных дисковых устройствах в этой записи будет содержаться только таблица разделов, а оставшаяся ее часть будет пустой. В любом случае полезно сохранять сведения о таблицах разделов всех жестких дисков компьютера.

Для восстановления главной загрузочной записи жесткого диска достаточно использовать следующую команду:

```
dd of=/dev/hda if=$BACKUP_FILE bs=512 count=l
```

Помимо использования команды `dd` и резервной копии главной загрузочной записи жесткого диска, таблицу разделов можно восстановить вручную, воспользовавшись информацией, предварительно сохраненной или распечатанной с помощью команды `fdisk -l`. Полезно сохранить копию этих данных непосредственно в процессе исходной установки системы. В результате, вы всегда будете знать местонахождение этих данных, независимо от последующих перемещений системы.

Понятно, что когда потребуется воспользоваться созданной копией главной загрузочной записи жесткого диска, саму операционную систему загрузить не удастся. Поэтому копии всех резервных файлов главной загрузочной записи жесткого диска должны быть помещены на аварийные дискеты и, может быть, в корневую файловую систему вашего компьютера.

Поскольку полученный с помощью команды `dd` файл является двоичным, вы не можете без особых ухищрений получить из него информацию о разбиении жесткого диска. Поэтому желательно хранить также текстовый вариант списка всех разделов диска (с указанием их размеров и расположения), причем еще и в распечатанном виде, предназначенном для чтения человеком. Этот список легко может быть получен с помощью команд:

```
fdisk -l
```

или

```
mount
```

В результате, даже при отсутствии резервной копии главной загрузочной записи жесткого диска всегда можно будет восстановить таблицу разделов вручную. Восстановить вручную текст программы первоначальной загрузки можно с помощью программы LILO. Для этого достаточно выполнить команду `/sbin/lilo` с указанием корректного файла параметров /etc/lilo.conf.

Новое ядро операционной системы не загружается

Для обновления карты размещения используемых при загрузке файлов необходимо выполнить программу LILO. В этой карте содержится информация о точном расположении на жестком диске каждого из файлов, которые программа LILO использует в процессе загрузки системы. В число этих файлов входит и файл ядра операционной системы.

Если до остановки процесса загрузки операционной системы система выводит сообщение **Loading Kernel...**, то, возможно, при компиляции ядра неверно были выбраны параметры компиляции. В этом случае для определения и последующего устранения проблемы попробуйте воспользоваться параметрами ядра `reserve=` и `exclude=`, которые можно задавать в командной строке программы LILO.

Новое ядро выдает сообщение о превышении размера ядра

Некоторое время назад ядро операционной системы было компактным само по себе. В последние годы из-за серьезного увеличения функциональности ядра операционной системы загрузить несжатое ядро программа-загрузчик не в состоянии. Так что если вы пользуетесь устаревшими рекомендациями по компиляции — ваше новое ядро получится больше, чем может загрузить программа-загрузчик. Поэтому почитайте рекомендации по компиляции ядра на сайте фирмы — производителя дистрибутива. Обычно после компиляции используется создание из полученного ядра операционной системы его сжатого образа, который после загрузки распаковывается в оперативной памяти компьютера.

Кроме того, можно создать ядро меньших размеров за счет перемещения большего количества необходимых функций в отдельные загрузочные модули и отказа от вкомпилирования в ядро поддержки устройств, не установленных на вашем компьютере.

Ядро выдает сообщение о невозможности монтирования корневого каталога

В ядре Linux определено устанавливаемое по умолчанию устройство и раздел, на котором располагается корневая файловая система. Это значение, задаваемое прямо в исходном тексте ядра, можно изменять с помощью команды `rdev`. Существует еще несколько подобных значений по умолчанию, которые жестко записываются в ядро в процессе его компиляции. Их также можно изменять с помощью различных параметров команды `rdev`, что позволяет избежать перекомпиляции ядра.

Если существующее имя корневой файловой системы не соответствует значению, установленному в ядре, то при попытке ее монтирования будет выдано упомянутое выше сообщение. Самый простой способ изменить записанное в ядре и принимаемое по умолчанию значение — указать требуемое имя в параметре `root=`.

После успешной загрузки системы выполните команду `rdev` и/или модифицируйте файл /etc/lilo. conf, чтобы добавить в него директиву, например `append="root=hda2"`.

Помимо вышеприведенного случая, такое сообщение можно получить, если при компиляции ядра операционной системы драйверы устройства, на котором размещается корневая файловая система, не были вкомпилированы в ядро или были вынесены в загружаемый модуль, а поскольку загружаемые модули ядра грузятся с подмонтированного жесткого диска — эта проблема и возникает.

Экран мерцает и на нем отсутствует приглашение к регистрации в системе

Если рабочая станция настроена на использование при загрузке графического приглашения для регистрации пользователя, и на экране монитора заметны повторяющиеся безрезультатные попытки системы начать процедуру регистрации, проверьте состояние мыши.

Вначале проверьте, подключена ли она к компьютеру. Затем вручную перезагрузите систему в режиме одного пользователя и убедитесь, что в каталоге /dev имеется соответствующий файл устройства. Затем попробуйте выполнить команду `gpm` — это позволит убедиться, что система знает о существовании мыши и может с ней работать. В противном случае вручную запустите команду `startx` и проанализируйте выводимые сообщения об ошибках. Проверьте состояние используемых системой X Window файлов настройки.

Другая проблема, не позволяющая системе X Window нормально начать работу, может заключаться в отсутствии доступа к каталогу со шрифтами — локальному или расположенному на некотором сервере.

Если все упомянутые условия выполнены, то ошибка может заключаться в неверной настройке X Window — либо установлен не тот тип видеокарты, либо завышены частоты монитора.

Проблемы с запуском программ

В этом разделе рассматриваются вопросы устранения проблем, возникающих при попытке запуска различных программ. Обычно такого плана проблемы возникают при неверно установленных правах доступа или отсутствующих системных библиотеках, необходимых данной программе.

Повреждение или удаление разделяемых библиотек

В случае повреждения разделяемых библиотек операционную систему, как правило, можно будет перезагрузить только с помощью аварийной загрузочной дискеты.

Поскольку работа всех компонентов операционной системы Linux полностью зависит от разделяемых библиотек, в случае их отсутствия или повреждения ни одну из обычных команд и утилит выполнить невозможно. В последних версиях Linux лишь очень небольшое количество программ связано с библиотеками статически. Именно по этой причине стандарт File Hierarchy Standard (Стандарт иерархии размещения файлов) требует, чтобы каталог /lib находился непосредственно в корневом каталоге, а также рекомендует избегать его использования в качестве точки монтирования.

Поскольку программы, используемые в нормальном процессе остановки системы, также могут быть динамически связаны с системными библиотеками, самым лучшим способом безопасной перезагрузки систем будет использование метода Magic SysRq, описанного ранее.

В противном случае потребуется перезагрузить машину с аварийной дискеты, после чего восстановить в системе корректные копии разделяемых библиотек.

Сообщение *"getcwd: cannot access parent directories"*

Это сообщение выводится в том случае, если некоторый процесс переходит в каталог с ограниченным доступом. Здесь этот процесс отменяет свои привилегии или вызывает функции `setuidO` или `setgidO` для объекта, который не имеет права доступа к одному из родительских каталогов, входящих в путь, ведущий в текущий рабочий каталог.

Как правило, в этом случае дочерний процесс, не имеющий необходимых привилегий, не может использовать команду `is` или даже команду `echo *`.

Чаще всего подобная ситуация возникает тогда, когда некоторым пользователям присвоены неверные права по отношению к каталогу, ведущему к их основному каталогу.

Программа вызывает SIG11

Если программа сообщила, что было вызвано прерывание SIG11 и получен дамп ядра, это обычно означает, что в вашей системе проблемы с оборудованием.

Обычно такого плана ошибки вызывают модули памяти, отдельные ячейки микросхем которых некорректно работают, причем эта проблема может не проявляться неделями. Реже подобную ошибку вызывает нестабильно работающая материнская плата.

Народное средство проверки нестабильной памяти — несколько раз подряд произвести компиляцию ядра операционной системы. Если попытка откомпилировать ядро операционной системы Linux завершится выдачей сообщения **Internal compiler error** со ссылкой на прерывание SIG11, причина, вероятнее всего, в ненадежной работе оперативной памяти.

К сожалению, в современных микросхемах оперативной памяти чрезвычайно трудно надежно выявить непостоянные отказы. Компьютеры и операционные системы настолько сложны, что простая последовательность операций "запись, чтение, проверка" в оперативной памяти едва ли будет пригодна для выявления проблем с оборудованием.

Если предполагается, что ошибка связана с оборудованием, попробуйте установить в компьютер другие модули памяти.

Превышение максимального количества открытых файлов

Ядро имеет ограничение, связанное с максимальным количеством одновременно открытых файлов, которое задается при компиляции ядра операционной системы. Достижение операционной системой этого предела приводит к тому, что операционная система отказывает в открытии файла.

Изменить текущее значение этого параметра можно отредактировав псевдофайлы /proc/sys/kernel/file-max и /proc/ sys/kernel/inode-max.

Например:

```
inode-max = 32768 file-max .=5.120
```

Два параметра системы — максимальное количество задач в системе и максимальное количество задач для одного пользователя — переопределяются при компиляции ядра. Используемые значения задаются в файле параметров ядра.

Проблемы с файловыми системами

Далее речь пойдет об устранении различных проблем, которые возникают при работе с файловыми системами.

Ошибка *"unable to find swap-space signature"*

Подобная ошибка может возникнуть в том случае, когда одно и то же дисковое пространство страниц виртуальной памяти используется одновременно несколькими операционными системами, либо была повреждена таблица swap-раздела.

При появлении такой ошибки необходимо воспользоваться командой `fdisk` для повторной проверки типов разделов, описанных в таблице разделов диска. Убедившись, что все выполненные для разделов назначения корректны, введите команду `mkswap`.

Переполнение файловой системы

Если пользователь заполнит все дисковое пространство, выделенное файловой системе, то за пользователем root резервируется некоторый свободный объем дискового пространства. Как справедливо предусмотрели разработчики файловой системы, администратору и некоторым утилитам необходимо наличие некоторого пустого дискового пространства для нормальной работы с переполненным разделом.

Разрешение на использование этого резервного пространства может быть предоставлено отдельному пользователю или группе пользователей при помощи утилиты tune2fs.

Очевидным решением этой проблемы является удаление некоторых файлов, либо архивирование редко используемых файлов.

В том случае, если пользователь root или процесс, запущенный с правами пользователя root, вызовет переполнение диска, начнется заполнение резервного пространства диска. По этой причине почта для пользователя root всегда должна посылаться на учетную запись, не имеющую особых привилегий, а ротация файлов журналов должна тщательно контролироваться.

Для предупреждения случаев переполнения файловых систем целесообразно использовать какую-либо программу мониторинга состояния операционной системы.

Переполнение числа блоков индекса файловой системы

Переполнение числа блоков индекса файловой системы возможно даже в том случае, когда основное пространство файловой системы еще не заполнено. Этот показатель не имеет отношения к параметру ядра, описывающему максимальное количество одновременно открытых блоков индексов. Если файловая система содержит большое количество файлов размером менее 4 Кбайт, то все блоки индекса такой файловой системы могут оказаться заполненными раньше, чем ее основное пространство.

Отношение количества блоков индекса к количеству блоков данных любой заданной файловой системы устанавливается при ее создании (параметр `-i` команды `mke2fs`). Файловые системы, предназначенные для размещения спула групп новостей, всегда должны иметь увеличенное отношение числа блоков индекса.

Подозрение на наличие сбойного кластера или сектора

В том случае, если вы заподозрили, что на вашем жестком диске появились сбойные кластеры, можно запустить утилиту для проверки жесткого диска на наличие сбойных секторов. Эту операцию необходимо производить в то время, когда никто не работает с компьютером, поскольку она может затянуться на достаточно длительное время.

Для выявления сбойных блоков и помещения сведений о них в соответствующий список файловой системы типа Ext2 можно использовать команду `e2fsck -c`.

При выполнении команды *mount* доступ к системе блокируется

В некоторых случаях выполняемый процесс может "зависнуть", если команда `mount` применяется к файловой системе на устройстве, не отвечающем на запросы системы. Кроме того, подобная ситуация иногда возникает при обращении к устройствам активной SCSI-цепочки, которые отсоединены или выключены.

Подобные ситуации могут происходить и при переключении на другие виртуальные консоли, регистрации через последовательные терминалы или соединения telnet и т. п. Если запустить утилиту ps, то подобные "подвешенные" процессы отмечаются как находящиеся в состоянии D. Выполнение для подобного процесса команды `kill -9` не оказывает на этот процесс никакого влияния, поскольку обработка сигналов блокируется на все время, пока процесс ожидает завершения выполнения подпрограммы системного вызова ядра операционной системы.

В подобном состоянии операционная система может находиться сколько угодно долго, причем она будет нормально функционировать до тех пор, пока не будет предпринята попытка обращения к "подвешенному" процессу или устройству. Чтобы выйти из этого положения, необходимо корректно завершить все процессы операционной системы (которые не находятся в "подвешенном" состоянии), после чего компьютер можно будет перезагрузить.

Случайное удаление файла

Если все ссылки на файл и все связанные с ним блоки обработки уже удалены, то после закрытия последнего открытого для него дескриптора занятое файлом пространство становится доступным для системного драйвера сборки мусора. Как только занимаемое ранее файлом пространство будет очищено этим драйвером, файл будет утрачен навсегда.

В состав Linux включен документ "Undelete HOWTO" и несколько редакторов шестнадцатеричных данных. В частности, программы ext2ed и debugfs предоставляют некоторые инструменты, которые могут оказаться полезными при устранении проблем подобного рода.

Так же можно воспользоваться программой mc (Midnight Commander). Для этого запускаем `mc` и в командной строке набираем `cd /#undel:/hda`. В результате получаем панель, в которой находится список удаленных файлов, причем имя файла — номер inode. Эти файлы можно просмотреть и, выбрав нужный, восстановить.

Разрушение данных

Команда `fsck` используется для проверки и восстановления файловых систем. Восстановленные блоки индекса помещаются в зарезервированный каталог lost+found, который существует в каждом физическом разделе Ext2.

В том случае, если резервной копии данных не существует, можно попробовать разобраться в каталоге lost+found и попытаться вручную восстановить данные.

Проблемы с сетью

В этом разделе рассматривается устранение проблем, которые возникают в случае некорректной настройки, неправильного функционирования или повреждения сети.

К системе нет доступа из сети

Проверьте значения параметров TCP, содержащихся в файлах /etc/hosts.allow и /etc/hosts.deny. Кроме того, проверьте все другие аспекты организации работы брандмауэра, которые применимы к данной машине. Проконтролируйте состояние сетевого кабеля в тех точках, в которых он подключается к машине и к остальной части сети.

Используйте утилиту ping для проверки функционирования сети.

Проблемы ввода/вывода данных

Во многих приложениях можно устанавливать комбинации клавиш, предназначенные для вызова специальных функций. Если проблема с вводом возникает только в одной программе (например, emacs), то назначить комбинации клавиш можно с помощью команд этого же приложения.

Любой текст воспроизводится в виде двоичных символов

Чаще всего подобная ситуация возникает при использовании простых утилит, предназначенных для чтения двоичных файлов. Терминал воспринимает одну или более двоичных комбинаций как команду изменения символического шрифта. В результате, прочесть выводимые на экран сообщения будет невозможно. Введите команду `reset`, не обращая внимания на то, что будет выведено на экран. В результате все параметры терминала будут приведены к значениям, принимаемым по умолчанию.

Система не реагирует на команды, вводимые с клавиатуры

Убедитесь, что клавиатура подключена к компьютеру правильно, а не, скажем, к порту мыши. Если доступ к машине через сеть все еще возможен, то с помощью команды `loadkeys -d` восстановите карту ключей клавиатуры, используемую в системе по умолчанию. В противном случае не избежать перезагрузки системы со всеми вытекающими последствиями.

Переназначение клавиш

Утилита xmodmap предоставляет средства переназначения клавиш клавиатуры. Однако внесенные изменения остаются в силе только на время сеанса X Window. Для изменения раскладки клавиатуры в сеансах работы с текстовой консолью следует использовать утилиту loadkeys.

Окно сеанса X Window не воспринимает команд с клавиатуры и сигналов мыши

В среде X Window был выдан запрос, захвативший фокус ввода. Если выдавшее его приложение или задача "зависнет", менеджер окон окажется заблокированным и любой направленный в среду X Window ввод будет игнорироваться.

Для решения этой проблемы необходимо получить доступ к компьютеру по сети либо через последовательный терминал и после этого выполнить команду `kill -9` для заблокированного задания. Если этого окажется недостаточно, продолжайте указанную процедуру, поднимаясь по соответствующему дереву процессов. В самом худшем случае остановка процесса X-сервера вынудит процесс init "собрать мусор" в его ресурсах и ресурсах всех порожденных им процессов. Как правило, "убиения" заблокированного процесса или его родителей бывает достаточно для разблокирования устройства ввода.

Прочие аварийные ситуации

Некоторые аварийные ситуации нельзя отнести к какой-нибудь конкретной категории. Об этих ситуациях мы и поговорим в данном разделе.

Не работает устройство, подключенное к параллельному порту

Параллельный порт в настоящее время является точкой подключения различных периферийных устройств: принтеров, сканеров, CD-RW, ZIP Drive

и многих других. В том случае, если в вашей операционной системе для устройств, подключаемых к параллельному порту, используются загружаемые модули, то вы должны с помощью команды `lsmod` проверить, соответствует ли загруженный модуль тому типу устройства, которое в данный момент подключено к параллельному порту.

Настоятельно рекомендуется не предпринимать попыток выгрузить модули до тех пор, пока не будет демонтирована файловая система, связанная с данным устройством

Работа системы кажется медленной, хотя объем оперативной памяти превосходит 64 Мбайт

Подобная проблема может быть связана с конструктивными недостатками определенных материнских плат, не способных кэшировать ячейки памяти, адреса которых расположены выше 64 Мбайт. Обычно такая проблема возникала с ранними платами для процессоров Pentium. Эта проблема особенно остро отражается на системе Linux, поскольку она распределяет доступную память в направлении сверху вниз, начиная с самых верхних адресов.

Возможные решения проблемы заключаются в замене материнской платы компьютера или в использовании параметра `mem=` ядра Linux для установки лимита используемой оперативной памяти на уровне 64 Мбайт, хотя в последнем случае вы теряете весь объем памяти выше 64 Мбайт.

После увеличения объема оперативной памяти система работает нестабильно

Некоторые материнские платы используют для собственных нужд небольшой блок ячеек оперативной памяти, расположенный у ее верхней границы. Попробуйте указать параметр ядра `mem=xxxM`, где значение `xxx` — на один мегабайт меньше полного объема установленной в компьютере оперативной памяти.

После увеличения объема оперативной памяти система не видит добавленную память

Некоторые материнские платы (в основном для Pentium и ранние платы для Pentium II) страдают подобным недостатком. Для исправления ситуации можно указать параметр ядра `mem=xxxM`, где значение `xxx` — полный объем

установленной оперативной памяти. Если операционная система покажет вам полный объем оперативной памяти, но будет вести себя нестабильно — воспользуйтесь предыдущим советом.

Ссылки

- **http://www.bitwizard.nl/sig11**— "SIG11 Problem". Описание проблемы SIG11 и пути ее решения.
- Соответствующие HOWTO *(см. гл. 13)*:
  - Multiboot Using LILO mini-HOWTO;
  - LILO mini-HOWTO.

Часть VII

Приложения

Приложение 1

Физическая структура файловой системы Ext2

Таблица П1.1. *Структура суперблока*

Название поля	Тип	Описание поля
s_inodes_count	ULONG	Число индексных дескрипторов в файловой системе
s_blocks_count	ULONG	Число блоков в файловой системе
s_r_blocks_count	ULONG	Число блоков, зарезервированных для пользователя root
s_free_blocks_count	ULONG	Счетчик числа свободных блоков
s_free_inodes_count	ULONG	Счетчик числа свободных индексных дескрипторов
s_first_data_block	ULONG	Первый блок, который содержит данные
s_log_block_size	ULONG	Индикатор размера логического блока: 0 = 1 Кбайт, 1 = 2 Кбайт, 2 = 4 Кбайт
s_log_frag_size	LONG	Индикатор размера фрагментов
s_blocks_per_group	ULONG	Число блоков в каждой группе блоков
s_frags_per_group	ULONG	Число фрагментов в каждой группе блоков
s_inodes_per_group	ULONG	Число индексных дескрипторов в каждой группе блоков
s_mtime	ULONG	Время, когда в последний раз была смонтирована файловая система
s_wtime	ULONG	Время, когда в последний раз производилась запись в файловую систему
s_mnt_count	USHORT	Счетчик числа монтирований файловой системы
s_max_mnt_count	SHORT	Число, определяющее, сколько раз может быть смонтирована файловая система без ее проверки

Таблица П1.1 *(окончание)*

Название поля	Тип	Описание поля
s_magic	USHORT	"Магическое число" (0xEF53), указывающее, что файловая система принадлежит к типу ex2fs
s_state	USHORT	Флаги, указывающие текущее состояние файловой системы
s_errors	USHORT	Флаги, задающие процедуры обработки сообщений об ошибках
s_pad	USHORT	Заполнение
s_lastcheck	ULONG	Время последней проверки файловой системы
s_checkinterval	ULONG	Максимальный период времени между проверками файловой системы
s_creator_os	ULONG	Указание на тип операционной системы, в которой создана файловая система
s_rev_level	ULONG	Версия (revision level) файловой системы
s_reserved	ULONG[235]	Дополнение до 1024 байт

Таблица П1.2. *Структура описания группы блоков*

Название поля	Тип	Описание поля
bg_block_bitmap	ULONG	Адрес блока, содержащего битовую карту блоков группы
bg_inode_bitmap	ULONG	Адрес блока, содержащего битовую карту индексных дескрипторов группы
bg_inode_table	ULONG	Адрес блока, содержащего таблицу индексных дескрипторов группы
bg_free_blocks_count	USHORT	Счетчик числа свободных блоков в группе
bg_free_inodes_count	USHORT	Число свободных индексных дескрипторов в группе
bg_used_dirs_count	USHORT	Число индексных дескрипторов в группе, которые являются каталогами
bg_pad	USHORT	Заполнение
bg_reserved	ULONG[3]	Зарезервировано

Таблица П1.3. *Структура индексного дескриптора файла*

Название поля	Тип	Описание поля
i_mode	USHORT	Тип и права доступа файла
i_uid	USHORT	Идентификатор владельца файла
i_size	ULONG	Размер файла в байтах
i_atime	ULONG	Время последнего обращения к файлу
i_ctime	ULONG	Время создания файла
i_mtime	ULONG	Время последней модификации файла
i_dtime	ULONG	Время удаления файла
i_gid	USHORT	Идентификатор группы, которой принадлежит файл
i_links_count	USHORT	Счетчик числа ссылок
i_blocks	ULONG	Число блоков, занимаемых файлом
i_flags	ULONG	Флаги файла
i_reserved1	ULONG	Зарезервировано
i_block	ULONG[15]	Указатели на блоки, в которых записаны данные файла
i_version	ULONG	Версия файла (для использования NFS)
i_file_acl	ULONG	ACL (Access Control List, список прав доступа) файла
i_dir_acl	ULONG	ACL каталога
i_faddr	ULONG	Адрес фрагмента
i_frag	UCHAR	Номер фрагмента
i_fsize	UCHAR	Размер фрагмента
i_pad1	USHORT	Заполнение
i_reserved2	ULONG[2]	Зарезервировано

Таблица П1.4. *Тип и права доступа к файлу*

Название поля	Тип	Описание поля
S_IFMT	**F000**	**Маска для типа файла**
S_IFSOCK	A000	Доменное гнездо (socket)
S_IFLNK	C000	Символическая ссылка

Таблица П1.4 *(окончание)*

Название поля	Тип	Описание поля
S_IFREG	8000	Обычный файл (regular)
S_IFBLK	6000	Блок-ориентированное устройство
S_IFDIR	4000	Каталог
S_IFCHR	2000	Байт-ориентированное устройство
S_IFIFO	1000	Именованный канал (fifo)
S_ISUID	0800	SUID — бит смены владельца
S_ISGID	0400	SGID — бит смены группы
S_ISVTX	0200	Бит сохранения задачи (sticky bit, "липкий" бит)
S_IRWXU	**01C0**	**Маска прав владельца файла**
S_IRUSR	0100	Право на чтение
S_IWUSR	0080	Право на запись
S_IXUSR	0040	Право на выполнение
S_IRWXG	**0038**	**Маска прав группы**
S_IRGRP	0020	Право на чтение
S_IWGRP	0010	Право на запись
S_IXGRP	0008	Право на выполнение
S_IRWXO	**0007**	**Маска прав остальных пользователей**
S_IROTH	0004	Право на чтение
S_IWOTH	0002	Право на запись
S_IXOTH	0001	Право на выполнение

Таблица П1.5. *Специальные индексные дескрипторы*

Название	Значение	Описание
EXT2_BAD_INO	1	Индексный дескриптор, в котором перечислены адреса дефектных блоков на диске
EXT2_ROOT_INO	2	Индексный дескриптор корневого каталога файловой системы
EXT2_ACL_IDX_INO	3	Индексный дескриптор ACL
EXT2_ACL_DATA_INO	4	Индексный дескриптор ACL

Таблица П1.5 *(окончание)*

Название	Значение	Описание
EXT2_BOOT_LOADER_INO	5	Индексный дескриптор загрузчика
EXT2_UNDEL_DIR_INO	6	Неудаляемый индексный дескриптор каталога
EXT2_FIRST_INO	11	Первый свободный индексный дескриптор

Таблица П1.6. *Структура записи в файле каталога*

Название поля	Тип	Описание поля
Inode	ULONG	Номер индексного дескриптора файла
rec_len	USHORT	Длина записи
name_len	USHORT	Длина имени файла
Name	CHAR[0]	Имя файла

Приложение 2

HOWTO

- The Linux 3Dfx HOWTO — описывает установку графического акселератора 3Dfx.
- 4mb Laptop HOWTO — установка Linux на слабые (RAM 4 Мбайт, жесткий диск менее 200 Мбайт) ноутбуки.
- Linux Access HOWTO — как адаптировать Linux для доступа тем, кто его не использует.
- Установка Linux на Acer LapTop HOWTO — описывает установку Linux на ноутбуки Acer.
- Advanced Bash-Scripting HOWTO — руководство по использованию языка сценариев командной оболочки Bash.
- Linux 2.4 Advanced Routing HOWTO — описание процесса маршрутизации для ядра Linux версии 2.4.
- Linux AI & Alife HOWTO — информация об искусственном интеллекте, программах для Linux по данному вопросу, ссылки.
- Alpha-HOWTO, Brief Introduction to Alpha Systems and Processors — краткий обзор по существующим процессорам Alpha, чипсетам и системам.
- Antares-RAID-sparcLinux-HOWTO — описывает, как устанавливать, конфигурировать и сопровождать аппаратный RAID, построенный на 5070 SBUS RAID-контроллере фирмы Antares Microsystems.
- Apache-Overview-HOWTO — обзор Web-сервера Apache.
- Linux Assembly HOWTO — программирование на языке ассемблера для Linux под процессоры i386.
- Linux Astronomy HOWTO — документ рассказывает об использовании Linux-решений в астрономии.
- Linux AX25-HOWTO, Amateur Radio — как установить и сконфигурировать поддержку для пакетного радиопротокола AX25.

- Bandwidth Limiting HOWTO — описывает, как установить Linux-сервер для ограничения входящего трафика, а также более эффективного использования интернет-соединения.
- BASH Programming — Introduction HOWTO — введение в программирование скриптов командного интерпретатора.
- BASH Prompt HOWTO — создание и управление терминалом из командной строки. Стандартные escape-последовательности и т. п.
- Linux Belarusian HOWTO — краткое руководство по установке белорусского языка в Linux-консоли, X Window System, Web-браузерах, текстовых редакторах и т. п.
- The Belgian HOWTO — Linux-ресурсы для бельгийцев.
- Linux Benchmarking HOWTO — обсуждение различных способов определения производительности операционной системы.
- Beowulf HOWTO — введение в архитектуру Beowulf Supercomputer и представление информации о параллельном программировании. Включает ссылки на другие ресурсы.
- Boot + Root + Raid + LILO: Software Raid mini-HOWTO — руководство по установке RAID-массива с использованием raidtools.
- The Linux Bootdisk HOWTO — как строить собственные загрузочные и корневые дискеты для загрузки Linux.
- The Linux BootPrompt HOWTO — использование аргументов командной строки для передачи параметров ядру Linux при загрузке.
- Linux BRIDGE-STP-HOWTO — описывает установку сетевого моста.
- C++ Programming HOWTO — обсуждаются методы программирования на C++.
- C-C++ Beautifier HOWTO — рекомендации по форматированию программ на C/C++ для лучшей читабельности.
- C editing with VIM HOWTO — введение по использованию редактора vi/VIM для программирования.
- Cable Modem Providers HOWTO — базовые вопросы по подключению к Linux кабельного модема.
- CDServer-HOWTO — описывает организацию CD-сервера на Linux.
- CD-Writing HOWTO — запись CD-ROM под Linux.
- The Linux CD-ROM HOWTO — установка, конфигурация и использование приводов CD-ROM под Linux.
- Chinese HOWTO — установка китайского языка.

- Chroot-BIND HOWTO — описывает установку программы BIND, запускаемой в chroot для усиления безопасности системы.
- Linux Cluster HOWTO — организация высокопроизводительных кластерных систем на Linux.
- Linux Commercial HOWTO — список коммерческого программного обеспечения для Linux.
- Compaq-Remote-Insight-Board-HOWTO — инсталляция Linux на сервере Compaq ProLiant без физического доступа к системе.
- Conexant/Rockwell modem HOWTO — использование программных модемов (soft-модемов) на чипсетах Conexant и Rockwell под Linux.
- Configuration HOWTO — качественная и быстрая настройка операционной системы Linux box. Конфигурация большинства распространенных программ и сервисов.
- Linux Consultants HOWTO — список программ, предоставляющих коммерческую поддержку Linux. Заменило Linux Consultants Guide.
- CPU Design HOWTO — рассказывает о разработке и производстве процессоров.
- CVS-RCS HOWTO — практическое руководство по быстрой установке CVS/RCS — системы контроля версий исходного кода программ.
- Cyrus IMAP HOWTO — руководство по установке, конфигурации и запуску Cyrus Imap и Cyrus Sasl.
- The Linux Danish/International HOWTO — настройка локали для датчан.
- DB2 for Linux HOWTO — инструкции по установке DB2 Universal Database for Linux для следующих дистрибутивов Caldera OpenLinux 2.4, Debian, Red Hat Linux 6.2, SuSE Linux 6.2 и 6.3, TurboLinux 6.0.
- Diald HOWTO — несколько типичных сценариев для легкого использования Diald. Заменило Diald mini-HOWTO.
- Diskless Nodes HOWTO for Linux — как установить бездисковую станцию с операционной системой Linux.
- Diskless-root-NFS-HOWTO — настройка сервера и конфигурация клиентов для бездисковой загрузки по сети.
- English-language GNU/Linux Distributions on CD-ROM — список англоязычных дистрибутивов Linux.
- DNS HOWTO — настройка DNS.
- From DOS/Windows to Linux HOWTO — для мигрирующих с DOS и Windows на операционную систему Linux.
- The dosemu HOWTO — настройка DOSEMU — программы эмуляции DOS.

- DSL HOWTO for Linux — настройка DSL-соединений.
- DVD Playing HOWTO — воспроизведение дисков DVD в операционной системе Linux.
- Linux Ecology HOWTO — обсуждение, как компьютеры с операционной системой Linux влияют на экологию.
- Emacs Beginner's HOWTO — введение в редактор Emacs.
- The Linux Emacspeak HOWTO — использование синтезатора речи.
- Enterprise Java for Linux HOWTO — установка Enterprise Java environment на Linux, включая JDK, Web-сервер, поддержку Java servlets, доступ к базам данных через JDBC и поддержку Enterprise Java Beans (EJBs).
- Linux-Esperanto-HOWTO — Linux и эсперанто.
- Linux Ethernet HOWTO — настройка сетевых карт Ethernet, решение проблем, конфигурация.
- Event HOWTO — создание презентаций.
- Filesystems HOWTO — о файловых системах и доступе к файловым системам.
- Finnish HOWTO — использование финского языка в Linux.
- Firewall and Proxy Server HOWTO — установка и настройка соответствующих сервисов.
- Font HOWTO — вопросы об использовании шрифтов в Linux.
- Framebuffer HOWTO — как использовать framebuffer в Linux на различных аппаратных платформах.
- Francophones-HOWTO — франкоговорящие и Linux.
- From Power Up To Bash Prompt — что происходит с Linux с момента включения питания до появления приглашения командной оболочки.
- Ftape HOWTO — использование ленточных накопителей.
- The Linux GCC HOWTO — установка C-компилятора и библиотек GNU под Linux, компиляция, линковка, запуск и отладка приложений.
- German HOWTO — немецкий язык и Linux.
- Glibc 2 HOWTO — инсталляция и использование библиотеки GNU C v2 (libc6) на Linux.
- Linux Hardware Compatibility HOWTO — список аппаратуры, поддерживаемой Linux, и помощь в отыскании необходимых драйверов.
- The Hebrew HOWTO — как сконфигурировать Linux для использования символов иврита в X Windows.
- Hellenic HOWTO — греческий и Linux.

- HOWTO HOWTO — список инструментов, процедур и подсказок для написания HOWTO.
- The Linux HOWTO Index — список существующих HOWTO.
- HP HOWTO — использование продукции Hewlett-Packard (HP) с Linux.
- i810 with XFree86 4.*x* HOWTO — описывает, как настроить и запустить XFree86 4.*x* на чипсете i810, используя специальные свойства ядра версии 2.4.0.
- Installing LinuxPPC-2000 on the IBM RS/6000 43P model 7248 HOWTO — установка LinuxPPC-2000 на IBM RS/6000 43P model 7248 series.
- Linux Information Sheet — содержит основную информацию об операционной системе Linux.
- Linux Infrared HOWTO — использование инфракрасного порта в Linux.
- Ingress II HOWTO — установка Ingress II Relational Database Management System на Linux.
- The Linux Installation HOWTO — описывает установку Linux.
- The Linux Intranet Server HOWTO — настройка Intranet с использованием Linux как сервера доступа к UNIX, Netware, NT и Windows.
- Linux IP Masquerade HOWTO — как разрешить Linux IP Masquerade на Linux-системе.
- Linux IPCHAINS HOWTO — установка и конфигурирование брандмауэра (firewall).
- Linux IPX HOWTO — установка и конфигурирование поддержки в ядре Linux протокола IPX.
- ISP-Hookup HOWTO — как, используя Linux, подключиться к Internet Service Provider.
- "Pocket" ISP based on Red Hat Linux — как установить на одной машине на базе дистрибутива Red Hat Linux сервер входящих модемных соединений, виртуальный Web-хостинг, почтовый и FTP-серверы.
- Linux Italian HOWTO — итальянский и Linux.
- Java CGI HOWTO — использование языка программирования Java для написания CGI-программ.
- Java Decompiler HOWTO — декомпиляция Java-программ.
- Linux on the Sun JavaStation NC HOWTO — описывает установку Linux OS на Sun JavaStation NC.
- Jaz-drive HOWTO — конфигурация и использование привода Iomega Jaz под Linux.

- The Linux Kernel HOWTO — детальное описание конфигурации ядра операционной системы, компиляции, обновления.
- The Linux keyboard and console HOWTO — содержит информацию о клавиатуре и консоли Linux и об использовании не-ASCII-символов.
- RedHat Linux KickStart HOWTO — описывает использование утилиты Red Hat Linux KickStart для быстрой инсталляции большого числа идентичных Linux-систем.
- Kiosk HOWTO — организация Web-киоска с использованием Linux, X11R6, FVWM2, Netscape Navigator 4.*x*.
- Kodak Digital Camera HOWTO — совместная работа цифрового аппарата Kodak и Linux.
- Linux Laptop HOWTO — описывает установку Linux на ноутбуки (конфигурация, драйверы и т. п.).
- Large Disk HOWTO — все о проблеме 1024 цилиндра и о геометрии жесткого диска.
- LDAP Linux HOWTO — информация об установке, конфигурировании, запуске и сопровождении службы LDAP (Lightweight Directory Access Protocol) на операционной системе Linux.
- LDAP Implementation HOWTO — технические аспекты сохранения приложений на сервере LDAP. В основном — о конфигурации различных приложений.
- Linux Documentation Project Reviewer HOWTO — обзор проекта документации Linux (LDP).
- Linux Crash Rescue HOWTO — обсуждаются методы восстановления Linux-системы в случае ее отказа.
- Linmodem-Mini-HOWTO — описывает Lin-модем (поддержку под Linux программного модема, т. н. soft-модема).
- Linux 2.4.x Initialization for IA-32 HOWTO — описывает инициализацию ядра Linux 2.4 на архитектуре IA-32 процессоров.
- Linux From Scratch HOWTO — описывает процесс создания собственной Linux-системы из уже установленного дистрибутива Linux. Заменил Linux From Scratch guide.
- Linux + Windows HOWTO — совместная установка и работа систем Linux и Windows на одном компьютере.
- LinuxDoc + Emacs + Ispell HOWTO — для писателей и переводчиков Linux HOWTO или других документов из Linux Documentation Project. Подсказки по использованию Emacs и Ispell.

- Loopback Encrypted Filesystem HOWTO — рассказывает об установке и использовании файловой системы, монтируемой пользователем, которая на лету шифрует и дешифрует хранящиеся на ней данные.
- Logical Volume Manager HOWTO — описание Linux LVM.
- The Linux Electronic Mail Administrator HOWTO — описывает установку и обслуживание электронной почты (e-mail) под Linux. Предназначен для администраторов Linux-систем.
- The Linux Mail User HOWTO — введение в мир электронной почты (e-mail) под Linux. Ориентировано на пользователей почты, подключенных к Интернету через интернет-провайдеров (ISP). Рассказывает о типичных конфигурациях Linux-систем для домашнего использования и использования в малом бизнесе.
- Majordomo and MajorCool HOWTO — описывает инсталляцию и конфигурацию Majordomo Mailing List Software (система тематической рассылки электронной почты) и MajorCool (утилита для администрирования списков рассылки Majordomo через CGI-скрипт).
- Masquerading Made Simple HOWTO — описывает, как разрешить использовать свойства ядра Linux — IP Masquerade. Используется совместно с IP-Masquerade-HOWTO.
- Linux Medicine-HOWTO — список программного обеспечения для использования Linux в медицинских целях.
- Linux Meta-FAQ — список источников информации о Linux.
- The MGR Window System HOWTO — информация об инсталляции, конфигурировании и запуске MGR Window System.
- Alpha Miniloader HOWTO — описывает Alpha Linux Miniloader (также известный как MILO) — программу, предназначенную для систем на базе процессора Alpha, используемую для инициализации компьютера и загрузки Linux.
- Encrypted Tunnels using SSH and MindTerm HOWTO — использование SSH и Java-программы MindTerm для быстрого создания зашифрованного VPN-like туннеля через небезопасные сетевые соединения.
- Linux/MIPS HOWTO — описывает MIPS-ориентированную версию Linux, общие проблемы и их решения.
- Modem HOWTO — помогает в выборе, соединении, настройке и решении проблем с модемами для персональных компьютеров.
- Modem-Dialup-NT HOW-TO — помогает в установке модема для дозвона к удаленному серверу, подобному Windows NT RAS или Linux RAS (Remote Access Server, сервер удаленного доступа).

- Linux Loadable Kernel Module HOWTO — рассказывает о том, что такое загружаемые модули ядра (LKM), как их использовать и создавать.
- The Linux MP3 HOWTO — описывает аппаратуру, программное обеспечение и процедуры, необходимые для создания и воспроизведения mp3-файлов под Linux.
- MP3 Player Box HOWTO — описывает, как создать, сконфигурировать, установить и использовать MP3-плеер на базе операционной системы Linux. Содержит список необходимого аппаратного обеспечения и ответы на наиболее часто задаваемые вопросы.
- Multi Disk System Tuning — как лучше всего использовать диски и разделы для операционной системы Linux.
- Multicast over TCP/IP HOWTO — описывается пересылка одновременно нескольких потоков с использованием TCP/IP-сетей.
- Managing Multiple Operating Systems HOWTO — описывается, как использовать сменные диски для установки и управления альтернативными операционными системами, если существует постоянный жесткий диск и первичная операционная система.
- Mutt-i, GnuPG and PGP HOWTO — кратко рассказывается, как конфигурировать Mutt-i, PGP и GnuPG.
- Linux Netstation HOWTO — описывает, как настроить IBM Netstation в локальной сети, используя систему с операционной системой Linux в качестве сервера.
- Linux NCD mini-HOWTO — описывает, как настроить NCD ThinSTAR в локальной сети, используя систему с операционной системой Linux в качестве сервера.
- Linux Networking HOWTO — информация о сети для Linux. Ранее называлось Net 3/4 и Net-3 HOWTO.
- Linux NETMEETING HOWTO — описывает, как взаимодействует Microsoft NetMeeting с Linux-системой.
- Network boot and exotic root HOWTO — описывает, как, используя IP-протокол, быстро установить Linux-сервер для предоставления бездисковым Linux-клиентам всего, что им необходимо для старта и работы.
- The Linux Networking Overview HOWTO — обзор сетевых возможностей операционной системы Linux. Предоставляет ссылки на источники информации для более подробного ознакомления с деталями.
- NFS HOWTO — как установить NFS-клиенты и серверы.
- The Linux NIS(YP)/NYS/NIS+ HOWTO — как сконфигурировать Linux в качестве клиента NIS(YP) или NIS+ и как установить NIS-сервер.

- NetWare Loadable Module Programming HOWTO — как разрабатывать NetWare Loadable Modules под Linux, используя GNU CC и nlmconv из GNU binutils.
- Online Troubleshooting Resources HOWTO — предоставляет Linux-пользователям список ресурсов Интернета, позволяющих помочь в решении проблем, связанных с Linux.
- Linux Optical Disk HOWTO — описывает установку и конфигурацию оптических дисков в Linux.
- Oracle 7 Database HOWTO — руководство по установке и конфигурированию Oracle 7 Database Server на операционной системе Linux.
- Oracle 8 for Linux Installation HOWTO — руководство по установке и конфигурированию Oracle 8i Enterprise Edition на операционной системе Linux.
- Palm OS Desktop HOWTO — как использовать устройство с операционной системой Palm OS совместно с операционной системой Linux.
- Linux Parallel Processing HOWTO — описываются базовые функции параллельного вычисления, доступные для Linux-пользователей: SMP Linux-системы, кластеры Linux-систем, параллельное исполнение с использованием мультимедийных инструкций процессора MMX, присоединенных (параллельных) процессоров.
- Linux PCI-HOWTO — информация о работе Linux с PCI-устройствами.
- Linux PCMCIA HOWTO — как установить и использовать PCMCIA Card для Linux.
- phhttpd-HOWTO — HTTP-акселератор phhttpd. Позволяет ускорить обработку статических HTTP-запросов.
- PHP HOWTO — как разрабатывать PHP-программы и мигрировать с GUI-приложений Windows 95 на связку PHP + HTML + DHTML + XML + + Java Applets + JavaScript. Применимо ко всем операционным системам, где используется PHP.
- PLIP Install HOWTO — как установить Gnu XXPIPE Linux-дистрибутив на компьютер без сетевой Ethernet-карты, без привода CD-ROM, но имея дисковод и удаленный NFS-сервер, подключенный параллельным кабелем через null-modem.
- The Linux Plug and Play HOWTO — как операционная система Linux поддерживает Plug and Play.
- Polish HOWTO — польский язык и Linux.
- Portuguese HOWTO — португальский язык и Linux.

- Database-SQL-RDBMS HOWTO for Linux (PostgreSQL Object Relational Database System) — практическое руководство по очень быстрой инсталляции SQL Database и соответствующих утилит на UNIX-систему.
- Linux PPP HOWTO — как подключить Linux-систему к PPP-серверу, как, используя PPP, соединить две локальные сети вместе, методы настройки системы на базе операционной системы Linux в качестве PPP-сервера.
- The Linux Printing HOWTO — подборка информации о том, как создавать, просматривать, печатать или отправлять факсом документы под Linux.
- The Linux Printing Usage HOWTO — как использовать print system под Linux.
- Process Monitor HOWTO for Linux — как отслеживать Linux/UNIX-процессы и в случае зависания перестартовать их без ручного вмешательства (автоматически).
- Program Library HOWTO — руководство для программистов, описывает как создавать и использовать программные библиотеки (ключая статические, разделяемые и динамически загружаемые) в операционной системе Linux.
- Linux and Psion HOWTO — как использовать PDA Psion (КПК, карманный персональный компьютер) совместно с операционной системой Linux.
- Qmail VMailMgr and Courier-Imap HOWTO — построение почтового сервера, который может поддерживать виртуальные домены и предоставлять SMTP-, POP3- и IMAP-сервисы, используя альтернативное (sendmail) программное обеспечение.
- Linux Quake HOWTO — как установить, запустить Quake, QuakeWorld и Quake II на системе Intel Linux.
- The Linux Reading List HOWTO — список книг по операционной системе UNIX (Linux).
- Burning a RedHat CD HOWTO — как создать свой компакт-диск Red Hat Linux.
- Remote Serial Console HOWTO — как установить аппаратное обеспечение для использования последовательной консоли.
- Root RAID HOWTO cookbook — описывает использование только старых версий raidtools (0.50 и младше).
- RPM HOWTO — RPM at Idle — использование RPM Package Manager.
- RPM-for-UNIX HOW-TO — использование Redhat RPM-программы на различных UNIX-системах.

- Sat (Satellite Technology) HOWTO — настройка и установка спутниковых соединений в Linux.
- The Linux SCSI subsystem in 2.4 HOWTO — описывает подсистему SCSI.
- The Linux SCSI programming HOWTO — программирование под Linux SCSI-интерфейса.
- Secure Programming for Linux and UNIX HOWTO — описание набора правил по написанию безопасных программ для операционных систем Linux и UNIX.
- Linux Security HOWTO — базовый обзор методов обеспечения безопасности операционной системы Linux.
- Serbian HOWTO — сербский язык и Linux.
- Serial HOWTO — описывает работу с последовательным портом, в том числе мультипортовые карты.
- Serial Laplink HOWTO — описывает, как установить laplink-соединение между двумя компьютерами.
- Serial Programming HOWTO — как программировать соединения с различными устройствами, используя последовательный порт.
- Linux Shadow Password HOWTO — как установить и сконфигурировать Linux password Shadow.
- Slovenian HOWTO — словенский язык и Linux.
- SMB HOWTO — как с помощью пакета Samba использовать с Linux протокол Server Message Block, SMB (иногда называемый Session Message Block, NetBIOS или LanManager).
- Linux SMP HOWTO — конфигурация SMP под Linux.
- Building and Installing Software Packages for Linux — руководство по компиляции и установке программного обеспечения UNIX под Linux.
- Free Software Project Management HOWTO — руководство по нетехническим аспектам свободного программного обеспечения.
- Software-RAID HOWTO — как использовать программный RAID под Linux.
- Software Release Practice HOWTO — описывает Linux open-source-проекты.
- The Linux Sound HOWTO — описывает поддержку звука под Linux (аппаратура, конфигурация и т. п.).
- The Linux Sound Playing HOWTO — список программ для Linux, которые воспроизводят музыкальные файлы различных форматов.
- Spanish Linux HOWTO — испанский язык и Linux.

- SPARC-HOWTO — описывает установку Linux на рабочих станциях SPARC.
- Speech Recognition HOWTO — автоматическое распознавание речи (Automatic Speech Recognition, ASR) в Linux. Описываются несколько пакетов, доступных для пользователей и разработчиков.
- SRM Firmware HOWTO — описывает, как загрузить Linux/Alpha, используя SRM-консоль.
- Building a Secure RedHat Apache Server HOWTO — описывается, как PKI и SSL работают вместе.
- Sybase Adaptive Server Anywhere for Linux HOWTO — руководство по инсталляции и базовому администрированию SQL Anywhere Studio для Linux.
- Tango 2000 HOWTO — описывает инсталляцию и конфигурирование Pervasive Software's Tango Application Server на Sun Solaris и Linux.
- The Linux Tcl and Tk HOWTO — использование языка Tcl под Linux.
- The teTeX HOWTO: The Linux-teTeX Local Guide — инсталляция и использование teTeX, TeX и LaTeX.
- Text-Terminal HOWTO — что такое текстовые терминалы, как они работают, как их инсталлировать и конфигурировать.
- The Linux Thai HOWTO — тайский язык и Linux.
- Thin Client: New User Guide — как превратить старые компьютеры в быстрые терминалы.
- The Linux Tips HOWTO — как настроить Linux для лучшей производительности.
- Turkish HOWTO — турецкий язык и Linux.
- UMSDOS HOWTO — как использовать Umsdos в различных конфигурациях.
- The Unicode HOWTO — как адаптировать Linux-систему для использования UTF-8.
- The UNIX and Internet Fundamentals HOWTO — описывает работу PC-совместимых компьютеров, UNIX-подобных операционных систем и Интернета нетехническим языком.
- The UNIX Hardware Buyer HOWTO — информация о том, как покупать и конфигурировать аппаратное обеспечение для UNIX-подобных операционных систем.
- The UPS HOWTO — подключение источника бесперебойного питания к Linux.

- User Authentication HOWTO — описывается, как хранится информация о пользователях и группах, как происходит аутентификация пользователя в Linux-системе (PAM) и как обеспечивается безопасность в процессе аутентификации пользователя.
- Linux User Group HOWTO — все о группах пользователей в операционной системе Linux.
- The Linux UUCP HOWTO — описывает установку и сопровождение UUCP под Linux.
- Linux VAR HOWTO — список компаний, поддерживающих существующие продукты.
- VCR-HOWTO (Using your GNU/Linux computer as a VCR) — руководство по настройке компьютера в качестве цифрового видеомагнитофона с использованием драйвера video4linux и поддерживаемой TV-карты.
- Vim Color Editor HOWTO (Vim Improved w/syntax color highlighting) — руководство по очень быстрой инсталляции редактора Vim на системах Linux или UNIX.
- Virtual Services HOWTO — руководство по виртуализации сервиса.
- VMailMgr HOWTO — как установить VMailMgr с поддержкой POP3 виртуальных доменов в кооперации с Qmail.
- VME HOWTO — запуск VMEbus Pentium, основанных на VMEbus.
- From VMS to Linux HOWTO — использование VMS и Linux.
- VoIP Howto — передача голоса через Интернет (Voice Over IP).
- VPN HOWTO — как установить виртуальную частную сеть (Virtual Private Network) с помощью Linux.
- Linux VPN Masquerade HOWTO — как сконфигурировать Linux-брандмауэр (firewall) с использованием IP-маскарада.
- Wacom Tablet HOWTO — использование графических планшетов фирмы Wacom под Linux и xfree86.
- Windows LAN server HOW-TO — использование Linux как офисного сервера для клиентов с Microsoft Windows 9*x*.
- Winmodems-and-Linux HOWTO — как заставить функционировать Winмодем (программный модем для Windows) под Linux.
- Wireless HOWTO for Linux Systems — инсталляция и настройка беспроводных сетей под Linux.
- Linux WWW HOWTO — информация о настройке WWW-сервисов под Linux (клиентской и серверной частей).

- A mSQL and perl Web Server HOWTO — как построить SQL client/server-базу данных, используя WWW и HTML для пользовательского интерфейса.
- Linux XDMCP HOWTO — как установить XDMCP (X Display Manager Control Protocol).
- The Linux XFree86 HOWTO — как установить и сконфигурировать XFree86 версию X Window System (X11R6) для Linux-систем.
- Linux Touch Screen HOWTO — как установить и настроить сенсорный экран под XFree86.
- XFree86 Video Timings HOWTO — как настроить комбинацию видеокарта/монитор под XFree86.
- Using the Xinerama Extensions to MultiHead X — как сконфигурировать XFree86 с поддержкой нескольких мониторов и Xinerama-расширений.
- XML-RPC HOWTO — описывает, как использовать XML-RPC для поддержки клиентов и серверов с различными языками.
- X Window System Architecture Overview HOWTO — обзор архитектуры X Window System.
- The X Window User HOWTO — информация о конфигурировании X Window.

Приложение 3

Мини-HOWTO

- The 3 Button Serial Mouse mini-HOWTO — настройка трехкнопочной последовательной мыши для работы в Linux.
- 3D Graphics Modelling and Rendering mini-HOWTO — детальная инструкция по получению и установке программы графического рендеринга и трехмерного моделирования для дистрибутива Red Hat Linux.
- Linux ACP Modem (Mwave) mini-HOWTO — описывает, как скомпилировать, установить и использовать модем ACP (Mwave) для ноутбуков IBM Thinkpad 600E, 600, и 770x.
- ADSL HOWTO for Linux Systems — заменил DSL HOWTO for Linux.
- Linux ADSM mini-HOWTO — как установить и использовать клиент для коммерческого ADSM-резервного копирования для Linux/i386.
- Linux Advocacy mini-HOWTO — эффективное использование Linux.
- Alsa-sound-mini-HOWTO — описывает инсталляцию звуковых драйверов ALSA для Linux.
- Apache+DSO+mod_ssl+mod_perl+php+mod_auth_nds+mod_auth_mysql+mod_fastcgi mini-HOWTO — инсталляция Web-сервера Apache с поддержкой модулей mod_perl, mod_ssl и php.
- Linux Apache SSL PHP/FI frontpage mini-HOWTO — о построении Web-сервера, поддерживающего динамически изменяемое содержимое.
- Automount mini-HOWTO — описывает autofs automounter (конфигурирование).
- Linux Backspace/Delete mini-HOWTO — как заставить работать клавиши <Backspace> и <Delete> в консоли и в X.
- Backup-With-MSDOS mini-HOWTO — как использовать Linux-совместимый ленточный накопитель (стример), установленный на компьютере MS-DOS.
- Battery Powered Linux Mini-HOWTO — как настроить потребление питания в Linux.

- Installing Boca Card Mini-HOWTO — установка шестнадцатипортовой последовательной карты Boca в Linux.
- BogoMips mini-HOWTO — некоторая информация о BogoMips.
- Bridging mini-HOWTO — описывает, как установить Ethernet-мост.
- Linux Bridge+Firewall Mini-HOWTO — настройка Ethernet-моста и брандмауэра (firewall).
- Bridge + Firewall + DSL Mini-HOWTO — конфигурирование Linux-системы в качестве брандмауэра (firewall) и моста с DSL-сетевым соединением.
- The BTTV Mini-HOWTO — описывает аппаратуру, программное обеспечение и настройки, необходимые для использования TV-карт на базе чипсета bt8x8 под Linux.
- Bzip2 mini-HOWTO — как использовать программу сжатия bzip2.
- Call-back mini HOWTO — как настроить Call-back, используя Linux-систему и модем.
- The Linux Cipe+Masquerading mini-HOWTO — как установить VPN, используя Cipe на Linux masquerading firewall.
- The Clock Mini-HOWTO — как установить и сохранить время на компьютере.
- COFFEE-HOWTO — да, Linux может делать кофе, и это получается хорошо.
- Commercial Port Advocacy mini-HOWTO — обсуждение темы об использовании коммерческого программного обеспечения и портировании его под Linux.
- Compressed TCP/IP-Sessions using SSH-like tools — как сжимать потоки данных.
- DHCP mini-HOWTO — отвечает на основные вопросы, как установить на Linux DHCP-сервер или DHCP-клиент.
- Divert Sockets mini-HOWTO — описывает, как получить, скомпилировать и использовать FreeBSD divert sockets под Linux.
- DocBook Install mini-HOWTO — практическое руководство по быстрой установке DocBook и преобразованию SGML-файлов в HTML, PostScript и PDF на GNU/Linux-системе.
- Setting Up Your New Domain Mini-HOWTO — установка новых компьютеров в вашем домене.
- Linux DPT Hardware RAID mini-HOWTO — как установить аппаратное обеспечение RAID под Linux.

- Linux Ext2fs Undeletion mini-HOWTO — как восстановить удаленные файлы в Second Extended File System.
- Ext2fs Undeletion of Directory Structures — как восстановить удаленную структуру каталогов файл за файлом.
- Linux simple fax printer server mini-HOWTO (faxsrv-mini-HOWTO) — наиболее простой путь установить fax-сервер на вашей Linux-системе.
- FBB Packet-radio BBS mini-HOWTO — установка и использование программного обеспечения для пакетного радио.
- XFree86 Font Deuglification Mini HOWTO — как исправить плохую читаемость шрифтов X Window.
- Firewall Piercing mini-HOWTO — как использовать PPP поверх telnet для прохождения через firewall.
- FTP mini-HOWTO — как использовать FTP-клиенты и FTP-серверы.
- GIS-GRASS mini-HOWTO — как инсталлировать и конфигурировать Геоинформационную систему (Geographic Information System, GIS) The Geographic Resources Analysis Support System (GRASS).
- Handspring-Visor mini-HOWTO — использование Visor с Linux через USB-порт.
- Hard Disk Upgrade Mini HOWTO — как скопировать Linux-систему с одного жесткого диска на другой.
- Home Electrical Device Control mini HOWTO — использование Linux для управления домашними электрическими устройствами.
- Home-Network-mini-HOWTO, Red Hat Linux 6. X as an Internet Gateway for a Home Network — конфигурирование дистрибутива Red Hat Linux как маршрутизатора для небольшой домашней или офисной сети. В том числе: masquerading, DNS, DHCP и основы безопасности.
- Howtos-with-LinuxDoc mini-HOWTO — как писать HOWTO, используя простой LinuxDoc-шаблон.
- Linux Install From PPA-Zip drive mini-HOWTO — как установить Linux с ZIP-привода, подключенного к параллельному порту, используя дистрибутив Slackware Linux.
- Linux Installation Strategies — обсуждение стратегии инсталляции при двойном использовании Windows и Linux.
- How to setup international keyboard in X Windows — модификация xmodmap и kimap для поддержки национальных клавиатур.
- Linux I/O port programming mini-HOWTO — как программировать аппаратные порты ввода/вывода.

- IP Sub-Networking mini-HOWTO — рассказывает о том, что такое подсети класса A, B или C.
- Setting Up IP Aliasing On A Linux Machine Mini-HOWTO — как установить и запустить IP aliasing на Linux-системе.
- IPMasquerading+Napster mini-HOWTO — как в системах, использующих IPMasq'd, работать с Napster.
- Linux IRC mini-HOWTO — описывает программы IRC для Linux.
- ISP-Connectivity mini-HOWTO — как установить PPP, соединиться с провайдером, сконфигурировать почту и новости, получить доменное имя и т. п.
- The Linux kerneld mini-HOWTO — как можно использовать функцию kerneld в ядрах Linux.
- The LBX mini-HOWTO — LBX (Low Bandwidth X) расширение X server, использующее сжатие X-протокола.
- Leased line mini-HOWTO — использование модема и конфигурация pppd.
- Lego Mindstorm with Linux Mini-HOWTO — использование Lego Group's Mindstorm Robotics Invention System (RIS) и Linux.
- Lilo mini-HOWTO — конфигурация Lilo.
- The Linux "Linux-DOS-Win95-OS2" mini-HOWTO — как могут уживаться четыре операционные системы на одном жестком диске.
- Linuxdoc Reference — An Introduction to the linuxdoc DTD, руководство по SGML.
- LinuxGL (GLX) QuakeWorld Client compile mini-HOWTO — как скомпилировать и настроить клиента OpenGL/GLX Linux QuakeWorld.
- The Linux+FreeBSD mini-HOWTO — как использовать Linux и FreeBSD на одной системе.
- Linux-Modem-Sharing Modem sharing mini-HOWTO — как установить на Linux-системе разделяемый через TCP/IP модем.
- NT OS Loader + Linux mini-HOWTO — как использовать загрузчик Windows NT для старта Linux.
- Linux+Solaris mini-HOWTO — как использовать совместно Solaris и Linux на одном компьютере.
- Linux + Windows 95 mini-HOWTO — как установить Linux на компьютер, на котором уже установлена Windows 95.
- Linux+WindowsNT mini-HOWTO — как установить совместно Linux и Windows NT на один компьютер и как их загружать из меню LILO.
- The Loadlin+Win95/98/ME mini-HOWTO — описывает, как использовать программу Loadlin с Windows 95/98/ME для загрузки Linux.

- The Loopback Root Filesystem HOWTO — как использовать устройство Linux loopback для создания файловой системы Linux.
- The MacTerminal mini-HOWTO — как конфигурировать компьютер Mac для использования в качестве Linux-терминала.
- Linux Mail-Queue mini-HOWTO — изменения конфигурации sendmail так, чтобы программа отправляла локальную почту немедленно.
- Mail2News mini-HOWTO — как отправить список рассылки через сервер новостей.
- The Linux Man Page mini-HOWTO — как писать man-документацию.
- Linux Modules Installation mini-HOWTO — в данное время не поддерживается.
- Linux MP3 CD Burning mini-HOWTO — создание звуковых компакт-дисков из MP3-файлов.
- Connecting to MS SQL 6.x+ via Openlink/PHP/ODBC mini-HOWTO — как подключиться к серверу баз данных MS SQL 6.*x*+ через ODBC-функции PHP3 с использованием драйверов Openlink под Linux.
- Multiboot with GRUB Mini-HOWTO — как установить на одном компьютере Windows 98, Windows 2000, DOS и Linux, используя менеджер загрузки GRUB.
- Win95 + WinNT + Linux multiboot using LILO mini-HOWTO — как переключать загрузку между Windows 95, Windows NT и Linux.
- NCD X terminal mini HOWTO — как подключить X-терминал NCD к UNIX-хосту.
- Netrom-Node mini-Howto — как установить пакет ax25-utilities.
- HOWTO for inHouse IntraNet — описывает, как создать сеть дома.
- News Leafsite mini-HOWTO — использование программы Leafnode для создания сервера новостей.
- NFS-Root mini-HOWTO — как создать бездисковую рабочую станцию на базе Linux, которая монтирует корневую файловую систему через NFS.
- NFS-Root-Client Mini-HOWTO — настройка клиента NFS.
- Nvidia OpenGL Configuration mini-HOWTO — как установить драйверы OpenGL для видеокарт фирмы nVIDIA на Linux. Также описывается, как установить XFree86, библиотеку OpenGL (Mesa), инструментарий OpenGL (Glut) и т. п.
- Linux off-line mailing method (offline mailaddr with 1 account) — как получать электронную почту на один электронный адрес для нескольких пользователей без постоянного подключения к Интернету.

- MS Outlook to UNIX Mailbox Conversion mini HOWTO — как конвертировать почтовую базу Microsoft Outlook в стандартные UNIX-форматы.
- The Linux Alphanumeric Pager Gateway Mini-HOWTO — как установить систему e-mail to Pager.
- Linux Partition HOWTO — как спланировать и разбить дисковое пространство для Linux-системы.
- Partition Rescue mini HOWTO — как восстановить Linux-раздел, если он был удален.
- PATH HOWTO — описание системной переменной PATH.
- LINUX PLIP mini-HOWTO — как создать и использовать Parallel Line Interface Protocol.
- PortSlave How-To using the Linux Router — использование Portslave с Linux-маршрутизатором.
- GNU/Linux Pre-Installation Checklist — список того, что нужно сделать до инсталляции Linux.
- GNU/Linux Post-Installation Checklist — список того, что нужно сделать сразу после инсталляции Linux.
- Programming Languages mini-HOWTO — описание основных языков программирования для Linux, а также библиотек для создания графического пользовательского интерфейса (Graphical User Interface, GUI) под Linux.
- ProxyARP Subnetting HOWTO — использование протокола Proxy Address Resolution Protocol (ARP).
- Linux web browser station — описывает инсталляцию интернет-киоска, базирующегося на Linux, для предоставления публичного доступа к интернет-услугам.
- mini-HOWTO install qmail with MH — инсталляция qmail.
- Quota mini-HOWTO — как разрешить квотирование в Linux-системе (базовые команды).
- The RCS mini-HOWTO — инсталляция и использование The GNU Revision Control System (RCS) под Linux.
- Linux Remote-Boot mini-HOWTO: Configuring Remote-Boot Workstations with Linux, DOS, Windows 95/98 and Windows NT — конфигурация системы удаленной загрузки для персональных компьютеров с возможностью выбора типа загружаемой операционной системой при старте загрузки.
- Remote X Apps mini-HOWTO — как запустить удаленно приложения X Window.
- RPM+Slackware Mini-Howto — как установить и работать с RPM в дистрибутиве Slackware Linux.

- Saving Space mini-HOWTO — как сохранить место на жестком диске при инсталляции операционной системы Linux.
- Secure POP via SSH mini-HOWTO — как устанавливать безопасные POP-соединения используя SSH.
- Sendmail+UUCP HOWTO — как настроить одиночный компьютер без прямого доступа к Интернету для отправки и получения электронной почты.
- sendmail address rewriting mini-HOWTO — как сконфигурировать sendmail для домашних пользователей, имеющих доступ к Интернету через модемное соединение.
- SLIP/PPP Emulator mini-HOWTO — как Linux-системе получить соединение через SLIP/PPP-эмулятор, такой как SLiRP или TIA.
- Small Memory Mini-HOWTO — как запустить Linux на компьютере, имеющем малый объем оперативной памяти.
- Sound Blaster AWE 32/64 HOWTO — установка и конфигурация Sound Blaster 32 (SB AWE 32, SB AWE 64) в операционной системе Linux.
- Linux Swap Space Mini-HOWTO — как разделить swap-раздел Linux с Windows.
- Sybase-PHP-Apache mini-HOWTO — как установить и запустить в Linux Apache Web-сервер, используя PHP для доступа к базе данных Sybase-ASE.
- Using Term to Pierce an Internet Firewall — использование term.
- TkRat mini-HOWTO — использование Linux-системы для приема и отправления электронной почты.
- Token-Ring mini-HOWTO — установка и настройка оборудования Token-Ring под Linux.
- Transparent Proxy with Squid mini-HOWTO — как установить и настроить proxy-сервер transparent caching HTTP, используя операционную систему Linux и squid.
- TrueType Fonts in Debian mini-HOWTO — как сконфигурировать дистрибутив Debian Linux под использование TrueType-шрифтов для отображения информации и печати.
- TrueType Fonts with XFree86 4.0.*x* mini-HOWTO — описывает, как использовать TrueType-шрифты с XFree86 4.0.*x*.
- The Linux Ultra-DMA Mini-Howto — как использовать жесткие диски и интерфейсы Ultra-DMA (Ultra-ATA, Ultra33 и Ultra66) с операционной системой Linux.
- The Staying Updated mini-HOWTO — как обновлять операционную систему.

- Linux on Sony VAIO mini-HOWTO — установка операционной системы Linux на компьютеры Sony VAIO.
- Virtual Web mini-HOWTO — как создать виртуальный Web-сайт в операционной системе Linux.
- Visible bell mini-Howto — как запретить терминалу воспроизводить звуковые сигналы, соответствующие коду 7.
- The VPN HOWTO — как установить виртуальную частную сеть (Virtual Private Network).
- Wacom Graphire USB Mini-HOWTO — установка графического USB-планшета фирмы Wacom в Linux.
- X11-big-cursor mini-HOWTO — как сделать большой курсор мыши в системе X Window.
- XDM and X Terminal mini-HOWTO — использование XDM для управления X-терминалами.
- Second Mouse in X mini-HOWTO — настройка второй мыши в X Window.
- Linux XFree-to-Xinside mini-HOWTO — как конвертировать настройки монитора из XFree86 в XInside/XiGraphics.
- How to change the title of an xterm — как динамически изменять заголовок окна и подпись значка программы xterm.
- Zip Drive Mini-HOWTO — как установить и использовать привод Iomega ZIP с Linux.
- Installing Linux on ZIP disk using ppa ZIP Drive Mini-Howto — использование ZIP-привода с подключением к параллельному порту.

Приложение 4

Дерево параметров настройки ядра

В этом приложении приведено дерево настроек ядра Linux с настройками, используемыми по умолчанию в дистрибутиве Red Hat Linux 7.2. Используемые соглашения:

- `[*]` — вкомпилировано в ядро;
- `[ ]` — не компилируется;
- `<M>` — вынесено в модуль;
- `< >` — не компилировано, может быть вынесено в модуль;
- → — вложенное меню.

```
Code maturity level options  →
     [*] Prompt for development and/or incomplete code/drivers
Loadable module support  →
     [*] Enable loadable module support
     [*]   Set version information on all module symbols
     [*]   Kernel module loader
Processor type and features  →
     (Pentium-Pro/Celeron/Pentium-II) Processor family
     [ ] PGE extensions (not for Cyrix)
     <M> Toshiba Laptop support
     <M> /dev/cpu/microcode — Intel IA32 CPU microcode support
     <M> /dev/cpu/*/msr — Model-specific register support
     <M> /dev/cpu/*/cpuid — CPU information support
     [ ] E820 proc support
     (4GB) High Memory Support
     (3GB) Maximum Virtual Memory
     [ ] Math emulation
     [*] MTRR (Memory Type Range Register) support
```

```
     [ ] Symmetric multi-processing support
     [ ] APIC support on uniprocessors
     [*] Memory eXpansion Technology (MXT) Support
General setup  →
     [*] Networking support
     [*] PCI support
     (Any)   PCI access mode
     [*] PCI device name database
     [*] EISA support
     [ ] MCA support
     [*] Support for hot-pluggable devices
     PCMCIA/CardBus support  →
          <M> PCMCIA/CardBus support
          [*]   CardBus support
          [*]   i82365 compatible bridge support
          [*]   Databook TCIC host bridge support
     Hotplug PCI Support  →
          [*] Support for Hotplug PCI (EXPERIMENTAL)
          <M>   Compaq Hotplug PCI driver
     [*] System V IPC
     [*] BSD Process Accounting
     [*] Sysctl support
     (ELF) Kernel core (/proc/kcore) format
     <M> Kernel support for a.out binaries
     <*> Kernel support for ELF binaries
     <M> Kernel support for MISC binaries
     [*] Power Management support
     [ ]   ACPI support
     <*>   Advanced Power Management BIOS support
     [ ]     Ignore USER SUSPEND
     [ ]     Enable PM at boot time
     [ ]     Make CPU Idle calls when idle
     [ ]     Enable console blanking using APM
     [*]     RTC stores time in GMT
     [ ]     Allow interrupts during APM BIOS calls
     [ ]     Use real mode APM BIOS call to power off
Binary emulation of other systems  →
     <M> Support for binary emulation of other systems
     <M>  SVR3/SVR4 (and derivates) binary emulation support
```

```
--- You have to select at least one of the following emulations:
<M>  UnixWare 7.x binary emulation support
< >  Solaris 2.x binary emulation support
<M>  iBCS2/iABI binary emulation support
[ ]    Include ISC specifics
[ ]    Include Xenix specifics
<M>  SCO Unix (OpenServer) binary emulation support
< >  Wyse V/386 binary emulation support
--- Support for foreign binary formats
<M>  Kernel support for COFF binaries
<M>  Kernel support for x.out binaries
[ ]    Include Xenix 286 segmented binary specifics
--- Linux-ABI debugging settings
[ ]  Enable verbose errors
```

Memory Technology Devices (MTD) →

```
< > Memory Technology Device (MTD) support
```

Parallel port support →

```
<M> Parallel port support
<M>   PC-style hardware
<M>     Multi-IO cards (parallel and serial)
[ ]     Use FIFO/DMA if available (EXPERIMENTAL)
[ ]     SuperIO chipset support (EXPERIMENTAL)
<M>     Support for PCMCIA management for PC-style ports
[ ]   Support foreign hardware
[*]   IEEE 1284 transfer modes
```

Plug and Play configuration →

```
<*> Plug and Play support
<*>   ISA Plug and Play support
[ ]   PNPBIOS support (EXPERIMENTAL)
```

Block devices →

```
<*> Normal PC floppy disk support
<M> XT hard disk support
<M> Parallel port IDE device support
--- Parallel IDE high-level drivers
<M>   Parallel port IDE disks
<M>   Parallel port ATAPI CD-ROMs
<M>   Parallel port ATAPI disks
<M>   Parallel port ATAPI tapes
<M>   Parallel port generic ATAPI devices
```

```
--- Parallel IDE protocol modules
<M>     ATEN EH-100 protocol
<M>     MicroSolutions backpack (Series 5) protocol
<M>     MicroSolutions backpack (Series 6) protocol
<M>     DataStor Commuter protocol
<M>     DataStor EP-2000 protocol
<M>     FIT TD-2000 protocol
<M>     FIT TD-3000 protocol
<M>     Shuttle EPAT/EPEZ protocol
<M>     Shuttle EPIA protocol
<M>     Freecom IQ ASIC-2 protocol
<M>     FreeCom power protocol
<M>     KingByte KBIC-951A/971A protocols
<M>     KT PHd protocol
<M>     OnSpec 90c20 protocol
<M>     OnSpec 90c26 protocol
<M> Compaq SMART2 support
<M> Compaq Smart Array 5xxx support
<M> Mylex DAC960/DAC1100 PCI RAID Controller support
<M> Loopback device support
<M> Network block device support
<*> RAM disk support
(4096)   Default RAM disk size
[*]   Initial RAM disk (initrd) support
```

Multi-device support (RAID and LVM) →

```
[*] Multiple devices driver support (RAID and LVM)
<*>  RAID support
<M>   Linear (append) mode
<M>   RAID-0 (striping) mode
<M>   RAID-1 (mirroring) mode
<M>   RAID-4/RAID-5 mode
<M>   Multipath I/O support
< >  Logical volume manager (LVM) support
```

Networking options →

```
<*> Packet socket
[*]   Packet socket: mmapped IO
[*] Kernel/User netlink socket
[*]   Routing messages
```

```
<*>   Netlink device emulation
[*] Network packet filtering (replaces ipchains)
[ ]   Network packet filtering debugging
[*] Socket Filtering
<*> Unix domain sockets
[*] TCP/IP networking
<M>   Threaded linUX application protocol accelerator layer (TUX)
[*]     External CGI module
[ ]  extended TUX logging format
[ ]  debug TUX
[*]   IP: multicasting
[*]   IP: advanced router
[*]     IP: policy routing
[*]       IP: use netfilter MARK value as routing key
[*]       IP: fast network address translation
[*]     IP: equal cost multipath
[*]     IP: use TOS value as routing key
[*]     IP: verbose route monitoring
[*]     IP: large routing tables
[ ]   IP: kernel level autoconfiguration
<M>   IP: tunneling
<M>   IP: GRE tunnels over IP
[*]     IP: broadcast GRE over IP
[*]   IP: multicast routing
[*]     IP: PIM-SM version 1 support
[*]     IP: PIM-SM version 2 support
[ ]   IP: ARP daemon support (EXPERIMENTAL)
[*]   IP: TCP Explicit Congestion Notification support
[*]   IP: TCP syncookie support (disabled per default)
```

IP: Netfilter Configuration →

```
     <M> Connection tracking (required for masq/NAT)
     <M>   FTP protocol support
     <M>   IRC protocol support
     <M> Userspace queueing via NETLINK (EXPERIMENTAL)
     <M> IP tables support (required for filtering/masq/NAT)
     <M>   limit match support
     <M>   MAC address match support
     <M>   netfilter MARK match support
```

```
      <M>   Multiple port match support
      <M>   TOS match support
      <M>   tcpmss match support
      <M>   Connection state match support
      <M>   Unclean match support (EXPERIMENTAL)
      <M>   Owner match support (EXPERIMENTAL)
      <M>   Packet filtering
      <M>     REJECT target support
      <M>     MIRROR target support (EXPERIMENTAL)
      <M>   Full NAT
      <M>     MASQUERADE target support
      <M>     REDIRECT target support
      <M>   Packet mangling
      <M>     TOS target support
      <M>     MARK target support
      <M>   LOG target support
      <M>   TCPMSS target support
      <M> ipchains (2.2-style) support
      <M> ipfwadm (2.0-style) support
IP: Virtual Server Configuration  →
      <M> virtual server support (EXPERIMENTAL)
      [ ]   IP virtual server debugging
      (16)   IPVS connection table size (the Nth power of 2)
      --- IPVS scheduler
      <M>   round-robin scheduling
      <M>   weighted round-robin scheduling
      <M>   least-connection scheduling scheduling
      <M>   weighted least-connection scheduling
      <M>   locality-based least-connection scheduling
      <M>   locality-based least-connection with replication scheduling
      <M>   destination hashing scheduling
      <M>   source hashing scheduling
      --- IPVS application helper
      <M>   FTP protocol helper
<M>   The IPv6 protocol (EXPERIMENTAL)
IPv6: Netfilter Configuration  →
      <M> IP6 tables support (required for filtering/masq/NAT)
      <M>   limit match support
```

```
      <M>   netfilter MARK match support
      <M>   Packet filtering
      <M>   Packet mangling
      <M>     MARK target support
< >   Kernel httpd acceleration (EXPERIMENTAL)
[*] Asynchronous Transfer Mode (ATM) (EXPERIMENTAL)
[*]   Classical IP over ATM
[ ]     Do NOT send ICMP if no neighbour
<M>   LAN Emulation (LANE) support
<M>     Multi-Protocol Over ATM (MPOA) support
---
<M> The IPX protocol
[ ]   IPX: Full internal IPX network
<M> Appletalk protocol support
<M> DECnet Support
[*]   DECnet: SIOCGIFCONF support
[*]   DECnet: router support (EXPERIMENTAL)
[*]     DECnet: use FWMARK value as routing key (EXPERIMENTAL)
<M> 802.1d Ethernet Bridging
< > CCITT X.25 Packet Layer (EXPERIMENTAL)
< > LAPB Data Link Driver (EXPERIMENTAL)
[ ] 802.2 LLC (EXPERIMENTAL)
[ ] Frame Diverter (EXPERIMENTAL)
< > Acorn Econet/AUN protocols (EXPERIMENTAL)
<M> WAN router
[ ] Fast switching (read help!)
[ ] Forwarding between high speed interfaces
QoS and/or fair queueing  →
      [*] QoS and/or fair queueing
      <M>   CBQ packet scheduler
      <M>   CSZ packet scheduler
      [ ]   ATM pseudo-scheduler
      <M>   The simplest PRIO pseudoscheduler
      <M>   RED queue
      <M>   SFQ queue
      <M>   TEQL queue
      <M>   TBF queue
      <M>   GRED queue
```

```
          <M>   Diffserv field marker
          <M>   Ingress Qdisc
          [*]   QoS support
          [*]     Rate estimator
          [*]   Packet classifier API
          <M>     TC index classifier
          <M>     Routing table based classifier
          <M>     Firewall based classifier
          <M>     U32 classifier
          <M>     Special RSVP classifier
          <M>     Special RSVP classifier for IPv6
          [*]     Traffic policing (needed for in/egress)
Telephony Support  →
     <M> Linux telephony support
     <M> QuickNet Internet LineJack/PhoneJack support
ATA/IDE/MFM/RLL support  →
     <*> ATA/IDE/MFM/RLL support
     IDE, ATA and ATAPI Block devices  →
          <*> Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support
          --- Please see Documentation/ide.txt for help on IDE drives
          [ ]   Use old disk-only driver on primary interface
          <*>   Include IDE/ATA-2 DISK support
          [*]     Use multi-mode by default
          <M>   PCMCIA IDE support
          <M>   Include IDE/ATAPI CDROM support
          <M>   Include IDE/ATAPI TAPE support
          <*>   Include IDE/ATAPI FLOPPY support
          <M>   SCSI emulation support
          --- IDE chipset support/bugfixes
          [*]   CMD640 chipset bugfix/support
          [ ]     CMD640 enhanced support
          [*]   ISA-PNP EIDE support
          [*]   RZ1000 chipset bugfix/support
          [*]   Generic PCI IDE chipset support
          [*]     Sharing PCI IDE interrupts support
          [*]     Generic PCI bus-master DMA support
          [ ]     Boot off-board chipsets first support
          [*]       Use PCI DMA by default when available
```

```
          [*]     Enable DMA only for disks
          [ ]       ATA Work(s) In Progress (EXPERIMENTAL)
          [*]     AEC62XX chipset support
          [*]       AEC62XX Tuning support
          [*]     ALI M15x3 chipset support
          [ ]       ALI M15x3 WDC support (DANGEROUS)
          [*]     AMD Viper support
          [*]     CMD64X chipset support
          [*]     CY82C693 chipset support
          [*]     Cyrix CS5530 MediaGX chipset support
          [*]     HPT34X chipset support
          [*]     HPT366 chipset support
          [*]     Intel PIIXn chipsets support
          [*]       PIIXn Tuning support
          [ ]     NS87415 chipset support (EXPERIMENTAL)
          [ ]     OPTi 82C621 chipset enhanced support (EXPERIMENTAL)
          [*]     PROMISE PDC202{46|62|65|67|68} support
          [ ]       Special UDMA Feature
          [ ]       Special FastTrak Feature
          [*]     ServerWorks OSB4/CSB5 chipsets support
          [*]     SiS5513 chipset support
          [*]     SLC90E66 chipset support
          [ ]     Tekram TRM290 chipset support (EXPERIMENTAL)
          [*]     VIA82CXXX chipset support
          [ ]   Other IDE chipset support
          [ ]   IGNORE word93 Validation BITS
          <M> Support for IDE Raid controllers
          <M>    Support Promise software RAID (Fasttrak(tm))
          <M>    Highpoint 370 software RAID
SCSI support  →
     <M> SCSI support
     --- SCSI support type (disk, tape, CD-ROM)
     <M>   SCSI disk support
     (40) Maximum number of SCSI disks that can be loaded as modules
     <M>   SCSI tape support
     <M>   SCSI OnStream SC-x0 tape support
     <M>   SCSI CD-ROM support
     [*]     Enable vendor-specific extensions (for SCSI CDROM)
```

```
(4) Maximum number of CDROM devices that can be loaded as modules
<M>   SCSI generic support
--- Some SCSI devices (e.g. CD jukebox) support multiple LUNs
[ ]   Enable extra checks in new queueing code
[ ]   Probe all LUNs on each SCSI device
[*]   Verbose SCSI error reporting (kernel size +=12K)
[*]   SCSI logging facility
SCSI low-level drivers  →
     <M> 3ware Hardware ATA-RAID support
     <M> 7000FASST SCSI support
     <M> ACARD SCSI support
     <M> Adaptec AHA152X/2825 support
     <M> Adaptec AHA1542 support
     <M> Adaptec AHA1740 support
     <M> Adaptec AACRAID support
     <M> Adaptec AIC7xxx support
     (253)   Maximum number of TCQ commands per device
     (15000)   Initial bus reset delay in milli-seconds
     [ ]   Build Adapter Firmware with Kernel Build
     <M> Old Adaptec AIC7xxx support
    [*]   Enable Tagged Command Queueing (TCQ) by default
    (32)   Maximum number of TCQ commands per device
    [*]   Collect statistics to report in /proc
    <M> AdvanSys SCSI support
    <M> Always IN2000 SCSI support
    <M> AM53/79C974 PCI SCSI support
    <M> AMI MegaRAID support
    <M> BusLogic SCSI support
    [ ]   Omit FlashPoint support
    <M> Compaq Fibre Channel 64-bit/66Mhz HBA support
    <M> DMX3191D SCSI support
    <M> DTC3180/3280 SCSI support
    <M> EATA ISA/EISA/PCI (DPT and generic EATA/DMA-compliant
        boards)support
    [*]   enable tagged command queuing
    [ ]   enable elevator sorting
    (16)   maximum number of queued commands
    <M> EATA-DMA [Obsolete] (DPT, NEC, AT&T, SNI, AST, Olivetti,
        Alphatronix) support
```

```
<M> EATA-PIO (old DPT PM2001, PM2012A) support
<M> Future Domain 16xx SCSI/AHA-2920A support
<M> GDT SCSI Disk Array Controller support
<M> Generic NCR5380/53c400 SCSI support
[ ]   Enable NCR53c400 extensions
(Port) NCR5380/53c400 mapping method (use Port for T130B)
<M> IBM ServeRAID support
<M> Initio 9100U(W) support
<M> Initio INI-A100U2W support
<M> IOMEGA parallel port (ppa - older drives)
<M> IOMEGA parallel port (imm - newer drives)
[ ]   ppa/imm option - Use slow (but safe) EPP-16
[ ]   ppa/imm option - Assume slow parport control register
<M> NCR53c406a SCSI support
<M> NCR53c7,8xx SCSI support
[ ]   always negotiate synchronous transfers
[*]   allow FAST-SCSI [10MHz]
[*]   allow DISCONNECT
<M> NCR53C8XX SCSI support
<M> SYM53C8XX SCSI support
(8)   default tagged command queue depth
(32)   maximum number of queued commands
(20)   synchronous transfers frequency in MHz
[ ]   enable profiling
[ ]   use normal IO
[ ]   include support for the NCR PQS/PDS SCSI card
[ ]   assume boards are SYMBIOS compatible (EXPERIMENTAL)
<M> PAS16 SCSI support
<M> PCI2000 support
<M> PCI2220i support
<M> PSI240i support
<M> Qlogic FAS SCSI support
<M> Qlogic ISP SCSI support
<M> Qlogic ISP FC SCSI support
<M> Qlogic QLA 1280 SCSI support
<M> Qlogic QLA 2100 FC SCSI support
<M> Seagate ST-02 and Future Domain TMC-8xx SCSI support
<M> Simple 53c710 SCSI support (Compaq, NCR machines)
<M> Symbios 53c416 SCSI support
```

```
        <M> Tekram DC390(T) and Am53/79C974 SCSI support
        [ ]   _omit_ support for non-DC390 adapters
        <M> Trantor T128/T128F/T228 SCSI support
        <M> UltraStor 14F/34F support
        [ ]   enable elevator sorting
        (8)   maximum number of queued commands
        <M> UltraStor SCSI support
        <M> SCSI debugging host simulator (EXPERIMENTAL)
        <M> iSCSI support
    PCMCIA SCSI adapter support  →
         [*] PCMCIA SCSI adapter support
         <M>   Adaptec AHA152X PCMCIA support
        <M>   Future Domain PCMCIA support
        <M>   NinjaSCSI-3 / NinjaSCSI-32Bi (16bit) PCMCIA support
        <M>   Qlogic PCMCIA support
Fusion MPT device support  →
    <M> Fusion MPT (base + ScsiHost) drivers
    --- (ability to boot linux kernel from Fusion device is DISABLED!)
    < >   Enhanced SCSI error reporting
    <M>   Fusion MPT misc device (ioctl) driver
    <M>   Fusion MPT LAN driver
IEEE 1394 (FireWire) support (EXPERIMENTAL)  →
    <M> IEEE 1394 (FireWire) support (EXPERIMENTAL)
    --- Device Drivers
    <M>   Texas Instruments PCILynx support
    [ ]     Use PCILynx local RAM
    [*]     Support for non-IEEE1394 local ports
    <M>   OHCI-1394 support
    --- Protocol Drivers
    <M>   OHCI-1394 Video support
    <M>   SBP-2 support (Harddisks etc.)
    <M>   Raw IEEE1394 I/O support
    [ ] Excessive debugging output
I2O device support  →
    <M> I2O support
    <M>   I2O PCI support
    <M>   I2O Block OSM
    <M>   I2O LAN OSM
```

```
    <M>   I2O SCSI OSM
    <M>   I2O /proc support
Network device support  →
    [*] Network device support
    ARCnet devices  →
        < > ARCnet support
    Appletalk devices  →
        [*] Appletalk interfaces support
        <M>   Apple/Farallon LocalTalk PC support
        <M>   COPS LocalTalk PC support
        [*]     Dayna firmware support
        [*]     Tangent firmware support
        <M>   Appletalk-IP driver support
        [*]     IP to Appletalk-IP Encapsulation support
        [*]     Appletalk-IP to IP Decapsulation support
    <M> Dummy net driver support
    <M> Bonding driver support
    <M> EQL (serial line load balancing) support
    <M> Universal TUN/TAP device driver support
    <M> Ethertap network tap (OBSOLETE)
    <M> General Instruments Surfboard 1000
    Ethernet (10 or 100Mbit)  →
        [*] Ethernet (10 or 100Mbit)
        <M>   Sun Happy Meal 10/100baseT support
        <M>   Sun GEM support
        [*]   3COM cards
        <M>     3c501 "EtherLink" support
        <M>     3c503 "EtherLink II" support
        <M>     3c505 "EtherLink Plus" support
        <M>     3c507 "EtherLink 16" support (EXPERIMENTAL)
        <M>     3c509/3c529 (MCA)/3c579 "EtherLink III" support
        <M>     3c515 ISA "Fast EtherLink"
        <M>     3c590/3c900 series (592/595/597) "Vortex/Boomerang"
                support
        <M>   AMD LANCE and PCnet (AT1500 and NE2100) support
        [*]   Western Digital/SMC cards
        <M>     WD80*3 support
        <M>     SMC Ultra support
```

```
<M>     SMC Ultra32 EISA support
< >     SMC 9194 support
[*]   Racal-Interlan (Micom) NI cards
<M>     NI5010 support (EXPERIMENTAL)
<M>     NI5210 support
<M>     NI6510 support
<M>   AT1700/1720 support (EXPERIMENTAL)
<M>   DEPCA, DE10x, DE200, DE201, DE202, DE422 support
<M>   HP 10/100VG PCLAN (ISA, EISA, PCI) support
[*]   Other ISA cards
<M>     Cabletron E21xx support
<M>     EtherWORKS 3 (DE203, DE204, DE205) support
<M>     EtherExpress 16 support
<M>    EtherExpressPro support/EtherExpress 10 (i82595) support
<M>     HP PCLAN+ (27247B and 27252A) support
<M>     HP PCLAN (27245 and other 27xxx series) support
<M>     LP486E on board Ethernet
<M>     ICL EtherTeam 16i/32 support
<M>     NE2000/NE1000 support
[*]     EISA, VLB, PCI and on board controllers
<M>     AMD PCnet32 PCI support
<M>     Adaptec Starfire support (EXPERIMENTAL)
<M>     Ansel Communications EISA 3200 support (EXPERIMENTAL)
<M>     Apricot Xen-II on board Ethernet
<M>     CS89x0 support
<M>     DECchip Tulip (dc21x4x) PCI support
[ ]     New bus configuration (EXPERIMENTAL)
[*]     Use PCI shared mem for NIC registers
<M>     Generic DECchip & DIGITAL EtherWORKS PCI/EISA
<M>     Digi Intl. RightSwitch SE-X support
<M>     Davicom DM910x/DM980x support
<M>     EtherExpressPro/100 support
<M>    Intel PRO100/EtherExpressPro/100 support (alternate
       driver)
<M>     Mylex EISA LNE390A/B support (EXPERIMENTAL)
< >     Myson MTD-8xx PCI Ethernet support
<M>     National Semiconductor DP8381x series PCI Ethernet
        support
<M>     PCI NE2000 and clones support (see help)
```

```
      <M>     Novell/Eagle/Microdyne NE3210 EISA support
              (EXPERIMENTAL)
      <M>     Racal-Interlan EISA ES3210 support (EXPERIMENTAL)
      <M>     RealTek RTL-8139 PCI Fast Ethernet Adapter support
      [ ]     Use PIO instead of MMIO
      [ ]     Support for automatic channel equalization
              (EXPERIMENTAL)
      [*]     Support for older RTL-8129/8130 boards
      <M>     SiS 900/7016 PCI Fast Ethernet Adapter support
      <M>     SMC EtherPower II
      <M>     Sundance Alta support
      <M>     TI ThunderLAN support
      <M>     VIA Rhine support
      <M>     Winbond W89c840 Ethernet support
      [*]     Pocket and portable adapters
      <M>     AT-LAN-TEC/RealTek pocket adapter support
      <M>     D-Link DE600 pocket adapter support
      <M>     D-Link DE620 pocket adapter support
Ethernet (1000 Mbit)  →
      <M> Intel PRO/1000 support
      <M> Alteon AceNIC/3Com 3C985/NetGear GA620 Gigabit support
      [ ]   Omit support for old Tigon I based AceNICs
      <M> D-Link DL2000-based Gigabit Ethernet support
      <M> National Semiconductor DP83820 support
      <M> Broadcom BCM5700 support
      <M> Packet Engines Hamachi GNIC-II support
      <M> Packet Engines Yellowfin Gigabit-NIC support (EXPERIMENTAL)
      <M> SysKonnect SK-98xx support
[*] FDDI driver support
<M>   Digital DEFEA and DEFPA adapter support
<M>   SysKonnect FDDI PCI support
[ ] HIPPI driver support (EXPERIMENTAL)
<M> PLIP (parallel port) support
<M> PPP (point-to-point protocol) support
[*]   PPP multilink support (EXPERIMENTAL)
[*]   PPP filtering
<M>   PPP support for async serial ports
<M>   PPP support for sync tty ports
<M>   PPP Deflate compression
```

```
<M>   PPP BSD-Compress compression
< >   PPP over Ethernet (EXPERIMENTAL)
<M> SLIP (serial line) support
[*]   CSLIP compressed headers
[*]   Keepalive and linefill
[*]   Six bit SLIP encapsulation
```

Wireless LAN (non-hamradio) →

```
     <M> CIPE (Crypto IP Encapsulation)
     [*] Wireless LAN (non-hamradio)
     <M>   STRIP (Metricom starmode radio IP)
     <M>   AT&T WaveLAN & DEC RoamAbout DS support
     <M>   Aironet Arlan 655 & IC2200 DS support
     <M>   Aironet 4500/4800 series adapters
     <M>    Aironet 4500/4800 ISA/PCI/PNP/365 support
     [*]      Aironet 4500/4800 PNP support
     [*]      Aironet 4500/4800 PCI support
     [*]      Aironet 4500/4800 ISA broken support (EXPERIMENTAL)
     [*]      Aironet 4500/4800 I365 broken support (EXPERIMENTAL)
     <M>    Aironet 4500/4800 PROC interface
     <M>   Cisco/Aironet 34X/35X/4500/4800 ISA and PCI cards
     --- Wireless Pcmcia cards support
     <M>   Hermes support (Orinoco/WavelanIEEE/PrismII/Symbol
           802.11b cards)
     <M>   Cisco/Aironet 34X/35X/4500/4800 PCMCIA cards
```

Token Ring devices →

```
     [*] Token Ring driver support
     <M>   IBM Tropic chipset based adapter support
     <M>   IBM Olympic chipset PCI adapter support
     <M>   IBM Lanstreamer chipset PCI adapter support
     <M>   Generic TMS380 Token Ring ISA/PCI adapter support
     <M>     Generic TMS380 PCI support
     <M>     Generic TMS380 ISA support
     <M>     Madge Smart 16/4 PCI Mk2 support
     <M>   SMC ISA/MCA adapter support
[*] Fibre Channel driver support
<M>   Interphase 5526 Tachyon chipset based adapter support
<M> Red Creek Hardware VPN (EXPERIMENTAL)
<M> Traffic Shaper (EXPERIMENTAL)
```

```
Wan interfaces  →
     [*] Wan interfaces support
     <M>   Comtrol Hostess SV-11 support
     <M>   COSA/SRP sync serial boards support
     < >   MultiGate (COMX) synchronous serial boards support
     < >   Etinc PCISYNC serial board support (EXPERIMENTAL)
     <M>   FarSync T-Series X.21 (and V.35/V.24) cards
     < >   LanMedia Corp. SSI/V.35, T1/E1, HSSI, T3 boards
     <M>   Sealevel Systems 4021 support
     < >   SyncLink HDLC/SYNCPPP support
     < >   Generic HDLC driver
     <M>   Frame relay DLCI support
     (24)     Max open DLCI
     (8)     Max DLCI per device
     <M>     SDLA (Sangoma S502/S508) support
     [*]   WAN router drivers
     <M>     Sangoma WANPIPE(tm) multiprotocol cards
     [*]       WANPIPE Cisco HDLC support
     [*]       WANPIPE Frame Relay support
     [*]       WANPIPE X.25 support
     [*]       WANPIPE PPP support
     [*]       WANPIPE Multi-Port PPP support
     <M>     Cyclom 2X(tm) cards (EXPERIMENTAL)
     [*]       Cyclom 2X X.25 support
     <M>   Granch SBNI12 Leased Line adapter support
     [*]     Multiple line feature support
PCMCIA network device support  →
     [*] PCMCIA network device support
     <M>   3Com 3c589 PCMCIA support
     <M>   3Com 3c574 PCMCIA support
     <M>   Fujitsu FMV-J18x PCMCIA support
     <M>   NE2000 compatible PCMCIA support
     <M>   New Media PCMCIA support
     <M>   SMC 91Cxx PCMCIA support
     <M>   Xircom 16-bit PCMCIA support
     <M>   broken NS8390-cards support
     <M>   IBM PCMCIA tokenring adapter support
     <M>   Xircom CardBus support (new driver)
```

```
        <M>   Xircom Tulip-like CardBus support (old driver)
        [*]   Pcmcia Wireless LAN
        <M>     Aviator/Raytheon 2.4MHz wireless support
        <M>     Xircom Netwave AirSurfer wireless support
        <M>     AT&T/Lucent Wavelan wireless support
        <M>     AT&T/Lucent Wavelan IEEE 802.11 wireless support
        <M>     Aironet 4500/4800 PCMCIA support
        <M>     Alternate Aironet 4500/4800 PCMCIA support
```

ATM drivers →

```
        <M> ATM over TCP
        <M> Efficient Networks ENI155P
        [ ]   Enable extended debugging
        [ ]   Fine-tune burst settings
        <M> Fujitsu FireStream (FS50/FS155)
        <M> ZeitNet ZN1221/ZN1225
        [ ]   Enable extended debugging
        [*]   Enable usec resolution timestamps
        <M> IDT 77201 (NICStAR) (ForeRunnerLE)
        [*]   Use suni PHY driver (155Mbps)
        [*]   Use IDT77015 PHY driver (25Mbps)
        <M> Madge Ambassador (Collage PCI 155 Server)
        [ ]   Enable debugging messages
        <M> Madge Horizon [Ultra] (Collage PCI 25 and Collage PCI 155
            Client)
        [ ]   Enable debugging messages
        <M> Interphase ATM PCI x575/x525/x531
        [ ]   Enable debugging messages
        <M> FORE Systems 200E-series
        [*]   PCA-200E support
        [*]    Use default PCA-200E firmware (normally enabled)
        (16)   Maximum number of tx retries
        (0)   Debugging level (0-3)
```

Amateur Radio support →

```
     [ ] Amateur Radio support
```

IrDA (infrared) support →

```
     <M> IrDA subsystem support
     --- IrDA protocols
     <M>   IrLAN protocol
     <M>   IrNET protocol
```

```
<M>   IrCOMM protocol
[*]   Ultra (connectionless) protocol
[*]   IrDA protocol options
---   IrDA options
[*]     Cache last LSAP
[*]     Fast RRs
[ ]     Debug information
Infrared-port device drivers  →
     --- SIR device drivers
     <M> IrTTY (uses Linux serial driver)
     <M> IrPORT (IrDA serial driver)
     --- Dongle support
     [*] Serial dongle support
     <M>   ESI JetEye PC dongle
     <M>   ACTiSYS IR-220L and IR220L+ dongle
     <M>   Tekram IrMate 210B dongle
     <M>   Greenwich GIrBIL dongle
     <M>   Parallax LiteLink dongle
     <M>   Old Belkin dongle
     --- FIR device drivers
     <M> IrDA USB dongles (Experimental)
     <M> NSC PC87108/PC87338
     <M> Winbond W83977AF (IR)
     <M> Toshiba Type-O IR Port
     <M> SMC IrCC (Experimental)
     <M> ALi M5123 FIR (Experimental)
ISDN subsystem  →
     <M> ISDN support
     [*]   Support synchronous PPP
     [*]     Use VJ-compression with synchronous PPP
     [*]     Support generic MP (RFC 1717)
     <M>     Support BSD compression
     [*]   Support audio via ISDN
     [*]     Support AT-Fax Class 1 and 2 commands
     ISDN feature submodules  →
          <M> isdnloop support
          < > Support isdn diversion services
     --- low-level hardware drivers
```

Passive ISDN cards →

```
<M> HiSax SiemensChipSet driver support
---   D-channel protocol features
[*]   HiSax Support for EURO/DSS1
[*]     Support for german chargeinfo
[ ]     Disable sending complete
[ ]     Disable sending low layer compatibility
[ ]     Disable keypad protocol option
[*]   HiSax Support for german 1TR6
[*]   HiSax Support for US NI1
---   HiSax supported cards
[*]   Teles 16.0/8.0
[*]   Teles 16.3 or PNP or PCMCIA
[*]   Teles PCI
[*]   Teles S0Box
[*]   AVM A1 (Fritz)
[*]   AVM PnP/PCI (Fritz!PnP/PCI)
[*]   AVM A1 PCMCIA (Fritz)
[*]   Elsa cards
[*]   ITK ix1-micro Revision 2
[*]   Eicon.Diehl Diva cards
[*]   ASUSCOM ISA cards
[*]   TELEINT cards
[*]   HFC-S based cards
[*]   Sedlbauer cards
[*]   USR Sportster internal TA
[*]   MIC card
[*]   NETjet card
[*]   NETspider U card
[*]   Niccy PnP/PCI card
[*]   Siemens I-Surf card
[*]   HST Saphir card
[*]   Telekom A4T card
[*]   Scitel Quadro card
[*]   Gazel cards
[*]   HFC PCI-Bus cards
[*]   Winbond W6692 based cards
[*]   HFC-S+, HFC-SP, HFC-PCMCIA cards
```

```
        <M> Sedlbauer PCMCIA cards
        <M> ELSA PCMCIA MicroLink cards
        <M> AVM A1 PCMCIA cards
    Active ISDN cards  →
        <M> ICN 2B and 4B support
        <M> PCBIT-D support
        < > Spellcaster support
        < > IBM Active 2000 support
        [ ] Eicon active card support
        <M> Auvertech TurboPAM support
        <M> CAPI2.0 support
        [*]   Verbose reason code reporting (kernel size +=7K)
        [*]   CAPI2.0 Middleware support (EXPERIMENTAL)
        <M>   CAPI2.0 /dev/capi support
        [*]     CAPI2.0 filesystem support
        <M>   CAPI2.0 capidrv interface support
        <M>   AVM B1 ISA support
        <M>   AVM B1 PCI support
        [*]     AVM B1 PCI V4 support
        <M>   AVM T1/T1-B ISA support
        <M>   AVM B1/M1/M2 PCMCIA support
        <M>   AVM B1/M1/M2 PCMCIA cs module
        <M>   AVM T1/T1-B PCI support
        <M>   AVM C4/C2 support
        <M>   Hypercope HYSDN cards (Champ, Ergo, Metro) support
              (module only)
        [*]     HYSDN CAPI 2.0 support
Old CD-ROM drivers (not SCSI, not IDE)  →
    [*] Support non-SCSI/IDE/ATAPI CDROM drives
    <M>   Aztech/Orchid/Okano/Wearnes/TXC/CyDROM  CDROM support
    <M>   Goldstar R420 CDROM support
    <M>   Matsushita/Panasonic/Creative, Longshine, TEAC CDROM support
    <M>   Mitsumi (standard) [no XA/Multisession] CDROM support
    (11) MCD IRQ
    (300) MCD I/O base
    <M>   Mitsumi [XA/MultiSession] CDROM support
    <M>   Optics Storage DOLPHIN 8000AT CDROM support
    <M>   Philips/LMS CM206 CDROM support
    <M>   Sanyo CDR-H94A CDROM support
```

```
    <M>   ISP16/MAD16/Mozart soft configurable cdrom interface support
    <M>   Sony CDU31A/CDU33A CDROM support
    <M>   Sony CDU535 CDROM support
```

Input core support →

```
    <M> Input core support
    <M>   Keyboard support
    <M>   Mouse support
    (1024)    Horizontal screen resolution
    (768)    Vertical screen resolution
    <M>   Joystick support
    <M>   Event interface support
```

Character devices →

```
    [*] Virtual terminal
    <M>    ECC memory monitoring
    [*]   Support for console on virtual terminal
    <*> Standard/generic (8250/16550 and compatible UARTs) serial
        support
    [*]   Support for console on serial port
    [*] Extended dumb serial driver options
    [*]   Support more than 4 serial ports
    [*]   Support for sharing serial interrupts
    [ ]   Autodetect IRQ on standard ports (unsafe)
    [*]   Support special multiport boards
    [ ]   Support the Bell Technologies HUB6 card
    [*] Non-standard serial port support
    <M>   Computone IntelliPort Plus serial support
    <M>   Comtrol Rocketport support
    <M>   Cyclades async mux support
    [ ]     Cyclades-Z interrupt mode operation (EXPERIMENTAL)
    <M>   Digiboard Intelligent Async Support
    <M>   Hayes ESP serial port support
    <M>   Moxa Intellio support
    <M>   Moxa SmartIO support
    <M>   Multi-Tech multiport card support (EXPERIMENTAL)
    <M>   Microgate SyncLink card support
    <M>   HDLC line discipline support
    <M>   SDL RISCom/8 card support
    <M>   Specialix IO8+ card support
    [*]   Specialix DTR/RTS pin is RTS
```

```
<M>   Specialix SX (and SI) card support
< >   Specialix RIO system support
[*]   Stallion multiport serial support
<M>     Stallion EasyIO or EC8/32 support
<M>     Stallion EC8/64, ONboard, Brumby support
[*] Unix98 PTY support
(2048) Maximum number of Unix98 PTYs in use (0-2048)
<M> Parallel printer support
[*]   Support for console on line printer
<M> Support for user-space parallel port device drivers
```

I2C support →

```
     <M> I2C support
     <M> I2C bit-banging interfaces
     <M>   Philips style parallel port adapter
     <M>   ELV adapter
     <M>   Velleman K9000 adapter
     <M> I2C PCF 8584 interfaces
     <M>   Elektor ISA card
     [*] I2C mainboard interfaces
     <M>   Acer Labs ALI 1533 and 1543C
     <M>   Apple Hydra Mac I/O
     <M>   AMD 756
     <M>   Intel 82801AA, 82801AB and 82801BA
     <M>   Intel i810AA, i810AB and i815
     <M>   Intel 82371AB PIIX4(E)
     <M>   VIA Technologies, Inc. VT82C586B
     <M>   VIA Technologies, Inc. VT596A/B
     <M>   Voodoo3 I2C interface
     <M>   Pseudo ISA adapter (for hardware sensors modules)
     <M> I2C device interface
```

Hardware sensors support →

```
     <M> Hardware sensors support
     <M>   Analog Devices ADM1021 and compatibles
     <M>   Analog Devices ADM1025
     <M>   Analog Devices ADM9240 and compatibles
     <M>   Genesys Logic GL518SM
     <M>   Genesys Logic GL520SM
     <M>   National Semiconductors LM75
```

```
      <M>   National Semiconductors LM78
      <M>   National Semiconductors LM80
      <M>   National Semiconductors LM87
      <M>   Silicon Integrated Systems Corp. SiS5595
      <M>   Texas Instruments THMC50 and compatibles
      <M>   VIA 686a Integrated Hardware Monitor
      <M>   Winbond W83781D, W83782D and W83783S
      [*] Other I2C devices
      <M>   Brooktree BT869 Video Modulator
      <M>   DDC Monitor EDID EEPROM
      <M>   EEprom (DIMM) reader
      <M>   Linear Technologies LTC1710
      <M>   Matrix-Orbital LCD Displays
```

Mice →

```
      <M> Bus Mouse Support
      <M>   ATIXL busmouse support
      <M>   Logitech busmouse support
      <M>   Microsoft busmouse support
      <*> Mouse Support (not serial and bus mice)
      [*]   PS/2 mouse (aka "auxiliary device") support
      <M>   C&T 82C710 mouse port support (as on TI Travelmate)
      <M>   PC110 digitizer pad support
```

Joysticks →

```
      <M> Game port support
      <M>   Classic ISA/PnP gameports
      <M>   PDPI Lightning 4 gamecard
      <M>   Aureal Vortex and Trident 4DWave gameports
      <M>   Crystal SoundFusion gameports
      <M>   SoundBlaster Live! gameports
      <M> Serial port device support
      <M>   Serial port input line discipline
      --- Joysticks
      <M>   Classic PC analog joysticks and gamepads
      <M>   Assasin 3D and MadCatz Panther devices
      <M>   Logitech ADI digital joysticks and gamepads
      <M>   Creative Labs Blaster Cobra gamepad
      <M>   Genius Flight2000 Digital joysticks and gamepads
      <M>   Gravis GrIP joysticks and gamepads
```

```
      <M>   InterAct digital joysticks and gamepads
      <M>   ThrustMaster DirectConnect joysticks and gamepads
      <M>   Microsoft SideWinder digital joysticks and gamepads
      <M>   I-Force USB joysticks and wheels
      <M>   I-Force Serial joysticks and wheels
      <M>   Logitech WingMan Warrior joystick
      <M>   LogiCad3d Magellan/SpaceMouse 6dof controller
      <M>   SpaceTec SpaceOrb/Avenger 6dof controller
      <M>   SpaceTec SpaceBall 4000 FLX 6dof controller
      <M>   Gravis Stinger gamepad
      <M>   Multisystem, Sega Genesis, Saturn joysticks and gamepads
      <M>   Multisystem, NES, SNES, N64, PSX joysticks and gamepads
      <M>   Multisystem joysticks via TurboGraFX device
            < > QIC-02 tape support
Watchdog Cards  →
      [*] Watchdog Timer Support
      [ ]   Disable watchdog shutdown on close
      <M>   Software Watchdog
      <M>   WDT Watchdog timer
      <M>   WDT PCI Watchdog timer
      [ ]     WDT501 features
      <M>   Berkshire Products PC Watchdog
      <M>   Acquire SBC Watchdog Timer
      <M>   Advantech SBC Watchdog Timer
      < >   SBC-60XX Watchdog Timer
      <M>   W83877F (EMACS) Watchdog Timer
      < >   Mixcom Watchdog
      <M>   Intel i810 TCO timer / Watchdog
      <M>   ZF MachZ Watchdog
< > Intel i8x0 Random Number Generator support
<M> /dev/nvram support
<*> Enhanced Real Time Clock Support
<M> Double Talk PC internal speech card support
<M> Siemens R3964 line discipline
< > Applicom intelligent fieldbus card support
<M> Sony Vaio Programmable I/O Control Device support
Ftape, the floppy tape device driver  →
      <M> Ftape (QIC-80/Travan) support
```

```
          <M>   Zftape, the VFS interface
          (10240)   Default block size
          ---   The compressor will be built as a module only!
          (3)   Number of ftape buffers (EXPERIMENTAL)
          [ ]   Enable procfs status report (+2kb)
          (Normal) Debugging output
          --- Hardware configuration
          (Standard) Floppy tape controllers
          (8)   Default FIFO threshold (EXPERIMENTAL)
          (2000)   Maximal data rate to use (EXPERIMENTAL)
     <M> /dev/agpgart (AGP Support)
     [*]   Intel 440LX/BX/GX and I815/I840/I850 support
     [*]   Intel I810/I815 (on-board) support
     [*]   VIA chipset support
     [*]   AMD Irongate support
     [*]   Generic SiS support
     [*]   ALI chipset support
     [*]   Serverworks LE/HE support
     [*] Direct Rendering Manager (XFree86 DRI support)
     [*]   Build drivers for new (XFree 4.1) DRM
     <M>   3dfx Banshee/Voodoo3+
     <M>   3dlabs GMX 2000
     <M>   ATI Rage 128
     <M>   ATI Radeon
     <M>   Intel I810
     <M>   Matrox g200/g400
     PCMCIA character devices  →
          <M> PCMCIA serial device support
Multimedia devices  →
     <M> Video For Linux
     Video For Linux  →
          [*]   V4L information in proc filesystem
          <M>   I2C on parallel port
          --- Video Adapters
          <M>   BT848 Video For Linux
          <M>   Mediavision Pro Movie Studio Video For Linux
          <M>   Quickcam BW Video For Linux
          <M>   QuickCam Colour Video For Linux (EXPERIMENTAL)
```

```
        <M>   Winbond W9966CF Webcam Video For Linux (EXPERIMENTAL)
        <M>   CPiA Video For Linux
        <M>     CPiA Parallel Port Lowlevel Support
        <M>     CPiA USB Lowlevel Support
        <M>   SAA5249 Teletext processor
        <M>   SAB3036 tuner
        <M>   Stradis 4:2:2 MPEG-2 video driver  (EXPERIMENTAL)
        <M>   Zoran ZR36057/36060 Video For Linux
        <M>   Zoran ZR36120/36125 Video For Linux
        <M>   Sony Vaio Picturebook Motion Eye Video For Linux
    Radio Adapters  →
        <M>   ADS Cadet AM/FM Tuner
        <M>   AIMSlab RadioTrack (aka RadioReveal) support
        <M>   AIMSlab RadioTrack II support
        <M>   Aztech/Packard Bell Radio
        <M>   GemTek Radio Card support
        <M>   GemTek PCI Radio Card support
        <M>   Guillemot MAXI Radio FM 2000 radio
        <M>   Maestro on board radio
        <M>   miroSOUND PCM20 radio
        <M>     miroSOUND PCM20 radio RDS user interface (EXPERIMENTAL)
        <M>   SF16FMI Radio
        <M>   TerraTec ActiveRadio ISA Standalone
        <M>   Trust FM radio card
        <M>   Typhoon Radio (a.k.a. EcoRadio)
        [*]     Support for /proc/radio-typhoon
        <M>   Zoltrix Radio
Crypto Hardware support  →
    <M> Crypto Hardware Accelerator Support
    <M>   Broadcom 5820 SSL accelerator support
File systems  →
    [*] Quota support
    <M> Kernel automounter support
    <M> Kernel automounter version 4 support (also supports v3)
    <M> Reiserfs support
    [ ]   Have reiserfs do extra internal checking
    < > ADFS file system support
    < > Amiga FFS file system support (EXPERIMENTAL)
```

```
<M> Apple Macintosh file system support (EXPERIMENTAL)
<M> BFS file system support (EXPERIMENTAL)
<M> CMS file system support (EXPERIMENTAL)
<M> Ext3 journalling file system support (EXPERIMENTAL)
[ ]   JBD (ext3) debugging support
[ ] Buffer Head tracing (DEBUG)
<M> DOS FAT fs support
<M>   MSDOS fs support
<M>     UMSDOS: Unix-like file system on top of standard MSDOS fs
<M>   VFAT (Windows-95) fs support
< > EFS file system support (read only) (EXPERIMENTAL)
<M> Compressed ROM file system support
[*] Virtual memory file system support (former shm fs)
<M> Simple RAM-based file system support
<*> ISO 9660 CDROM file system support
[*]   Microsoft Joliet CDROM extensions
<M> Minix fs support
<M> FreeVxFS file system support (VERITAS VxFS(TM) compatible)
< > NTFS file system support (read only)
< > OS/2 HPFS file system support
[*] /proc file system support
[ ] /dev file system support (EXPERIMENTAL)
[*] /dev/pts file system for Unix98 PTYs
< > QNX4 file system support (read only) (EXPERIMENTAL)
<M> ROM file system support
<*> Second extended fs support
<M> System V/Xenix/V7/Coherent file system support
<M> UDF file system support (read only)
[ ]   UDF write support (DANGEROUS)
<M> UFS file system support (read only)
[ ]   UFS file system write support (DANGEROUS)
```

Network File Systems →

```
     <M> Coda file system support (advanced network fs)
     <M> NFS file system support
     [*]   Provide NFSv3 client support
     <M> NFS server support
     [*]   Provide NFSv3 server support
     <M> SMB file system support (to mount Windows shares etc.)
```

```
    [ ]   Use a default NLS
    <M> NCP file system support (to mount NetWare volumes)
    [*]   Packet signatures
    [*]   Proprietary file locking
    [*]   Clear remove/delete inhibit when needed
    [*]   Use NFS namespace if available
    [*]   Use LONG (OS/2) namespace if available
    [*]   Lowercase DOS filenames
    [*]   Use Native Language Support
    [*]   Enable symbolic links and execute flags
```

Partition Types →

```
    [*] Advanced partition selection
    [ ]   Acorn partition support
    [*]   Alpha OSF partition support
    [ ]   Amiga partition table support
    [ ]   Atari partition table support
    [ ]   Macintosh partition map support
    [*]   PC BIOS (MSDOS partition tables) support
    [*]     BSD disklabel (FreeBSD partition tables) support
    [*]     Minix subpartition support
    [*]     Solaris (x86) partition table support
    [*]     Unixware slices support
    [ ]   Windows Logical Disk Manager (Dynamic Disk) support
    [*]   SGI partition support
    [ ]   Ultrix partition table support
    [*]   Sun partition tables support
```

Native Language Support →

```
         Default NLS Option: "iso8859-1"
    <M> Codepage 437 (United States, Canada)
    <M> Codepage 737 (Greek)
    <M> Codepage 775 (Baltic Rim)
    <M> Codepage 850 (Europe)
    <M> Codepage 852 (Central/Eastern Europe)
    <M> Codepage 855 (Cyrillic)
    <M> Codepage 857 (Turkish)
    <M> Codepage 860 (Portuguese)
    <M> Codepage 861 (Icelandic)
    <M> Codepage 862 (Hebrew)
```

```
        <M> Codepage 863 (Canadian French)
        <M> Codepage 864 (Arabic)
        <M> Codepage 865 (Norwegian, Danish)
        <M> Codepage 866 (Cyrillic/Russian)
        <M> Codepage 869 (Greek)
        <M> Simplified Chinese charset (CP936, GB2312)
        <M> Traditional Chinese charset (Big5)
        <M> Japanese charsets (Shift-JIS, EUC-JP)
        <M> Korean charset (CP949, EUC-KR)
        <M> Thai charset (CP874, TIS-620)
        <M> Hebrew charsets (ISO-8859-8, CP1255)
        <M> Windows CP1251 (Bulgarian, Belarusian)
        <M> NLS ISO 8859-1  (Latin 1; Western European Languages)
        <M> NLS ISO 8859-2  (Latin 2; Slavic/Central European
            Languages)
        <M> NLS ISO 8859-3  (Latin 3; Esperanto, Galician, Maltese,
            Turkish)
        <M> NLS ISO 8859-4  (Latin 4; old Baltic charset)
        <M> NLS ISO 8859-5  (Cyrillic)
        <M> NLS ISO 8859-6  (Arabic)
        <M> NLS ISO 8859-7  (Modern Greek)
        <M> NLS ISO 8859-9  (Latin 5; Turkish)
        <M> NLS ISO 8859-13 (Latin 7; Baltic)
        <M> NLS ISO 8859-14 (Latin 8; Celtic)
        <M> NLS ISO 8859-15 (Latin 9; Western European Languages with
            Euro)
        <M> NLS KOI8-R (Russian)
        <M> NLS KOI8-U/RU (Ukrainian, Belarusian)
        <M> NLS UTF8
Console drivers  →
    [*] VGA text console
    [*] Video mode selection support
    [ ] Ignore bad video mode selections
    <M> MDA text console (dual-headed) (EXPERIMENTAL)
    Frame-buffer support  →
        [*] Support for frame buffer devices (EXPERIMENTAL)
        <M>   nVidia Riva support (EXPERIMENTAL)
        <M>   Cirrus Logic support (EXPERIMENTAL)
        <M>   Permedia2 support (EXPERIMENTAL)
        [ ]     enable FIFO disconnect feature
```

```
[*]     generic Permedia2 PCI board support
< >   Cyber2000 support
[*]   VESA VGA graphics console
< >   VGA 16-color graphics console
<M>   Hercules mono graphics console (EXPERIMENTAL)
[ ]   Epson 1355 framebuffer support
<M>   Matrox acceleration (EXPERIMENTAL)
[*]     Millennium I/II support
[*]     Mystique support
[*]     G100/G200/G400/G450 support
<M>       Matrox I2C support
< >       G400 second head support
< >       G450 second head support
[*]     Multihead support
<M>   ATI Mach64 display support (EXPERIMENTAL)
[*]     Mach64 GX support (EXPERIMENTAL)
[*]     Mach64 CT/VT/GT/LT (incl. 3D RAGE) support
[*]       Sony Vaio C1VE 1024x480 LCD support
<M>   ATI Radeon display support (EXPERIMENTAL)
<M>   ATI Rage128 display support (EXPERIMENTAL)
< >   SIS acceleration (EXPERIMENTAL)
<M>   3Dfx Banshee/Voodoo3 display support (EXPERIMENTAL)
< >   Virtual Frame Buffer support (ONLY FOR TESTING!)
[ ]   Advanced low level driver options
[ ]   Support only 8 pixels wide fonts
[ ]   Select compiled-in fonts
```

Sound →

```
<M> Sound card support
<M>   C-Media PCI (CMI8338/8738)
[*]     Enable legacy FM
(388)     FM I/O 388, 3C8, 3E0, 3E8
[*]     Enable legacy MPU-401
(330)     MPU-401 I/O 330, 320, 310, 300
[*]     Enable joystick
[*]     Support CMI8738 based audio cards
[ ]       Inverse S/PDIF in for CMI8738
[*]       Enable S/PDIF loop for CMI8738
(2)       Number of speakers 2, 4, 5, 6
<M>   Creative SBLive! (EMU10K1)
<M>   Crystal SoundFusion (CS4280/461x)
```

```
<M>   Crystal Sound CS4281
<M>   Ensoniq AudioPCI (ES1370)
<M>   Creative Ensoniq AudioPCI 97 (ES1371)
<M>   ESS Technology Solo1
<M>   ESS Maestro, Maestro2, Maestro2E driver
<M>   ESS Maestro3/Allegro driver (EXPERIMENTAL)
<M>   Intel ICH (i8xx) audio support
<M>   S3 SonicVibes
<M>   bt878 audio dma
<M>   Trident 4DWave DX/NX, SiS 7018 or ALi 5451 PCI Audio Core
<M>   Support for Turtle Beach MultiSound Classic, Tahiti, Monterey
      Full pathname of MSNDINIT.BIN firmware file:
      "/etc/sound/msndinit.bin"
      Full pathname of MSNDPERM.BIN firmware file:
      "/etc/sound/msndperm.bin"
<M>   Support for Turtle Beach MultiSound Pinnacle, Fiji
      Full pathname of PNDSPINI.BIN firmware file:
      "/etc/sound/pndspini.bin"
      Full pathname of PNDSPERM.BIN firmware file:
      "/etc/sound/pndsperm.bin"
<M>   VIA 82C686 Audio Codec
[*]   VIA 82C686 MIDI
<M>   OSS sound modules
[ ]       Verbose initialisation
[*]       Persistent DMA buffers
<M>     AD1816(A) based cards (EXPERIMENTAL)
<M>     Aztech Sound Galaxy (non-PnP) cards
<M>     Adlib Cards
<M>     ACI mixer (miroSOUND PCM1-pro/PCM12/PCM20)
<M>     Crystal CS4232 based (PnP) cards
<M>     Ensoniq SoundScape support
<M>     Gravis Ultrasound support
[*]       16 bit sampling option of GUS (_NOT_ GUS MAX)
[*]       GUS MAX support
<M>     Loopback MIDI device support
<M>     MediaTrix AudioTrix Pro support
<M>     Microsoft Sound System support
<M>     MPU-401 support (NOT for SB16)
<M>     NM256AV/NM256ZX audio support
<M>     OPTi MAD16 and/or Mozart based cards
[*]       Support MIDI in older MAD16 based cards (requires SB)
```

```
    <M>     ProAudioSpectrum 16 support
    <M>     PSS (AD1848, ADSP-2115, ESC614) support
    [ ]       Enable PSS mixer (Beethoven ADSP-16 and other compatibile)
    [ ]       Have DSPxxx.LD firmware file
    <M> 100% Sound Blaster compatibles (SB16/32/64, ESS, Jazz16) support
    <M>     AWE32 synth
    <M> Full support for Turtle Beach WaveFront (Tropez Plus, Tropez,
        Maui) synth/soundcards
    <M>     Limited support for Turtle Beach Wave Front (Maui, Tropez)
            synthesizers
    <M>     Yamaha FM synthesizer (YM3812/OPL-3) support
    <M>     Yamaha OPL3-SA1 audio controller
    <M>     Yamaha OPL3-SA2 and SA3 based PnP cards
    <M>     Yamaha YMF7xx PCI audio (native mode)
    [*]       Yamaha PCI legacy ports support
    <M>     6850 UART support
    <M>     Gallant Audio Cards (SC-6000 and SC-6600 based)
    [*]       SC-6600 based audio cards (new Audio Excel DSP 16)
    [*]         Activate SC-6600 Joystick Interface
    (4)   SC-6600 CDROM Interface (4=None, 3=IDE, 1=Panasonic, 0=?Sony?)
    (0)         SC-6600 CDROM Interface I/O Address
    [*]       Audio Excel DSP 16 (SBPro emulation)
    [*]       Audio Excel DSP 16 (MPU401 emulation)
    <M>   TV card (bt848) mixer support
USB support  →
    <M> Support for USB
    [ ]   USB verbose debug messages
    --- Miscellaneous USB options
    [*]   Preliminary USB device filesystem
    [ ]   Enforce USB bandwidth allocation (EXPERIMENTAL)
    [*]   Long timeout for slow-responding devices (some MGE Ellipse
          UPSes)
    [ ]   Large report fetching for "broken" devices (some APC UPSes)
    --- USB Controllers' 'CONFIG_USB_UHCI
    <M>   UHCI (Intel PIIX4, VIA, ...) support
    <M>   UHCI Alternate Driver (JE) support
    <M>   OHCI (Compaq, iMacs, OPTi, SiS, ALi, ...) support
    --- USB Device Class drivers
    <M>   USB Audio support
    <M>   USB Bluetooth support (EXPERIMENTAL)
    <M>   USB Mass Storage support
```

```
[ ]     USB Mass Storage verbose debug
[*]     Freecom USB/ATAPI Bridge support
[*]     Microtech CompactFlash/SmartMedia reader
[*]     SanDisk SDDR-09 SmartMedia reader support
[*]     Hewlett-Packard 8200e/8210e CD-Writer Plus support
<M>   USB Modem (CDC ACM) support
<M>   USB Printer support
--- USB Human Interface Devices (HID)
<M>   USB Human Interface Device (full HID) support
[*]     /dev/hiddev raw HID device support (EXPERIMENTAL)
<M>   USB HIDBP Keyboard (basic) support
< >   USB HIDBP Mouse (basic) support
<M>   Wacom Intuos/Graphire tablet support
--- USB Imaging devices
<M>   USB Kodak DC-2xx Camera support
<M>   USB Mustek MDC800 Digital Camera support (EXPERIMENTAL)
<M>   USB Scanner support
<M>   Microtek X6USB scanner support
<M>   HP 5300 C scanner support (EXPERIMENTAL)
--- USB Multimedia devices
<M>   USB IBM (Xirlink) C-it Camera support
<M>   USB OV511 Camera support
<M>   USB Philips Cameras
<M>   USB SE401 Camera support
<M>   D-Link USB FM radio support (EXPERIMENTAL)
<M>   DABUSB driver
--- USB Network adaptors
<M>   PLUSB Prolific USB-Network driver (EXPERIMENTAL)
<M>   USB ADMtek Pegasus-based ethernet device support
      (EXPERIMENTAL)
<M>   USB KLSI KL5USB101-based ethernet device support
      (EXPERIMENTAL)
<M>   USB CATC NetMate-based Ethernet device support (EXPERIMENTAL)
<M>   USB Communication Class Ethernet driver (EXPERIMENTAL)
<M>   USB-to-USB Networking cable device support (EXPERIMENTAL)
--- USB port drivers
<M>   USS720 parport driver
USB Serial Converter support  →
     <M> USB Serial Converter support
     [ ]   USB Serial Converter verbose debug
```

```
          [*]   USB Generic Serial Driver
          <M>   USB Belkin and Peracom Single Port Serial Driver
                (EXPERIMENTAL)
          <M>   USB ConnectTech WhiteHEAT Serial Driver (EXPERIMENTAL)
          <M>   USB Digi International AccelePort USB Serial Driver
          <M>   USB Empeg empeg-car Mark I/II Driver (EXPERIMENTAL)
          <M>   USB FTDI Single Port Serial Driver (EXPERIMENTAL)
          <M>   USB Handspring Visor Driver
          <M>   USB Inside Out Edgeport Serial Driver (EXPERIMENTAL)
          <M>   USB Keyspan PDA Single Port Serial Driver (EXPERIMENTAL)
          <M>   USB Keyspan USA-xxx Serial Driver (EXPERIMENTAL)
          [ ]     USB Keyspan USA-28 Firmware
          [ ]     USB Keyspan USA-28X Firmware
          [ ]     USB Keyspan USA-19 Firmware
          [ ]     USB Keyspan USA-18X Firmware
          [ ]     USB Keyspan USA-19W Firmware
          [ ]     USB Keyspan USA-49W Firmware
          <M>   USB MCT Single Port Serial Driver (EXPERIMENTAL)
          <M>   USB Prolific 2303 Single Port Serial Driver
                (EXPERIMENTAL)
          <M>   USB REINER SCT cyberJack pinpad/e-com chipcard reader
                     (EXPERIMENTAL)
          <M>   USB ZyXEL omni.net LCD Plus Driver (EXPERIMENTAL)
          --- Miscellaneous USB drivers
          <M>   USB Diamond Rio500 support (EXPERIMENTAL)
Bluetooth support  →
     <M> Bluetooth subsystem support
     <M> L2CAP protocol support
     Bluetooth device drivers  →
          <M> HCI USB driver
          <M> HCI UART driver
          <M> HCI EMU (virtual device) driver
Kernel hacking  →
     [ ] Prefer small over fast code
     [*] Kernel debugging
     [ ]   Debug memory allocations
     [ ]   Memory mapped I/O debugging
     [*]   Magic SysRq key
     [ ]   Spinlock debugging
     [*]   Verbose BUG() reporting (adds 70K)
```

Приложение 5

Дополнительная литература

Здесь приведен список литературы, использованной при написании книги. К сожалению, список невелик, в том числе из-за недостатка книг, посвященных операционной системе Linux, качественно переведенных на русский язык. Тем не менее, практически все перечисленные книги заслуживают внимания. Некоторые из них интересны глубоким освещением теоретической части, некоторые — более привлекательны для практиков. Бóльшая часть книг, как нам представляется, предназначена для специалистов среднего и высокого уровня.

- Немет Э., Снайдер Г., Сибасс С., Трент Р. Хейн. UNIX: руководство системного администратора: Пер. с англ. — 2-е изд. — К.: BHV, 1997.

  Уникальная книга. По охвату вопросов, раскрытию тем аналогичной книги, пожалуй, не найти. Написана специалистами-практиками и для практиков. Ориентирована на опытного пользователя. Первое и второе издания книги посвящены коммерческим реализациям UNIX и, на первый взгляд, имеют отдаленное отношение к операционной системе Linux. Однако знание UNIX — это 98% успеха в освоении Linux.

- Немет Э., Снайдер Г., Сибасс С., Хейн Т. Р. UNIX: руководство системного администратора. Для профессионалов: Пер. с англ. — 3-е изд., перераб. и доп.— СПб.: Питер; К.: Издательская группа BHV, 2002.

  Третье издание книги. Она существенно переработана. Уменьшено количество рассматриваемых операционных систем и, что очень важно, среди рассматриваемых операционных систем появилась Linux. Эта книга должна быть на столе у каждого администратора.

- Карлинг М., Деглер С., Деннис Дж. Системное администрирование Linux / Учебное пособие: Пер. с англ. — М.: Издательский дом "Вильямс", 2000.

  Хорошая книга по администрированию системы. Некоторые вопросы затронуты неглубоко, однако направление поиска задают верно. Очень рекомендуется для системных администраторов и специалистов IT.

- Такет Дж. (мл.), Гантер Д. Использование Linux : Пер. с англ. — 3-е изд. — К.; СПб.; М.: Издательский дом "Вильямс", 1998.

  Хорошая книга для начинающих пользователей. Переиздавалась несколько раз. В освоении операционной системы Linux на первых порах очень помогает.

- Зиглер Р. Брандмауэры в Linux / Учебное пособие: Пер. с англ. — М.: Издательский дом "Вильямс", 2000.

  Книга полностью посвящена построению защищенной сети. Рекомендуется для системных администраторов.

- Максвел С. Ядро Linux в комментариях: Пер. с англ. — К.: ДиаСофт, 2000.

  Содержание этой книги понятно из названия. С выходом ядер версии 2.4 информация несколько устарела, однако книга весьма полезна для ознакомления с общей идеологией ядра операционной системы Linux.

- Робачевский А. М. Операционная система UNIX. — СПб.: БХВ — Санкт-Петербург, 1999.

  Хорошая теоретическая книга. В ней объясняются принципы функционирования операционной системы, основные понятия, протокол TCP/IP и многое другое.

- Рейчард К., Фолькердинг П. Linux: справочник. — СПб.: Питер Ком, 1999.

  Хороший справочник по программам, утилитам и командам операционной системы Linux. Некоторые утилиты устарели, некоторые важные утилиты не описаны, но в целом эта книга — хорошее приобретение на книжную полку и пользователю, и администратору.

- Шевель А. Linux. Обработка текстов. Специальный справочник. — СПб.: Питер, 2001.

  Книга посвящена текстовым редакторам и системе CVS-управления и контроля версий текстов (исходных кодов программного обеспечения).

- Блам Р. Система электронной почты на основе Linux / Учебное пособие: Пер. с англ. — М.: Издательский дом "Вильямс", 2001.

  В книге рассматриваются настройка почтового сервера и конфигурирование почтовых клиентов. Хорошая книга для освоения начал работы с почтовыми сообщениями и построения простых почтовых серверов.

Приложение 6

Ссылки

Здесь собраны Web-ресурсы, тем или иным образом связанные с операционной системой Linux. Приведенный список далеко не полон, вряд ли он охватывает даже малую часть таких ресурсов. Создателей Web-ресурсов просим не обижаться — всех удовлетворить невозможно, да и Интернет необозримо велик. Если кто-либо знает не указанный в списке полезный ресурс — сообщите нам.

А теперь перейдем к собственно ссылкам.

Дистрибутивы:

- **www.altlinux.ru** — сайт дистрибутива AltLinux. Популярный российский дистрибутив.
- **www.asplinux.ru** — сайт дистрибутива ASPLinux. Популярный российский дистрибутив.
- **www.bestlinux.net** — сайт дистрибутива Best Linux.
- **www.blackcatlinux.com** — сайт дистрибутива Black Cat. Украинский дистрибутив. Более развиваться не будет — разработчики дистрибутива вошли в состав команды российского дистрибутива ASPLinux.
- **www.caldera.com** — сайт дистрибутива Caldera. Пожалуй самый коммерциализованный дистрибутив. Рассчитан на корпоративный сектор.
- **www.corel.com** — сайт фирмы Corel, производителя одноименного дистрибутива и программного обеспечения для Linux. К сожалению, фирма Corel отказалась от поддержки своего дистрибутива и сворачивает выпуск программного обеспечения для Linux.
- **www.debian.org** — сайт дистрибутива Debian. Один из распространенных дистрибутивов Linux. В его состав входит огромный набор программных пакетов. Одно из несомненных достоинств — постоянная поддержка дистрибутива разработчиками. Несомненным плюсом для русскоговорящих пользователей является Web-сайт на русском языке.
- **www.ksi-linux.com** — сайт дистрибутива KSI. Дистрибутив создан в Киеве Сергеем Кубушиным. По всей видимости, больше развиваться не будет.

- **linuxiso.org** — специальный сайт, содержащий ISO-образы дистрибутивов.
- **www.linuxmandrake.com/ru** — русская версия дистрибутива Mandrake. Достаточно популярен.
- **www.linuxrouter.org** — сайт однодискетного дистрибутива Linux. В основном используется для организации маршрутизатора.
- **www.redhat.com** — сайт фирмы Red Hat, производителя одноименного дистрибутива. Одним из достоинств данного дистрибутива является его хорошая поддержка, начиная с версии 4.*x* (создан в 1995 году) и заканчивая текущим. Дистрибутив Red Hat получил очень широкое распространение, его уже начали сравнивать с Windows. В настоящее время Red Hat стал стандартом de-facto для производителей коммерческого программного обеспечения и компьютерного оборудования. Дистрибутив существует в трех вариантах: базовый (доступен для скачивания через Интернет), Professional и Advanced Server. Кроме собственно дистрибутива на сайте присутствует и достаточно большое количество качественно написанной документации (на английском языке).
- **www.roslinux.com** — сайт дистрибутива RosLinux. Российский дистрибутив.
- **www.slackware.com** — сайт дистрибутива Slackware. Один из старейших дистрибутивов. Считается, что он достаточно сложен в инсталляции и настройке. Однако его несомненное достоинство — действительно понятный и логичный набор пакетов, а так же возможность создать малую по размеру установленную операционную систему.
- **www.suse.de** — сайт немецкого дистрибутива SuSE. Несомненное достоинство — большой набор программных пакетов, входящих в состав дистрибутива.
- **www.turbolinux.com** — сайт дистрибутива Turbo Linux.

Документация:

- **dc.internic.net/rfc/rfc2196.txt** — документ, посвященный политике безопасности системы.
- **www.bog.pp.ru** — великолепный сайт Сергея Богомолова. Статьи по установке и настройке множества программ.
- **www.citforum.ru** — огромное собрание русскоязычной документации и книг, в том числе и посвященных Linux. Ресурс интересен разнообразием предоставляемой информации: программирование, базы данных, описание протоколов, обзоры программ и многое другое.
- **dc.internic.net/rfc** — стандарты RFC.
- **www.inp.nsk.su/~baldin** — Балдин Евгений. The Linux Cyrillic HOWTO (rus).
- **www.lib.ru** — знаменитая библиотека Мошкова.
- **linuxbegin.by.ru** — сайт для начинающих пользователей.

- **http://www.linuxdoc.org/** — сайт с огромным количеством документации на английском языке.
- **www.linuxdocs.org** — одно из полнейших собраний документации о Linux. Ресурс англоязычный, зато почти все, что касается Linux, здесь можно тем или иным образом найти.
- **www.linuxfocus.org** — электронный журнал "LinuxFocus", есть русский перевод.
- **www.linux.hitech.by** — в основном статьи X-Stranger.
- **www.linuxoid.ru** — еще один русскоязычный ресурс, посвященный Linux. На сайте находятся статьи, переводы документации и HOWTO.
- **www.linux.org.ru** — один из основных русскоязычных сайтов, посвященных Linux. На сайте собрана различная информация, тем или иным образом касающаяся Linux и программного обеспечения для этой операционной среды. Рекомендуется всем пользователям Linux.
- **LinuxPrinting.org** — сайт, посвященный печати в Linux.
- **www.linuxrsp.ru** — неплохой русскоязычный сайт.
- **linuxtv.org** — сайт, посвященный телевидению и Linux.
- **www.linux-ve.chat.ru** — виртуальная энциклопедия "Linux по-русски". Составитель В. А. Костромин. Содержит ссылки на разнообразные ресурсы русскоязычного Интернета, разбитые на тематические разделы. К сожалению, не все ссылки являются "живыми", однако это не умаляет достоинств ресурса, который мы очень рекомендуем и начинающим, и опытным пользователям.
- **www.Linux-USB.org** — сайт, посвященный USB-устройствам и их применимости с точки зрения Linux.
- **www.opennet.ru** — очень хороший сайт, посвященный операционным системам и сетям.
- **www.pathname.com/fhs/** — Filesystem Hierarchy Standard в различных текстовых форматах.
- **www.redhat.com** — в разделе "Документация" содержится документация на дистрибутив и некоторый объем дополнительной информации.
- **www.rfc-editor.org** — сайт, посвященный стандартам RFC.
- **www.rpm.org** — сайт, полностью посвященный RPM.
- **squid.org.ua** — зона особого внимания: Squid — сайт, полностью посвященный программе squid.
- **www.softerra.ru/freeos/** — раздел сайта, посвященный открытым операционным системам.
- **www.tldp.org** — Linux Documentation project. Англоязычный сайт, содержащий в структурированном виде документацию по Linux.

- **www.tsu.ru/~pascal/x_locale/** — сайт Ивана Паскаля: локаль и X Window.
- **www.unicode.org** — сайт, посвященный Unicode.
- **linux.webclub.ru** — хорошая подборка документации по Linux.

Программное обеспечение:

- **acl.bestbits.at** — официальная страница проекта Linux ACLs (Access Control Lists).
- **www.amanda.ocg** — сайт программы AMANDA.
- **www.apache.org** — официальный сервер Apache.
- **apache.lexa.ru** — сервер группы разработчиков русского модуля Apache.
- **www.applix.com** — сайт фирмы-разработчика Applixware.
- **www.borland.com** — официальный сайт фирмы Borland. Фирма Borland широко известна программистам и пользователям стран СНГ, можно даже сказать, что на программных продуктах этой фирмы выросло не одно поколение программистов и пользователей. Помимо чисто коммерческих проектов, фирма занимается выпуском и бесплатных продуктов под лицензией GNU. Для программистов интересен проект Kylix — кросс-платформенная среда быстрой разработки программ (Windows, Linux), использующая модификацию языка Object Pascal. Благодаря Kylix стало возможно переносить программы, написанные на Delphi, в Linux. В скором времени фирма Borland обещает выпустить кросс-платформенную реализацию C Builder. Помимо этого, существует кросс-проект Java Builder и SQL-сервер Interbase, который рекомендуется для разработчиков баз данных: легкий, существуют версии для Windows и Linux, бесплатный, способен работать с большими объемами баз данных и поддерживать на обычном компьютере 20—30 одновременных соединений.
- **www.compu-art.de/mars_nwe** — домашняя страница Mars_NWE.
- **www.eecis.udel.edu/~ntp** — страница, посвященная серверу точного времени XNTP.
- **www.false.com/security/linux/** — Secure Linux patches by Solar Designer. Дополнения к ядру Linux, повышающие безопасность операционной системы.
- **www.freshmeat.net** — сайт, содержащий огромное количество программ для Linux.
- **www.gnokii.org** — официальная страница проекта Gnokii.
- **www.gnome.org/gnome-office** — официальная страница офисного пакета GNOME Office.
- **www.gnome.org** — официальный сайт GNOME.
- **www.gnu.org/software/grub/** — домашняя страница программы-загрузчика GRUB.

- **www.idsoftware.com** — сайт компании-разработчика игр Doom, Quake, Quake II, Quake III. Пожалуй, одна из первых (если не первая) коммерческая фирма, выпустившая почти все свои игры для Linux, причем исходные коды нескольких игр фирма открыла для свободного доступа.
- **www.interbase.com** — официальный сайт Interbase. Подразделение фирмы Borland, разрабатывающее SQL-сервер Interbase.
- **www.isc.org/products/INN** — официальный сайт сервера новостей INN.
- **www.kde.org** — официальный сайт KDE.
- **www.kdevelop.org** — официальный сайт среды программирования kdevelop.
- **koffice.kde.org** — официальный сайт офисного пакета Koffice.
- **www.kernel.org** — сайт ядра операционной системы Linux.
- **www.lids.org** — сайт проекта LIDS.
- **www.linmodems.org** — сайт, посвященный Win-модемам и драйверам для них под Linux.
- **linux.freeware.ru** — программное обеспечение для Linux.
- **www.linuxgames.org.ru** — русскоязычный сайт, посвященный играм для Linux. Как вы сами можете убедиться, большим ассортиментом современных игр Linux похвастать не может.
- **www.lirc.org/** — страничка проекта Linux Infrared Remote Control, LIRC.
- **www.lokigames.com** — сайт фирмы, которая портирует Windows-игры для Linux. Помимо этого, фирма выпустила библиотеки для облегчения разработки кросс-платформенных игр и для портирования Windows-игр на платформу Linux. К сожалению, фирма обанкротилась, и Linux-сообщество рискует остаться без современных игр.
- **www.mostang.com/sane** — официальная страница пакета SANE.
- **www.mozilla.org** — официальный сайт Mozilla. Кросс-платформенный Web-браузер с открытым исходным кодом Mozilla основан на движке Gecko фирмы Netscape. Динамично развивающийся проект. На сегодняшний день код практически очищен от ошибок. На основе кода Mozilla разрабатываются несколько альтернативных Web-браузеров.
- **www.mrtg.org** — официальный сайт пакета MRTG.
- **www.mysql.org** — официальный сайт SQL-сервера MySQL. Основными задачами разработчики поставили быстродействие, простоту реализации и нетребовательность к ресурсам. К сожалению, в этом сервере не реализовано много возможностей языка SQL, в частности, хранимые процедуры. Основной нишей MySQL являются простые проекты баз данных и хранилище информации для Web-сайтов.
- **www.netraverse.com** — сайт производителя Win4Lin.

- **www.opendivx.org** — сайт кодека DivX с открытым исходным кодом.
- **www.openoffice.org** — официальный сайт офисного пакета Open Office.
- **www.openssh.com** — сайт некоммерческой реализации SSH.
- **www.opera.com** — сайт фирмы-разработчика Web-браузера Opera. Очень неплохая кросс-платформенная альтернатива Web-браузеру Mozilla: легкий и удобный. Единственное "но" — это коммерческий продукт с закрытым исходным кодом.
- **www.psionic.com** — сайт Psionic Software — разработчика программы Portsentry.
- **rpmfind.net** — репозитарий и поисковая система RPM.
- **rrdtool.eu.org** — официальный сайт пакета rrdtool.
- **rufus.w3.org/linux/RPM** — репозитарий RPM.
- **www.samba.org** — официальный сайт проекта Samba.
- **www.slug.org.au/etherboot/** — страница пакета Etherboot, предназначенного для загрузки бездисковых станций.
- **www.squid-cache.org** — официальный сайт программы Squid.
- **stunnel.mirt.net** — официальный сайт пакета Stunnel.
- **www.tripwire.org** — сайт разработчиков Tripwire.
- **www.vmware.org** — официальный сайт проекта VMWare.
- **www.webmin.com** — официальный сайт проекта Webmin.
- **www.winehq.org** — официальный сайт проекта Wine.
- **www.ximian.com** — сайт активного участника разработки GNOME, а также почтового клиента Evolution — фирмы Ximian.
- **xmms.org** — сайт программы XMMS — воспроизведение аудио и видео.
- **www.xsane.org** — официальный сайт Xsane.

Безопасность:

- **linuxsecurity.com** — сайт, посвященный безопасности операционной системы Linux.
- **www.security.nnov.ru** — сайт, посвященный компьютерной безопасности.
- **www.rootshell.com** — сайт, посвященный компьютерной безопасности (англоязычный).

Предметный указатель

C

D

E

F

G

I

J

K

L

T

U

V

W

X