Администрирование Linux. Часть6.

Пособие для дистанционных курсов.
http://linuxnavigator.ru.

21.07.2009

ОГЛАВЛЕНИЕ

НАСТРОЙКА DNS СЕРВЕРА.

С увеличением количества компьютеров в сети стало неудобно обращаться к ним по IP адресу, поэтому компьютерам начали присваивать имена. Первоначально для преобразования имени компьютера в IP адрес и обратно использовался файл */etc/hosts*. Одним из условий правильного преобразования имен при помощи этого файла является наличие его на всех компьютерах в сети, причем с одинаковым содержанием.

Когда в сети несколько сотен компьютеров, синхронизация файла *hosts* между ними не вызывает особых проблем. Но если представить себе то количество машин, которое сейчас присутствует в Интернет, то сразу становится понятно, что использование файла *hosts* вызовет очень большие затруднения. Представьте себе, что при подключении новой машины в сеть необходимо обновить этот файл на всех машинах сети. Или вопрос поиска информации в этом огромном файле.

В начале 80-х годов была создана система DNS — Domain Name System. Эта система представляет из себя распределенную базу данных, предназначенную для преобразования имен компьютеров в IP адреса и наоборот. DNS лишена недостатков файла *hosts* и позволяет довольно быстро осуществлять преобразование имен.

ПРОСТРАНСТВО ИМЁН DNS.

С появлением системы DNS в имени машины были добавлены домены.

Сразу хочу предупредить, что понятие домена — это весьма заезженное *понятие. Чего только не называют доменом. Домены системы DNS применяются только для преобразования имён. Не путайте их с доменами Windows — это абсолютно другие домены, предназначенные для других целей.*

Доменная структура DNS является иерархической. Иерархия имен начинается с корневого домена, обозначаемого символом точка. В корневом домене находятся домены первого уровня. Они могут быть организованы по географическому принципу: ru, ua, fi и т.д. И по историческому принципу, домены, которые изначально использовались в США: com, net, mil и т.д. Дальше идут домены второго, третьего и т.д. уровней.

Когда мы пользуемся DNS, мы используем понятие FQDN — полностью квалифицированное доменное имя машины. Если раньше имя машины могло быть простым, например: petia, vasia. То теперь имя машины *всегда* состоит из двух частей: имя машины и имя домена.

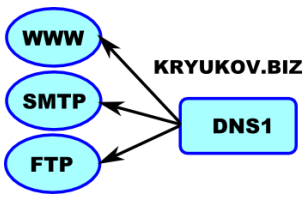
Например:

www.kryukov.biz — имя машины *www*, имя домена *kryukov.biz*.
c1.st1.kryukov.biz — имя машины *c1*, имя домена *st1.kryukov.biz*.

DNS умеет работать только с FQDN именами, она не понимает краткие имена машин. Подумайте, разве в браузере вы набираете имя машины www и нажимаете Enter? Или, всё таки пишите имя машины и имя домена?

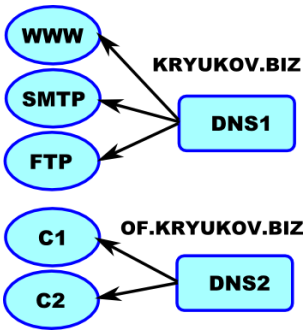
ЗОНА ОТВЕТСТВЕННОСТИ DNS СЕРВЕРА.

DNS — это распределённая база данных. Куски этой базы хранятся на DNS серверах. Ни один DNS сервер не хранит всю базу Интернет.



Каждый DNS сервер имеет свою зону ответственности. Он не отвечает за весь Интернет, а только за небольшую часть его. Зона ответственности — это компьютеры в домене, имена которых DNS сервер преобразует в IP адреса.

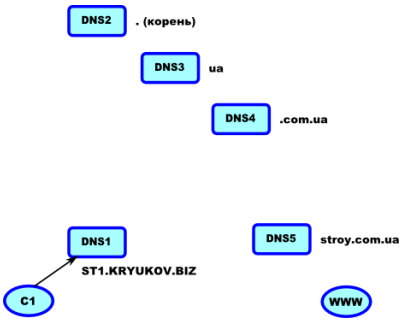
Например, DNS сервер, отвечающий за домен *kryukov.biz* предназначен для преобразования имен компьютеров, входящих в этот домен. Если посмотреть на схему, то становиться понятно, что этот DNS сервер может преобразовать имена компьютеров: *www.kryukov.biz*, *smtp.kryukov.biz* и *ftp.kryukov.biz*.



Предположим, что есть другой домен *of.kryukov.biz*. В нём есть две машины: *c1* и *c2*. За этот домен отвечает другой DNS сервер (DNS2). Если спросить у DNS1 IP адрес машины *c1* он его не даст, потому, что это не его зона ответственности. Он попросит вас обратиться к ответственному за зону DNS серверу. Точно так же DNS2 не отвечает за машины в домене *kryukov.biz* и не будет давать вам их IP адреса.

Современные программы DNS сервера могут отвечать за несколько доменов. Делать пересылки запросов по условию. Но об этом мы поговорим позже.

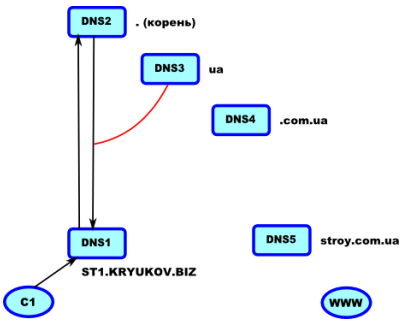
ПРИНЦИП РАБОТЫ DNS.



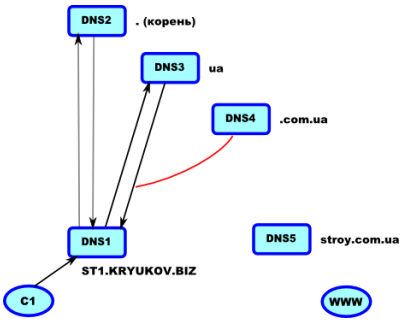
Предположим, что мы работаем на машине *c1.st1.kryukov.biz*. Клиент системы DNS на этой машине настроен таким образом, что все запросы он посылает на DNS сервер номер 1. Так же предположим, что все машины были только что включены и никаких запросов к DNS не было. На нашей машине мы запустили браузер и пытаемся подключиться к машине *www.stroy.com.ua*. Что при этом произойдёт?

Для того, что бы подключиться к WEB серверу *www.stroy.com.ua*, браузер должен знать IP адрес этой машины. Поскольку мы написали имя машины, он обращается к клиенту DNS, который в свою очередь посылает запрос на *DNS1*.

Сервер *DNS1* не отвечает за домен *story.com.ua*, поэтому не может сказать нам адрес машины *www* в этом домене. Но он настроен таким образом, что все вопросы по доменам за которые он не отвечает, он пересылает на DNS сервер, обслуживающий корневой домен (точка).

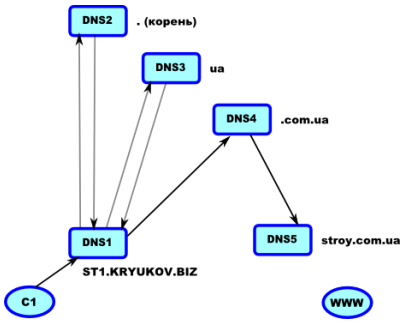


Запрос пришёл на корневой DNS сервер *DNS2*. Этот сервер тоже не отвечает за домен *story.com.ua* и он ничего не знает о машине *www* в этом домене. Но он говорит нам имя и IP адрес DNS сервера. Отвечающего за домен *ua*, поскольку последний входит в зону его ответственности.

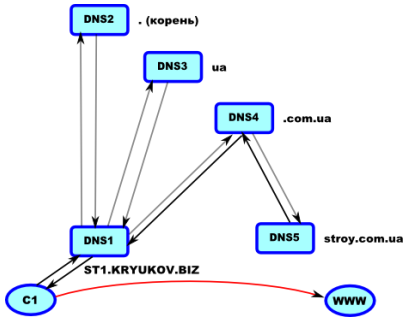


Теперь нам известен IP адрес DNS сервера, отвечающего за домен ua (*DNS3*). И мы обращаемся к нему с таким же вопросом: дайте IP машины *www.stroy.com.ua*. Этот сервер тоже не отвечает за домен и не знает IP машины *www*. Но он сообщает нам IP адрес DNS сервера, отвечающего за домен *com.ua*.

Мы обращаемся к *DNS4*. Несмотря на то, что этот сервер тоже не отвечает за домен *story.com.ua*, он знает IP адрес DNS сервера *DNS5* и сам обращается к нему с вопросом о машине *www*. Почему он сам пошел за адресом, а не переправил нас к *DNS5*? Пока скажем просто — он так настроен. Более подробно мы это рассмотрим несколько позже.



DNS5 отвечает за домен *story.com.ua* и знает IP адрес машины *www*. Он даёт IP DNS серверу *DNS4*, тот передаёт его нашему DNS серверу. Который в свою очередь, передаёт его клиенту. Клиент, получив IP, подключается к WEB серверу.



Важно запомнить, что задача системы DNS — это преобразование имён. Доставка пакетов от клиентской машины к WEB серверу — это задача роутинга. Если система DNS выдала IP адрес машины, это не значит, что в дальнейшем пакеты могут быть доставлены на указанный IP.

Так же следует помнить принцип, по которому происходит общение между серверами. Обычно DNS сервер сконфигурирован таким образом, что если к нему приходит запрос про домен, за который он не отвечает, первый запрос сервера всегда идёт на корневой DNS сервер. А дальше мы спускаемся по иерархии вниз до тех пор, пока не дойдём до DNS сервера, отвечающего за интересующий нас домен.

КЕШИРУЮЩИЕ DNS СЕРВЕРА.

По статистике на одной WEB страничке в среднем находится около 15-ти картинок. И если предположить, что мы используем старый WEB браузер, не поддерживающий протокол HTTP версии 1.1, то на каждую картинку будет формироваться отдельный запрос. Произойдет около 15-ти запросов к нашему DNS серверу, и вся процедура определения IP адреса будет повторяться при каждом запросе. Если бы все было именно так, как описано выше, то подавляющая часть Интернет трафика состояла бы из DNS запросов.

Для того, чтобы не возникало проблем с повторяющимися запросами, все DNS сервера являются кэширующими. Наш сервер поместит информацию о машине *www.stroy.com.ua* в свой локальный кэш и при следующем запросе выдаст информацию из локального кэш. Кроме того он поместит в кэш ответы DNS серверов *DNS2*, *DNS3* и *DNS4*.

Время хранения информации в кэш сервера зависит от настроек зоны. В большинстве случаев оно составляет около суток.

РЕКУРСИВНЫЕ И НЕ РЕКУРСИВНЫЕ DNS СЕРВЕРА.

Если посмотреть на приведенную схему, то можно обратить внимание на различие в поведении DNS серверов 2, 3 и 4. Первые два сервера выдавали информацию только о нижестоящих в иерархии DNS серверах. А *DNS4* сам обратился за информацией к серверу *DNS5*.

Первые два сервера являются не рекурсивными, а сервер *DNS4* рекурсивным. Рекурсивные сервера пытаются самостоятельно выполнить все шаги по получению интересующей информации, не рекурсивные выдают только информацию о зоне, за которую они отвечают или информацию, находящуюся в их кэш.

DNS сервера, отвечающие за домены первого и второго уровня, обычно являются не рекурсивными, поскольку им приходиться обрабатывать большое количество запросов. Например, DNS сервера, отвечающие за корневой домен, обрабатывают около 10000 запросов в секунду.

DNS сервер, обозначенный на схеме под номером 4, тоже можно сделать не рекурсивным. Будет ли ваш DNS сервер рекурсивным или не рекурсивным зависит от того, как вы его настроите.

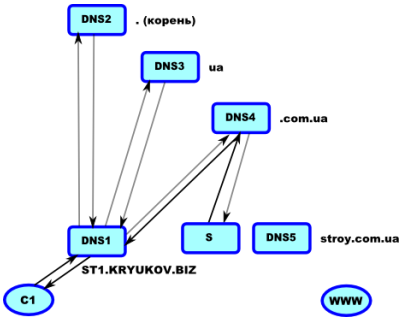
DNS сервера, обслуживающие запросы клиентов DNS, обязательно должны быть рекурсивными для клиентов. Клиенты DNS не понимают пересылок к другим DNS серверам.

DNS сервер, который будет рассматриваться на нашем курсе, позволяет сделать так, что для запросов внутренних клиентов DNS он будет выглядеть как рекурсивный сервер, а для внешних запросов как не рекурсивный.

АВТОРИТЕТНЫЕ И НЕ АВТОРИТЕТНЫЕ ОТВЕТЫ.

Предположим, что DNS сервер, отвечающий за домен *stroy.com.ua*, по каким-либо причинам не доступен. Тогда DNS сервер 4 не сможет получить информацию о машине *www.stroy.com.ua* и по истечении некоторого времени, отведенного на запрос, вернет нам отрицательный ответ — машина *www.stroy.com.ua* не существует. Эта информация попадет в кэш нашего сервера и будет выдана клиенту.

Теперь предположим, что через некоторый промежуток времени, DNS сервер, отвечающий за домен *stroy.com.ua*, станет доступен и начнет отвечать на вопросы. Но поскольку информация уже храниться в кэш нашего сервера, в среднем в течении суток наш сервер будет выдавать отрицательные ответы клиентам.



Чтобы такой ситуации не возникало, рекомендуется использовать дополнительные DNS сервера, поддерживающие ваш домен. На схеме такой сервер обозначен символом S. Серверу 4 известны все DNS сервера, отвечающие за домен *stroy.com.ua* и, если сервер 5 не отвечает на запросы, сервер 4 обратиться к другим DNS серверам, ответственным за домен.

Сервер 5 называется главным (*master*) DNS сервером. Дополнительные сервера называются подчиненными (*slave*) DNS серверами. Все изменения в домене описываются только на *master* сервере, на *slave* серверах информация непосредственно не редактируется, они получают её с *master* сервера.

Когда клиент DNS получает информацию от авторитетных серверов — *master* или *slave*, этот ответ называется авторитетным. Если информация получена из кэш, такой ответ называется не авторитетным.

DNS СЕРВЕР BIND.

Сервер BIND (Berkeley Internet Name Domain) — наиболее популярный в UNIX DNS сервер. Он распространяется организацией ISC (Internet Software Consorortium) в исходных кодах. Поставляется со всеми дистрибутивами Linux, в том числе и с CentOS.

BIND был создан в 1985 году Кевином Данлапом (Kevin Dunlap). На данный момент его последняя версия 9. Он поддерживает все основные особенности системы DNS, а также все новинки, которые были добавлены в последнее время, такие как:

- поддержка Ipv6.
- DNSSEC.
- TSIG.
- поддержка расширенного протокола DNS — EDNS0.

Особо хочется отметить поддержку EDNS0 сервером BIND. В расширенном протоколе добавлена возможность использования протокола TCP не только для передачи зон, но и для передачи запросов, в том числе и с цифровыми подписями. Кроме того, поддерживается инкрементальная пересылка зон и динамическое обновление зон (для DHCP серверов).

Bind в дистрибутиве CentOS поставляется в нескольких пакетах. Поэтому в первую очередь необходимо, что бы все, перечисленные пакеты были установлены. А именно: *bind*, *bind-utils*, *bind-libs*, *bind-chroot* и *caching-nameserver*. Если какой либо из пакетов не был установлен, установите его сейчас.

```
# yum install имя_пакета
```

КОНФИГУРАЦИОННЫЕ ФАЙЛЫ СЕРВЕРА BIND.

Основным конфигурационным файлом сервера BIND является файл */etc/named.conf*. Но в CentOS DNS сервер работает в chroot окружении, поэтому конфигурационный файл следует искать в директории */var/named/chroot/etc*.

```
# cd /var/named/chroot/etc/
```

Смотрим содержимое директории и содержимое файла *named.conf*. Если в директории присутствует файл *named.caching-nameserver.conf*, а *named.conf* пустой или отсутствует, тогда копируем шаблон в конфигурационный файл:

```
# cp named.caching-nameserver.conf named.conf
```

В файле *named.conf* определяются основные параметры сервера BIND, а так же зоны, которые он поддерживает.

В нём можно использовать комментарии в стиле C/C++ и shell script:

```
/* Это комментарий, который можно
располагать на нескольких строках */
// Комментарий на одной строке.
# Комментарий на одной строке.
```

Файл состоит из инструкций. Каждая инструкция начинается с ключевого слова, определяющего ее тип и должна завершаться символом *точка с запятой*. Если параметры инструкции не помещаются на одну строку, их необходимо брать в фигурные скобки: *{* и *}*.

Список поддерживаемых инструкций приведен в таблице:

Инструкция	Описание
include	Подключает внешний файл.
options	Определяет глобальные параметры сервера BIND.
server	Задает параметры сервера.
lwres	Конфигурирует сервер BIND 9 в качестве упрощенного распознавателя.
key	Определяет параметры аутентификации.
acl	Определяет списки управления.
zone	Определяет зоны, поддерживаемые сервером.
trusted-keys	Определенные в конфигурационном файле ключи шифрования.
controls	Определяет, как утилита rndc будет управлять сервером BIND.
logging	Определяет категории журнальных сообщений и каналы их распространения.
view	Определяет представление пространства имен.

НАПИСАНИЕ IP АДРЕСОВ.

В инструкциях достаточно часто необходимо описывать IP адреса машин и сетей. Можно писать:

IP адреса. Например, 178.11.201.64.
Адрес сети с маской подсети. Например, 193.16.18/24.
Имя ранее определенного списка контроля доступа (acl).
Оператор отрицания — !.

После каждого IP адреса необходимо ставить символ *точка с запятой*.

ИНСТРУКЦИЯ OPTIONS.

При помощи инструкции *options* задаются глобальные параметры сервера BIND.

```
options {
        параметр;
        параметр;
};
```

Ниже приведены некоторые параметры, определяемые в инструкции *options*:

Параметр	Описание
directory	Определяет текущую директорию сервера. В ней BIND будет искать файлы описания зон, и создавать различные дополнительные

	файлы. Обычно этот параметр ссылается на директорию */var/named*. Пример: `directory "/var/named"`
notify	Если параметр равен *yes*, то при изменении описания зоны будут посланы уведомления всем slave DNS серверам. Значение по умолчанию — *yes*. Пример: `notify no`
also-notify	При помощи этого параметра определяются DNS сервера, которые необходимо уведомить при изменении зоны. Slave DNS сервера в этом списке указывать не надо. Значение по умолчанию не определено. Пример: `also-notify {193.12.20.1; 193.12.38.200;};`
recursion	Параметр определяет, будет ли сервер BIND рекурсивным сервером. Значение по умолчанию — *yes*. Пример: `recursion yes`
allow-recursion	Этот параметр определяет адреса машин или сетей, для которых сервер BIND будет выступать в роли рекурсивного сервера. Значение по умолчанию не определенно. Пример: `allow-recursion {193.12.13.240; 194.12.34/24;};`

ИНСТРУКЦИЯ ACL.

Достаточно часто в конфигурационном файле сервера вам придётся определять списки IP адресов. Если одни и те же группы IP адресов встречаются часто, имеет смысл описать их при помощи параметра *acl*. Он позволяет присвоить имя такой группе.

```
acl имя {
        элементы списка;
};
```

Определение acl должно происходить до того, как его имя будет использоваться в параметрах. Например, мы определили acl с именем *internal*:

```
acl internal { 1.2.3.4; 2.3.4.5;
                    10.10.100/24; 192.168.0/24; };
```

Теперь, если где либо в конфигурационном файле *named.conf* нам потребуется указать эту группу IP адресов, мы можем просто написать имя acl — *internal*. Например, вот так:

```
allow-query { 127.0.0.1; internal; };
```

Существуют четыре заранее определенных списка:

any — соответствует всем узлам.
localnets — соответствует всем узлам локальной сети.
localhost — соответствует своему компьютеру.
none — не соответствует ни одному узлу.

ИНСТРУКЦИЯ ZONE.

Инструкция предназначена для описания зон ответственности DNS сервера. Для каждого поддерживаемого сервером DNS домена необходимо писать отдельную инструкцию zone.

При определении инструкции *zone* необходимо определять параметр *type*, при помощи которого задаётся тип зоны. Дополнительные параметры зависят от типа зоны.

```
zone "имя домена" IN {
        type тип_зоны;
        параметры зоны;
};
```

В таблице перечислены типы зон, которые поддерживаются DNS сервером BIND.

Тип	Описание
master	Определяет master зону.
slave	Определяет slave зону.
hint	Определяет зону подсказку.
forward	Определяет зону переадресации.
stub	Не рекомендуется для дальнейшего применения.

Рассмотрим пример описания зоны типа master. Предположим, что нам необходимо сделать так, что бы наш DNS сервер начал отвечать за домен *st1.kryukov.biz*. Для этого в конфигурационном файле *named.conf* необходимо добавить следующие строки:

```
zone "st1.kryukov.biz" {
        type master;
        file "master.st1.kryukov.biz";
        allow-update { none;};
        allow-query { any; };
        allow-tranfer { 1.2.3.4; 3.4.5.6;};
};
```

Непосредственно после инструкции zone, в двойных кавычках пишется имя домена. Затем открывается фигурная скобка и пишутся параметры зоны. Самым первым написан параметр *type*, указывающий, что это будет зона типа *master*.

Для зоны типа master обязательным является только параметр file, определяющий дополнительный файл, в котором будет описана данная зона. Он так и называется — файл описания зоны. Имя файл может быть любым, но в нём рекомендуется показать тип зоны и имя домена. В нашем случае по этому имени, *master.st1.kryukov.biz*, мы будем видеть, что это зона master для домена *st1.kryukov.biz*.

Остальные параметры отвечают за безопасность. Несмотря на то, что они не являются обязательными, настоятельно рекомендую вам их написать.

Параметр	Описание
allow-update	Определяет машины, которые могут удалённо менять информацию в файле описания зоны. Обычно тут пишут IP адреса DHCP серверов. Очень опасно оставлять этот параметр не определённым. Потому что по умолчанию удалённое изменение возможно с любой машины. Если у вас нет DHCP серверов, которые должны вносить изменения в зону, используйте acl *none*. Таким образом, вы явно запрещаете любое изменение.
allow-query	Определяет машины, с которых можно посылать запросы к DNS серверу.
allow-transfer	Определяет машины, которые могут требовать пересылку зоны. Тут следует писать IP адреса ваших slave серверов

ПРОВЕРКА СИНТАКСИСА.

Поскольку файл описания зоны — это текстовый файл. А вы не боги и можете допускать ошибки. После любого изменения в этом файле рекомендуется проверять его на наличие синтаксических ошибок. DNS сервер BIND очень нежная программа и может отказаться работать, если встретит ошибки.

Для проверки синтаксиса используется программа *named-checkconf*.

По умолчанию она проверяет файл */etc/named.conf*. Но если DNS сервер работает в chroot окружении, ей требуется явно указать путь к конфигурационному файлу, который она должна проверить.

```
# named-checkconf /var/named/chroot/etc/named.conf
```

Если в файле будут встречены ошибки, программа покажет вам номер строки, в которой обнаружена ошибка.

Наиболее часто встречаемая ошибка — это отсутствие *символа точка с запятой*.

ЛАБОРАТОРНАЯ РАБОТА. КОНФИГУРАЦИОННЫЙ ФАЙЛ NAMED.CONF.

Цель этой работы — создать конфигурационный файл *named.conf*, добавить новую зону ответственности и позаботиться о безопасности DNS сервера.

СОЗДАНИЕ КОНФИГУРАЦИОННОГО ФАЙЛА.

После установки DNS сервера необходимо создать конфигурационный файл *named.conf*. Поскольку сервер работает в chroot окружении (песочнице), файл будем создавать в директории */var/named/chroot/etc*.

```
# cd /var/named/chroot/etc/
```

В этой директории лежит файл шаблон *named.caching-nameserver.conf*, который мы используем для создания конфигурационного файла.

```
# cp named.caching-nameserver.conf named.conf
```

Открываем на редактирование *named.conf*. Содержимое файла по умолчанию такое:

```
options {
        listen-on port 53 { 127.0.0.1; };
        listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
```

```
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        query-source    port 53;
        query-source-v6 port 53;
        allow-query     { localhost; };
};
logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
view localhost_resolver {
        match-clients      { localhost; };
        match-destinations { localhost; };
        recursion yes;
        include "/etc/named.rfc1912.zones";
};
```

Как видно из листинга, DNS сервер слушает запросы только на *lo* интерфейсе (параметр *listen-on port 53 { 127.0.0.1; };)*. Непорядок, придется повозится.

Наша задача сделать так, что бы DNS сервер:

Был кеширующим.
Слушал запросы на всех сетевых интерфейсах.
Поддерживал наши зоны.
Позаботиться о безопасности.

После небольшой правки файл будет выглядеть так:

```
options {
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query     { localhost; };
};


logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
```

Параметр memstatistics-file*, пишем одной строкой.*

ДОБАВЛЕНИЕ ОПИСАНИЯ ЗОНЫ.

Имя домена и IP адреса серверов вы должны получить у преподавателя.

Теперь остается только добавлять в файле *named.conf* зоны ответственности нашего сервера. Обратите снимание, что в глобальной инструкции *options* мы разрешили принимать запросы только с нашей машины (*allow-query { localhost; };*). Это сделано по двум причинам:

> Безопасность. Существуют варианты атак на DNS сервера, от которых можно защититься только так. Мы в дальнейшем будем разрешать запросы к каждой зоне в отдельности.
> К нашему DNS серверу будут обращаться программы, работающие на нашей машине. Когда вы настроите клиентскую машину, мы добавим тут IP адрес этой машины или внутренней сети.

Нужно, что бы DNS отвечал за зону, которую вам делегировали, например, *st1.kryukov.biz*. Поэтому добавим в файл инструкцию *zone*:

```
zone "st1.kryukov.biz" IN {
        type master;
        file "master.st1.kryukov.biz";
        allow-update { none;};
        allow-query { any; };
        allow-tranfer { IP_SLAVE; };
};
```

Вместо st1.kryukov.biz вы напишете имя домена, которое вам дал преподаватель.

Тут мы запрещаем удалённое изменение содержимого (параметр *allow-update { none;};*). И разрешаем принимать запросы по поводу этой зоны с любых машин в Интернет (параметр *allow-query { any; };*). Пересылка зоны целиком разрешена только для наших slave серверов (параметр *allow-tranfer { IP_SLAVE; };*).

Вместо *IP_SLAVE* вы напишите IP адрес моего DNS сервера, который будет для вас slave сервером. Впрочем, если у вас есть возможность договориться с кем-то другим, что бы он поднял у себя slave зону, пишите IP их DNS сервера. Только не забудьте предупредить об этом меня, что бы я внёс соответствующие изменения в своей зоне.

ПРОВЕРКА СИНТАКСИСА. ДОПОЛНИТЕЛЬНЫЕ НАСТРОЙКИ.

Проверим синтаксис файла.

```
# named-checkconf /var/named/chroot/etc/named.conf
```

Передадим файл группе *named*, что бы DNS сервер мог его прочитать.

```
# chgrp named /var/named/chroot/etc/named.conf
```

ФАЙЛ ОПИСАНИЯ ЗОНЫ.

При создании инструкции *zone* мы указывали параметр file, в котором определяли имя файла описания зоны. Как я уже писал выше, DNS — это распределённая база данных, кусочки которой хранятся в DNS серверах. Файл описания зоны — это кусочек базы данных DNS.

Формат файла описания зоны не зависит от типа операционной системы и реализации программы сервера DNS. Он описан в RFC: 882, 1035, 1183, 2065 2181, 2308 и 2535.

СПЕЦИАЛЬНЫЕ СИМВОЛЫ.

В файле описания зоны можно использовать следующие специальные символы:

> Точка с запятой (;) — комментарий.

@ — Имя текущего домена. Берется либо из инструкции *zone*, либо определяется в файле описания зоны директивой *$ORIGIN*.
Круглые скобки () — разбивка данных на несколько строк.
Символ звёздочка (*) — используется только в именах машин или доменов.

ДИРЕКТИВЫ.

В файле описания зоны можно использовать директивы.

$TTL время — определяет время жизни записей в кэш DNS сервера для всех записей зоны.
$ORIGIN домен — определяет имя домена, подставляемое вместо символа @, или подставляется в конце не полностью определенного имени компьютера или домена.
$INCLUDE файл — подключает внешние файлы.
$GENERATE параметры — предназначен для создания наборов похожих записей.

ОСОБЕННОСТИ НАПИСАНИЯ FQDN ИМЕН В ФАЙЛАХ ОПИСАНИЯ ЗОНЫ.

В файлах описания зоны, *и только в файлах описания зоны*! При указании полностью квалифицированного доменного имени машины (FQDN) или полного имени домена. Требуется явно указывать имя корневого домена — символ точка.

Например, если необходимо указать FQDN имя машины *cosmos.kryukov.biz* в файле описания зоны, его необходимо описывать следующим образом:

```
cosmos.kryukov.biz.
```

Обратите внимание на точку в конце имени — это явное указание корневого домена.

Если в конце имени точки нет, то DNS сервер автоматически подставит имя текущего домена (имя берется либо из инструкции *zone*, конфигурационного файла *named.conf*, либо определяется при помощи директивы *$ORIGIN*). Например, если текущий домен *kryukov.biz*. И если в файле описания зоны его имя написано без точки, то в результате DNS сервер подставит следующее значение:

```
cosmos.kryukov.biz.kryukov.biz.
```

НАИБОЛЕЕ РАСПРОСТРАНЕННАЯ ОШИБКА ПРИ СОЗДАНИИ ФАЙЛА ОПИСАНИЯ ЗОНЫ — ЭТО ОТСУТСТВИЕ ТОЧКИ В КОНЦЕ FQDN.

ЗАПИСИ О РЕСУРСАХ.

Поскольку файл описания зоны — это кусочек базы данных, строки в этом файле называют записями. Строка имеет строго определённый формат:

```
[имя_компьютера|имя_домена] [ttl] [класс] тип параметры
```

Первое — это имя машины или домена. Что именно писать в этом поле, зависит от типа записи. Если это поле оставить пустым, то его значение берется из предыдущей записи.

ЕСЛИ ВЫ НЕ УКАЗЫВАЕТЕ ПЕРВОЕ ПОЛЕ, ТО В НАЧАЛЕ СТРОКИ ОБЯЗАТЕЛЬНО ДОЛЖЕН ПРИСУТСТВОВАТЬ ЛИБО СИМВОЛ ПРОБЕЛА, ЛИБО ТАБУЛЯЦИИ.

Второе поле — время жизни записи в кэш DNS сервера. Значение поля устанавливается в секундах. Если поле не определено, его значение берется из значения по умолчанию для данной зоны, определяемое при помощи инструкции *$TTL* в начале файла описания зоны.

Третье поле — класс сети. Можно использовать следующие классы сетей:

IN — Internet (значение по умолчанию).
CH — ChaosNet. В настоящее время не используется.
HS — Hesoid — информационная служба, являющаяся надстройкой пакета BIND. Используется крайне редко.

Тип записи — это зарезервированное слово. Основные типы записей приведены в таблице.

Тип	Описание
SOA	Определение параметров зоны DNS. Обязательная запись.
NS	Определение DNS серверов, авторитетных для зоны. Обязательная запись.
A	Преобразование имени в IP адрес.
AAAA	Преобразование имени в адрес IPv6.
A6	Преобразование имени в адрес IPv6.
PTR	Преобразование IP адреса в имя.
MX	Применяется для указания почтового сервера, отвечающего за почту домена.
KEY	Открытый ключ шифрования для DNS имени.
CNAME	Дополнительное имя машины (псевдоним).
SRV	Определение служб в пределах домена.
TXT	Дополнительные строки. Могут применять для работы различных программ.

Возможно применение некоторых других типов записей.

В ЛЮБОМ ФАЙЛЕ ОПИСАНИЯ ЗОНЫ ДОЛЖНЫ БЫТЬ ОПРЕДЕЛЕНЫ ДВЕ ОБЯЗАТЕЛЬНЫЕ ЗАПИСИ: SOA И NS.

ЗАПИСЬ SOA.

Запись SOA (Start Of Autority) — определяет начало описания зоны. Это одна из обязательных записей, которая должна присутствовать в файле описания зона. Она должна быть самой первой в файле описания зоны. Описание зоны в файле продолжается до тех пор, пока не встретиться другая запись SOA.

Пример записи SOA:

```
kryukov.biz. IN SOA cosmos.kryukov.biz. (
        artur.kryukov.biz. ; e-mail
        2008052000 ; серийный номер записи
        24H ; время обновления
        20M ; интервал между попытками
        2W ; интервал устаревания
        1D ) ; TTL
```

Рассмотрим из каких полей состоит запись SOA.

DNS не учитывает регистр букв. Вы можете писать записи любым регистром, но для удобства чтения информации я рекомендовал бы ключевые слова писать верхним регистром, а параметры нижним.

Первый параметр — это имя домена. В этом параметре можно использовать специальный символ @, вместо которого будет подставлено имя текущего домена.

Второй (необязательный) параметр *TTL* не указан, поэтому время жизни этой записи в кэш DNS сервера будет взято из значения по умолчанию для данной зоны.

Третий параметр — класс сети определен как *IN*. Несмотря на то, что это параметр не обязательный, его рекомендуют определять для улучшения читабельности файлов описания зоны.

Дальше идет тип записи *SOA*. Все остальные параметры — это параметры присущие только записи типа SOA.

Сначала указывается имя машины, на которой работает программа DNS сервер, являющийся master DNS для данного домена.

Второй параметр — почтовый адрес человека, отвечающего за данную зону. Поскольку символ @ имеет специальное значение, в email его не указывают. Данный параметр предназначен для чтения его человеком, а не машиной. Человек может легко понять как правильно писать email.

2008052000 — серийный номер записи зоны. Параметр необходим для нормальной работы slave серверов, отвечающих за зону. При изменении файла описания зоны вы *обязаны увеличить это число*, потому что оно используется slave DNS серверами для определения необходимости зонной пересылки. Если серийный номер на slave сервере больше или равен серийному номеру на master DNS сервере, зонной пересылки не будет. Если он меньше, то будет происходить зонная пересылка на slave сервер.

В качестве номера можно использовать: 1, 2, 3, 4 и т.д. Но в реальной практике рекомендуется в серийном номере указывать дату последнего изменения файла описания зоны. В приведенном примере, последнее изменение в зоне было в 2008 году, в Мае месяце, 20 числа. В этот день было одно изменение, о чем свидетельствует число 00.

В стандарте написано, что серийный номер — это 32-х битовое беззнаковое число. А это значит что максимальное число, которое можно написать в этом поле, равно 4294967295. До 4294 года далеко, поэтому разрядности числа на нашу жизнь хватит.

Дальше идет число, определяющее интервал времени, через который slave сервер будет обращаться к master серверу для сравнения серийных номеров записей. Согласно стандарта, это поле должно содержать количество секунд. Но DNS сервер BIND позволяет использовать сокращенные записи. В нашем случае *24H* — 24 часа. Т.е. раз в 24 часа, slave сервер будет обращаться к master DNS серверу с проверкой серийного номера записи. И если на мастере серийный номер больше, тогда slave скачает изменения в зоне к себе.

Если slave сервер не может подключиться к master серверу, он переходит в другой режим работы, при котором интервал времени обращения к master серверу уменьшается. Параметр *20M* определяет этот интервал. Если через 20 минут он не сможет подключиться, он повторяет попытку, потом ещё через *20M*. Если ему удалось подключиться к мастеру, он переходит в обычный режим, в нашем случае — раз в 24 часа.

В случае неудачного подключения slave сервер не может до бесконечности пытаться подключиться к master серверу, поэтому следующий параметр *2W* (две недели) определяет время, по прошествии которого slave сервер перестает поддерживать эту зону.

Последний параметр определяет время жизни по умолчанию в кэш DNS серверов отрицательных ответов нашего сервера. Например, наш сервер спрашивают: «скажи имя машины petia». А у нас в домене такой машины нет. Сервер естественно даёт ответ, но отрицательный. У каждой записи в нашей зоне есть специальное поле ttl, определяющее время жизни записи в кеш. А вот для отрицательных ответов сервера, сейчас используют последнее поле записи SOA.

ВСЕ ПАРАМЕТРЫ ЗАПИСИ ТИПА SOA ЯВЛЯЮТСЯ ОБЯЗАТЕЛЬНЫМИ.

ЗАПИСЬ NS.

Запись типа NS (Name Server) предназначена для описания всех DNS серверов, авторитетных для данного домена. При помощи этой записи вы должны описать все master и slave сервера.

NS ОТНОСИТЬСЯ К ОБЯЗАТЕЛЬНЫМ ЗАПИСЯМ. В ФАЙЛЕ ОПИСАНИЯ ЗОНЫ ДОЛЖНА БЫТЬ ОПРЕДЕЛЕНА КАК МИНИМУМ ОДНА ТАКАЯ ЗАПИСЬ.

Формат записи NS:

```
[зона] [TTL] [IN] NS имя_сервера
```

Например, за зону *kryukov.biz* отвечают три DNS сервера *cosmos.kryukov.biz*, *dns1.zenon.net* и *dns2.zenon.net*. Поэтому в файле описания этой будет три записи *NS*.

```
kryukov.biz.    IN      NS      cosmos.kryukov.biz.
kryukov.biz.    IN      NS      dns1.zenon.net.
kryukov.biz.    IN      NS      dns2.zenon.net.
```

Кто из перечисленных серверов является master, а кто slave определяется при помощи первого параметра записи *SOA*.

ЗАПИСИ MX.

Записи типа MX (Mail Exchanger) предназначены для указания почтовых серверов, отвечающих за прием почты для домена.

Формат записи MX:

```
[домен] [TTL] [IN] MX приоритет почтовый_сервер
```

При создании почтовой системы можно точно так же, как и в DNS, выделить основной и вспомогательный сервера. Приоритет отправки почты задается при помощи поля *приоритет*. Чем меньше число в этом поле, тем выше приоритет указанного сервера. То есть, по умолчанию почта будет отправляться на почтовый сервер с большим приоритетом. Но если он, по каким либо причинам не будет доступен, почта будет отправляться на сервер с меньшим приоритетом. Почтовые сервера должны быть сконфигурированы таким образом, чтобы они могли принимать почту для домена.

Предположим, что почту для домена kryukov.biz могут принимать два почтовых сервера: *cosmos.kryukov.biz* и *smtp.any.com*. Первый — основной почтовый сервер, второй — вспомагательный. Тогда в файле описания зоны необходимо добавить две записи:

```
@ IN MX  5 cosmos.kryukov.biz.
@ IN MX 10 smtp.any.com.
```

ЗАПИСЬ A.

Запись типа A (address) предназначена для преобразования имени машины в IP адрес.

Формат записи:

```
[имя_машины] [TTL] [IN] A IP_адрес
```

Предположим, что в домене *kryukov.biz* есть три машины: *master.kryukov.biz*, *c1.kryukov.biz* и *c2.kryukov.biz*. Тогда в файле описания зоны необходимо добавить три записи:

```
master IN A 192.168.1.1
c1     IN A 10.10.108.1
c2     IN A 172.17.88.4
```

ЗАПИСЬ CNAME.

Запись типа CNAME (Canonical Name) — позволяет добавить несколько имен одной и той же машине.

Формат записи CNAME:

```
дополнительное_имя [TTL] [IN] CNAME каноническое_имя
```

Одной машине можно присвоить несколько имен, в том числе и в разных доменах, которые преобразуются в один IP адрес. При помощи записи типа A рекомендуется определять только канонические имена. Дополнительные имена необходимо определять только при помощи записи типа CNAME.

Предположим, что машине *cosmos.kryukov.biz* необходимо присвоить дополнительные имена: *www.kryukov.biz*, *board.kryukov.biz* и *wiki.kryukov.biz*. Тогда в файл описания зоны будут добавлены следующие строки:

```
www   IN CNAME cosmos
board IN CNAME cosmos
wiki  IN CNAME cosmos
```

Важно запомнить, что параметр записи CNAME — это имя машины, а не ее IP адрес.

ЗАПИСЬ PTR.

Записи типа PTR (Pointer) предназначены для обратного преобразования — IP адреса в имя машины. Обычно эти записи применяются в зонах обратного преобразования.

Формат записи:

```
адрес [TTL] [IN] PTR имя
```

Более подробно этот тип записи будет рассмотрен в главе, посвящённой зонам обратного преобразования.

ПРОВЕРКА СИНТАКСИСА.

Для проверки синтаксиса файлов описания зоны с DNS сервером BIND поставляется специальная программа *named-checkzone*. В качестве аргумента командной строки программе необходимо указывать два параметра: имя домена и файл описания зоны.

Например, для проверки файла описания зоны домена *kryukov.biz*, программа запускается следующим образом:

```
# named-checkzone kryukov.biz \
/var/named/chroot/var/named/master.kryukov.biz
```

Если в файле будут ошибки, программа показывает номер строки, в которой была обнаружена ошибка.

ЛАБОРАТОРНАЯ РАБОТА. ФАЙЛ ОПИСАНИЯ ЗОНЫ.

В этой лабораторной работе мы создадим файл описания зоны. *IP адреса машин вы должны получить у преподавателя*.

СОЗДАНИЕ ФАЙЛА.

В директории */var/named/chroot/var/named* создадим файл описания зоны *master.st1.kryukov.biz* следующего содержания:

```
$TTL 1D
@       IN      SOA     ns.st1.kryukov.biz. (
                student.st1.kryukov.biz.
                2008052800 ; тут напишите
                                ; свой серийный номер
                24H 20M 2W 1D )

@       IN      NS      ns.st1.kryukov.biz.
@       IN      NS      cosmos.kryukov.biz. ;или имя вашего
                                            ;slave сервера
@       IN      MX      5 ns.st1.kryukov.biz.
```

```
ns        IN      A        IP ; вместо IP напишите IP адрес
                                ; который вам даст преподаватель
www       IN      CNAME    ns
mail      IN      CNAME    ns
```

В этом файле есть маленький нюанс. Дело в том, что ваш сервер стоит во внутренней сети за моим сервером и не имеет реального IP адреса. Поэтому в качестве IP адреса машины *ns* вам надо будет указывать IP моего сервера (*спросите IP у преподавателя*). А я, все пакеты, приходящие на порт 53, буду передавать на вашу машину.

ПРОВЕРКА СИНТАКСИСА.

Проверяем синтаксис.

```
# named-checkzone st1.kryukov.biz \
/var/named/chroot/var/named/master.st1.kryukov.biz
```

УПРАВЛЕНИЕ DNS СЕРВЕРОМ.

Запускать, останавливать и использовать другие возможности DNS сервера можно при помощи различных программ:

Поставляемой с BIND 9 программой rndc.
При помощи стартовых скриптов системы инициализации.
При помощи сигналов.

УПРАВЛЕНИЕ DNS СЕРВЕРОМ ПРИ ПОМОЩИ ПРОГРАММЫ RNDC.

Для управления DNS сервером BIND можно использовать программу rndc. Она может делать с сервером практически всё. Единственно, что она не может — это запускать DNS сервер.

В таблице показаны основные команды программы.

Команда	Описание
reload	Перечитать все конфигурационные файлы и файлы описания зон.
reload zone [class [view]]	Перечитать определенную зону.
reconfig	Перечитать конфигурационный файл и загрузить только новые зоны.
stats	Записать статистику сервера в специальный файл named.stats
dumpdb	Сохранить содержимое кэш DNS сервера в файл named_dump.db
stop	Сохранить внесенные изменения и остановить DNS сервер.
halt	То же, что и выше, но без сохранения.
trace	Увеличить уровень отладки на один. Если сервер работал в обычном режиме, при включении отладки создается файл named.run, в который будет попадать вся отладочная информация.
trace level	Установить заданный уровень отладки.

notrace	Выключить режим отладки.
flush	Очистить кэш DNS сервера.

ЗАПУСК DNS СЕРВЕРА.

Для включения DNS сервера необходимо запустить на выполнение демон *named*. Это можно сделать вручную в командной строке или при помощи стартовых скриптов системы инициализации.

Если использовать стартовые скрипты, то запуск сервера происходит при помощи следующих команд:

RedHat Linux (CentOS):

```
service named start
```

SuSE Linux:

```
rcnamed start
```

Slackware Linux:

```
/etc/rc.d/rc.bind start
```

После запуска сервера обязательно проконтролируйте, какую информацию он поместил в файлы журнальной регистрации */var/log/messages*:

```
tail -25 /avr/log/messages
```

В файле обязательно должны быть строки, показывающие на каких интерфейсах работает DNS сервер. Например, такие:

```
Jul 12 14:31:53 master named[476]: listening on IPv4 interface lo, 127.0.0.1#53
Jul 12 14:31:53 master named[476]: listening on IPv4 interface eth0, 192.168.23.101#53
```

Если в файле написано, что сервер не может открыть 53 порт — это означает, что вы запустили вторую копию программы. Она запустилась и не смогла открыть порты на прослушивание потому, что они уже заняты первым экземпляром программы. В этом случае необходимо выключить обе программы и запустить named по новой.

Так же в файле журнальной регистрации необходимо контролировать информацию о загрузке зоны. Например:

```
Jul 12 14:31:53 cosmos named[476]: zone kryukov.biz/IN: loaded serial 2008052200
```

Если вы такой строки не увидите — это означает, что зона не загрузилась. Наиболее вероятная причина — имя файла описания зоны, написанное в файле *named.conf* не совпадает с именем файла в файловой системе. Проверьте правильность написания имени файла описания зоны.

ЗАВЕРШЕНИЕ РАБОТЫ DNS СЕРВЕРА.

Завершение работы DNS сервера возможно тремя способами.

При помощи сигнала:

```
killall named
```

При помощи стартового скрипта, которому в качестве аргумента передается параметр stop.

```
service named stop
```

При помощи программы rndc:

```
rndc stop
```

ПЕРЕЗАГРУЗКА КОНФИГУРАЦИОННЫХ ФАЙЛОВ И ФАЙЛОВ ОПИСАНИЯ ЗОН.

Если демону послать сигнал *HUP*, он перечитает все конфигурационные файлы и файлы описания зон.

```
killall -HUP named
```

Программа *rndc* позволяет более гибкое управление DNS сервером.

```
rndc reload
```

Будут перечитаны все конфигурационные файлы и файлы описания зон.

Если необходимо перегрузить только определенную зону, тогда лучше использовать команду *reload zone*.

```
rndc reload zone kryukov.biz
```

НАСТРОЙКА КЛИЕНТА DNS.

Конфигурационный файл клиента DNS в UNIX — */etc/resolv.conf*. В этом файле указываются DNS сервера, которым будет обращаться клиент с вопросами преобразования имён.

DNS сервера определяются при помощи параметра *nameserver*. Для каждого DNS сервера необходимо определять отдельный параметр *nameserver*, но не более трёх.

Например:

```
nameserver 10.10.108.20
nameserver 10.10.1.1
```

Также в этом файле можно определить один параметр *search*, при помощи которого можно вводить не полные имена. В качестве параметров *search* используют имена доменов, разделенные пробелами. Например:

```
search kryukov.biz kryukov.ru
```

Если программе *ping* в качестве параметра передать имя *www*, при передаче его клиенту DNS на преобразование, клиент не обнаружит в нем точек и поймет, что это не FQDN имя. Клиент сначала подставит к имени домен *kryukov.biz* и отправит его на преобразование. Если в ответ он получит сообщение об ошибке, клиент подставит следующий в списке домен.

На машине на которой работает сервер DNS, тоже необходимо настраивать клиент. DNS сервер — это ещё одна программа, и если она работает на вашей машине — это не значит, что клиент DNS автоматически будет обращаться к этому серверу. Клиенту придётся явным образам указать на вашу машину.

НАСТРОЙКА FIREWALL ДЛЯ РАБОТЫ DNS СЕРВЕРА.

При запуске DNS сервер открывает на прослушивание 53 порт.

В своей работе DNS сервер использует два транспортных протокола: *tcp* и *udp*. По *tcp* происходит передача зон при обновлении с master DNS и передача запросов, если вы настроили шифрование при передаче. По *udp* передаются запросы к серверам.

Соответственно, в firewall вам придётся открывать 53 порт протокол *udp* и *tcp*.

ЛАБОРАТОРНАЯ РАБОТА. ЗАПУСК DNS СЕРВЕРА.

Задача этой лабораторной работы:

- Запустить DNS сервер.
- Настроить клиента DNS.

ЗАПУСКАЕМ DNS СЕРВЕР.

```
# service named start
```

Смотрим конец файла журнальной регистрации */var/log/messages*. Убеждаемся, что сервер открыл на прослушивание необходимые порты и загрузил зоны.

```
# tail -30 /var/log/messages
```

Делаем так, что бы сервер запускался при старте компьютера.

```
# chkconfig named on
```

Не забываем открыть доступ к DNS серверу в firewall. В файле *~/bin/rc.fw*, в функции *init* добавляем следующие строки:

```
### DNS server
$IPT -A INPUT -p tcp --dport 53 -j ACCEPT
$IPT -A INPUT -p udp --dport 53 -j ACCEPT
```

Изменяем правила firewall в оперативной памяти:

```
rc.fw init
```

На вашей домашней машине, проверяем, есть ли в сети машина с именем *ns.st1.kryukov.biz* (тут вы подставите имя машины, которое вам дал преподаватель). Если вы дома работаете в Linux, тогда это делается так:

```
host ns.st1.kryukov.biz
```

Если в Windows, запускаете программу *cmd* и в командной строке вводим:

```
nslookup ns.st1.kryukov.biz
```

Если в ответ вы получили IP адрес машины, значит DNS сервер работает и мы можем настраивать остальные службы.

Если вы не получили IP адрес машины, то появятся некоторые сложности. DNS сервер провайдера закеширует отрицательный ответ (машины ns.st1.kryukov.biz нет в природе) и будет давать отрицательные ответы в течении суток. Поэтому, если вы в дальнейшем все же правильно настроите ваш DNS сервер, сервер провайдера в течении суток все равно будет давать отрицательный ответ.

Для решения этой проблемы вам некоторое время придется напрямую обращаться к вашему DNS серверу. В Linux:

```
dig ns.st1.kryukov.biz @IP
```

В Windows:

```
nslookup ns.st1.kryukov.biz IP
```

Вместо *IP* следует указать IP адрес вашего сервера. *О программе dig и host будет рассказано в следующем разделе*.

К сожалению, все остальные программы вашей машины обращаются только к клиенту DNS вашей машины и в течении суток не будут «видеть» машины *ns* в Интернет. Это издержки системы DNS и с ними приходится мириться.

НАСТРАИВАЕМ КЛИЕНТ DNS.

В файле */etc/resolv.conf* изменяете строку с параметром *nameserver*. Вместо старого IP адреса, укажите *127.0.0.1*.

```
nameserver 127.0.0.1
```

В файле должен быть только один параметр *nameserver*.

НАСТРОЙКА ПОДДЕРЖКИ SLAVE ЗОНЫ.

В BIND slave зона настраивается очень просто, достаточно описать инструкцию *zone* в файле *named.conf*.

```
zone "kryukov.biz" IN {
        type slave;
        masters { 84.19.178.61; };
        file "slave.kryukov.biz";
};
```

Параметр *masters* является обязательным. Он определяет IP адрес master сервера, к которому наш DNS сервер будет обращаться за информацией о зоне.

Параметр *file* не является обязательным, но его желательно описывать. В этом параметре определяется файл, в котором будет храниться информация о зоне. Если файл не определен, то информация будет храниться только в оперативной памяти и при перезапуске DNS сервер будет вынужден снова получить содержимое зоны с master сервера.

В нашем курсе не предусмотрено включение поддержки slave зоны. Но если кто то обратиться к вам с просьбой поднять slave зону для домена, не отказываете им в этом ☺. Вам всего лишь придётся написать пару строк в файле *named.conf* — это не так сложно. Заодно потренируетесь. Только не забудьте, что после окончания курса, ваша виртуальная машина будет удалена. Предупредите об этом хозяина master сервера, для которого вы поднимаете зону.

Кстати, мы можем оставить вашу виртуальную машину работать дальше, у нашей компании есть услуги хостинга. Тогда всё, что вы сейчас настроили, будет продолжать работать. Конечно, вам придётся все настройки переделать на ваш собственный домен.

ОТЛАДКА.

Для тестирования и отладки DNS сервера можно использовать несколько способов:

- Просмотр содержания журнальных файлов.
- Запуск сервера в отладочном режиме.
- Использование программ, посылающих запросы DNS серверу.

При старте демон *named* помещает в журнальные файлы различную информацию:

- Интерфейсы, на которых демон слушает запросы и команды.
- Какие зоны были загружены или перезагружены.

К сожалению, информация, которая поступает в журнальные файлы, не является подробной и иногда ее не хватает для определения ошибок работы DNS сервера.

BIND позволяет выбирать, какая информация и в каком объеме должна попадать в журнальные файлы. Для этого используется инструкция *logging*. Но ее применение не оправдано из-за больших затрат времени на настройку. Кроме того, отладочный режим сервера необходим достаточно редко. Поэтому для отладки следует использовать программу *rndc* и/или программы посылающие, запросы серверу DNS из командной строки.

ИСПОЛЬЗОВАНИЕ УРОВНЕЙ ОТЛАДКИ DNS СЕРВЕРА.

Уровни отладки DNS сервера обозначаются числами от 0 до 11. Чем больше число, тем более подробную информацию будет содержать вывод программы.

Уровень 0 выключает отладочный режим. Уровни 1 и 2 используются для отладки конфигурационных файлов и файлов описания зон. Остальные уровни отладки используются разработчиками BIND.

Самый простой способ включения отладочного режима — запуск демона *named* с опцией *-d*.

```
named -d2
```

После запуска демона вся отладочная информация будет помещаться в файл */var/named/named.run*. В случае запуска сервера в chroot, например в RedHat или CentOS, файл *named.run* будет создаваться в директории */var/named/chroot/var/named/data*.

Если DNS сервер уже работает, включить режим отладки можно при помощи утилиты *rndc*.

```
rndc trace 2
```

В этом случае тоже появляется файл *named.run*.

Для отключения отладочного режима *rndc* вызывают с параметром *notrace*:

```
rndc notrace
```

Отладочный режим позволяет определить ошибки связанные с неправильным именем файла описания зоны. Например, если после запуска сервера он не загружает зону, т.е. в файле журнальной регистрации нет информации о загрузке зоны. Скорее всего имя файла описания зоны в инструкции *zone* не совпадает с именем файла, созданного в файловой системе. Информацию о такой ошибке можно увидеть только в файле *named.run*.

В любом случае, прежде чем использовать режимы отладки, настоятельно рекомендуется произвести поиск синтаксических ошибок при помощи программ named-checkconf *и* named-checkzone.

ИСПОЛЬЗОВАНИЕ ПРОГРАММ DIG И HOST.

Достаточно большое количество ошибок в работе DNS сервера связаны с обыкновенными опечатками в полях с данными. Такие ошибки не отлавливаются программами проверки синтаксиса и не помещаются в отладочную информацию в файле *named.run*.

Самый распространенный пример такой ошибки — забыли поставить точку в конце имени машины. Это не синтаксическая ошибка, поэтому программы *named-checkconf* и *named-checkzone* её не увидят. Да и отладочный режим нам ничем помочь не сможет.

В этом случае придётся посылать запросы непосредственно серверу и смотреть какие ответы он возвращает. Для посылки запросов удобно использовать программу *dig*. Почему не *nslookup*[1]? *Dig* удобнее. В отличии от *nslookup* она выдаёт информацию в формате файла описания зоны и никак её не приукрашивает.

Самый простой вариант, получение IP адреса машины. Для этого программе в качестве аргумента командной строки необходимо указать имя интересующей вас машины.

```
# dig cosmos.kryukov.biz


; <<>> DiG 9.3.3rc2 <<>> cosmos.kryukov.biz
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33433
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 2


;; QUESTION SECTION:
;cosmos.kryukov.biz.            IN      A


;; ANSWER SECTION:
cosmos.kryukov.biz.  86400 IN      A       84.19.178.61
```

1 Nslookup используется для посылки запросов в Windows. В Linux она тоже присутствует.

```
;; AUTHORITY SECTION:
kryukov.biz.          86400 IN      NS      cosmos.kryukov.biz.
kryukov.biz.          86400 IN      NS      dns1.zenon.net.
kryukov.biz.          86400 IN      NS      dns2.zenon.net.

;; ADDITIONAL SECTION:
dns1.zenon.net.       19224 IN      A       195.2.64.38
dns2.zenon.net.       19224 IN      A       195.2.83.38

;; Query time: 0 msec
;; SERVER: 84.19.178.61#53(84.19.178.61)
;; WHEN: Tue Jun 17 13:34:53 2008
;; MSG SIZE  rcvd: 145
```

Давайте подробно рассмотрим, что вывела программа *dig* на стандартный вывод. Во первых вспомните, что в файле описания зоны символ точка с запятой — это комментарий. Поэтому многие строки просто комментарии, поясняющие выводимую информацию.

Вывод программы разбит на секции. В самом начале вы видите *QUESTION SECTION*, в которой описывается запрос, который будет обрабатывать программа. В нашем случае мы пытаемся получить IP адрес машины *cosmos.kryukov.biz*, т.е. получить запись *A* у сервера DNS.

В *ANSWER SECTION* нам показывается ответ, который был получен от сервера. Ответ показан в формате файла описания зоны, т.е. он содержит все поля записи *A*. Обратите внимание на второе поле — время жизни записи в кеш DNS сервера. Если повторить этот запрос несколько раз, то вы увидите, как значение этого поля уменьшается.

Секция *AUTHORITY SECTION* содержит информацию об ответственных за зону DNS серверах. А в *ADDITIONAL SECTION* показаны IP адреса этих серверов.

В самом конце показан сервер, который ответил на наш запрос.

Рассмотрим другой пример, запросим имя машины www.kryukov.biz.

```
# dig www.kryukov.biz

; <<>> DiG 9.3.3rc2 <<>> www.kryukov.biz
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52905
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 2

;; QUESTION SECTION:
;www.kryukov.biz.               IN      A

;; ANSWER SECTION:
www.kryukov.biz.    86400 IN      CNAME   cosmos.kryukov.biz.
cosmos.kryukov.biz. 86400 IN      A       84.19.178.61

;; AUTHORITY SECTION:
```

```
kryukov.biz.        86400 IN      NS      dns2.zenon.net.
kryukov.biz.        86400 IN      NS      cosmos.kryukov.biz.
kryukov.biz.        86400 IN      NS      dns1.zenon.net.

;; ADDITIONAL SECTION:
dns1.zenon.net.     18512 IN      A       195.2.64.38
dns2.zenon.net.     18512 IN      A       195.2.83.38

;; Query time: 0 msec
;; SERVER: 84.19.178.61#53(84.19.178.61)
;; WHEN: Tue Jun 17 13:46:45 2008
;; MSG SIZE  rcvd: 163
```

Тут мы тоже пытались получить IP адрес машины. Но в *ANSWER SECTION* уже две строки. Оказывается, что машина *www* — это просто ссылка на каноническое имя *cosmos* (первая строка в ответе), а машина *cosmos* имеет указанный IP адрес (вторая строка в ответе).

При помощи *dig* можно получать и другие записи зоны, просто необходимо это явно указать. Например, как узнать email человека, отвечающего за зону *kryukov.biz*? Надо посмотреть запись *SOA*, в которой он указан.

```
# dig kryukov.biz SOA
;; QUESTION SECTION:
;kryukov.biz.                   IN      SOA
;; ANSWER SECTION:
kryukov.biz.   86400 IN SOA  cosmos.kryukov.biz.
       artur.kryukov.biz. 2008052700 86400 1200 1209600 86400
```

Для экономии места я привел краткий вывод программы.

По умолчанию, программа *dig* обращается к тому же серверу, что и клиент DNS вашей машины. Если вы решили послать запрос другому серверу, его необходимо указать явно.

```
# dig kryukov.biz SOA @84.19.179.61
;; QUESTION SECTION:
;kryukov.biz.                   IN      SOA
;; ANSWER SECTION:
kryukov.biz.   86400 IN SOA  cosmos.kryukov.biz. artur.kryukov.biz. 2008052700 86400
1200 1209600 86400

;; Query time: 134 msec
;; SERVER: 84.19.179.61#53(84.19.179.61)
;; WHEN: Tue Jun 17 14:23:58 2008
;; MSG SIZE  rcvd: 171
```

Программа host выдает ответы не в формате зоны.

```
# host cosmos.kryukov.biz
cosmos.kryukov.biz has address 84.19.178.61

# host www.kryukov.biz
www.kryukov.biz is an alias for cosmos.kryukov.biz.
```

```
cosmos.kryukov.biz has address 84.19.178.61
```

ЗОНЫ ОБРАТНОГО ПРЕОБРАЗОВАНИЯ.

Все примеры, которые мы до сих пор рассматривали, относились к прямому преобразованию, когда имя машины преобразовывалось в IP адрес. DNS сервера позволяют осуществлять и обратное преобразование — IP адрес в имя машины.

Давайте посмотрим на следующий пример. Существует машина *cosmos.kryukov.b*iz, это полностью квалифицированное доменное имя машины (FQDN), которое состоит из вух частей:

cosmos — это имя.
kryukov.biz — домен.

Эта машина имеет IP адрес — *84.19.178.61*. Который тоже состоит из двух частей: адрес сети и адрес машины в этой сети. Какая часть является адресом машины или сети, определяется при помощи маски подсети. Предположим, что маска *255.255.255.0*. Тогда адрес сети — *84.19.178*, а адрес машины — *61*.

Правда, похоже на FQDN? Тоже две части, общая и частная. Но проблема заключается в том, что DNS сервера не понимают адрес сети и адрес машины в сети. Они умеют работать только с FQDN именами. Поэтому нам придется преобразовать IP адрес в формат понятный серверам DNS.

Делается это очень просто. Сначала IP адрес зеркально переворачивают. Т.е. *84.19.178.61*, превращается в *61.178.19.84*. К этому адресу добавляется имя специального домена *in-addr.arpa* и получается FQDN IP адреса.

```
61.178.19.84.in-addr.arpa
```

После такого преобразования мы видим, что имя машины — *61*, а имя домена *178.19.84.in-addr.arpa*.

Теперь мы можем указать новую зону ответственности для нашего сервера: *178.19.84.in-addr.arpa*. Т.е. в файле *named.conf* написать инструкцию *zone*. Например, такую:

```
zone "178.19.84.in-addr.arpa" IN {
type master;
file "master.178.19.84.in-addr.arpa";
allow-update { none;};
allow-query { any; };
allow-tranfer { IP_SLAVE; };
};
```

Разумеется, придется создать файл описания зоны. В любом файле описания зоны будут две обязательные записи: *SOA* и *NS*. А вот относительно других записей стоит подумать, использовать их или нет.

Запись *MX* определяет почтовый сервер, принимающий почту для домена. Вы когда нибудь видели подобные email: *artur@178.19.84.in-add.arpa*? Я думаю, что никогда не увидите. Поэтому запись *MX* в зонах обратного преобразования не используется.

Запись *A* преобразует имя в IP адрес, а у нас совсем другая задача. Поэтому *A* мы тоже использовать не будем.

Для обратного преобразования предназначена специальная запись — *PTR*. Первое поле записи — это IP адрес, но не просто адрес, а после го преобразования. Т.е. если вы напишите 84.19.178.61 — это будет ошибкой. Параметр у записи всего один — имя машины.

Что бы продемонстрировать, как выглядит эта запись, воспользуемся программой *dig*. Для того, что бы получить имя по IP адресу, при запуске программы необходимо указывать параметр *–x*.

```
# dig -x 84.19.178.61


;; QUESTION SECTION:
```

```
;61.178.19.84.in-addr.arpa.     IN  PTR

;; ANSWER SECTION:
61.178.19.84.in-addr.arpa. 86400 IN PTR     cosmos.kryukov.biz.

;; AUTHORITY SECTION:
178.19.84.in-addr.arpa. 86400   IN  NS      ns.keyweb.de.
178.19.84.in-addr.arpa. 86400   IN  NS      ns2.keyweb.de.

;; ADDITIONAL SECTION:
ns.keyweb.de.           22698   IN  A       62.141.60.15
ns2.keyweb.de.          22698   IN  A       62.141.49.15

;; Query time: 20 msec
;; SERVER: 84.19.178.61#53(84.19.178.61)
;; WHEN: Tue Jun 17 15:10:33 2008
;; MSG SIZE  rcvd: 151
```

Как видите, нам была показана запись *PTR*. Кстати, есть интересный вопрос — кто должен поднимать зону обратного преобразования? Ответ простой — хозяин пула IP адресов. В нашем случае — это мой провайдер. Посмотрите на *AUTHORITY SECTION*, там написаны имена DNS серверов, отвечающих за этот домен.

Я когда арендовал машину у провайдера попросил его, что бы он прописал в зоне обратного преобразования для домена *178.19.84.in-addr.arpa* мою машину. Провайдер должен это делать без дополнительной оплаты и по возможности быстро.