

# DNS HOWTO

---

Nicolai Langfeldt janl@math.uio.no, перевод Alex Ott ott@phtd.tpu.edu.ru v2.2, 11 Февраля 1999

Как стать за короткое время администратором DNS.

## Содержание

<b>1</b>	<b>Преамбула</b>	<b>1</b>
<b>2</b>	<b>Введение</b>	<b>2</b>
<b>3</b>	<b>Настройка кеширующего сервера имен.</b>	<b>3</b>
<b>4</b>	<b>Простой домен.</b>	<b>7</b>
<b>5</b>	<b>Пример реального домена</b>	<b>19</b>
<b>6</b>	<b>Сопровождение</b>	<b>24</b>
<b>7</b>	<b>Преобразование настроек named из версии 4 в версию 8</b>	<b>26</b>
<b>8</b>	<b>Вопросы и ответы</b>	<b>27</b>
<b>9</b>	<b>Как стать более знающим администратором DNS.</b>	<b>30</b>

**Примечание переводчика:** Шлите мне любые комментарии и замечания, даже небольшие.

## 1 Преамбула

Ключевые слова: DNS, bind, bind-4, bind-8, named, dialup, ppp, slip, isdn, Internet, домен (domain), имя (name), сервера (hosts), разрешение имен (resolving), кэширование.

Этот документ является частью Linux Documentation Project.

### 1.1 Официальная часть

Авторские права (c) Nicolai Langfeldt, 1995-1999. Не исправлять без изменения авторских прав, распространяется свободно при сохранении уведомления об авторских правах.

### 1.2 Признательности и запрос о помощи

Я хочу поблагодарить Arnt Gulbrandsen, который постоянно читает черновики этого документа и дает много советов и рекомендаций. Я также хочу поблагодарить людей, которые прислали мне свои предложения и замечания.

Этот документ никогда не будет закончен, пожалуйста посылайте мне сообщения о ваших успехах и неудачах — это сделает данный документ лучше. Также посылайте комментарии и/или вопросы на адрес janl@math.uio.no. Если вы посылаете сообщение и хотите получить ответ, то пожалуйста проявите вежливость и *убедитесь*, что обратный адрес является правильным и работает. Также **пожалуйста** прочитайте раздел 8 (Вопросы и ответы) до того как посылать сообщение. Также я понимаю только Английский и Норвежский языки. (*От переводчика: вопросы на русском языке вы можете посылать мне на адрес, приведенный в начале документа*).

Если вы хотите перевести этот документ, то пожалуйста сообщите мне об этом, что бы я мог отслеживать на каких языках этот документ был опубликован, и также чтобы я мог известить вас, когда этот документ будет обновлен.

### 1.3 Посвящение

Этот документ посвящен Anne Line Norheim Langfeldt. Хотя она вероятно никогда его не прочтает, поскольку она не относится к классу девушек, увлеченных данной проблемой.

## 2 Введение

### Что это такое, и чем этот документ не является

DNS — это Доменная Система Имен (Domain Name System). DNS преобразует имена машин в IP-номера, которые являются адресами машин, она преобразует из имен в адреса и из адресов в имена. Этот документ показывает как определить такие преобразования, используя систему на базе Linux. Преобразование — это просто создание ассоциации между двумя вещами, в нашем случае между именем машины, таким как `ftp.linux.org`, и номером IP этой машины, например `199.249.150.4`.

DNS является, для непосвященных (для вас ;-), одной из самых непрозрачных областей сетевого администрирования. Этот документ постарается сделать некоторые вещи понятнее. Здесь описывается как настроить *простой* сервер DNS. Начав с настройки кэширующего сервера, мы перейдем к настройке основного сервера DNS для домена. Для более сложных настроек вы можете посмотреть раздел 8 (Вопросы и ответы) этого документа. Если он не описывает, то что вам нужно, то вам понадобится *прочитать* полную документацию. Я возвращусь к тому из чего состоит эта полная документация в 9 (последней главе).

До начала этого процесса, вы должны настроить вашу машину так, чтобы вы могли выполнять команду `telnet` на нее/с нее, и производить любые сетевые соединения, и вам совершенно необходимо иметь возможность выполнять команду `telnet 127.0.0.1` и входить на свою собственную машину (протестируйте это сейчас!). Вам также необходимо иметь правильно настроенные файлы `/etc/nsswitch.conf` (или `/etc/host.conf`), `/etc/resolv.conf` и `/etc/hosts`, поскольку я не буду объяснять их функции в этом документе. Если работа сети у вас еще не настроена, то в *NET-3-HOWTO* и/или *PPP-HOWTO* вам объяснят как это сделать. Прочитайте эти документы.

Когда я говорю 'ваша машина' я подразумеваю ту машину, на которой вы пытаетесь настроить DNS. А ни какая другая машина, которую вы могли вовлечь в ваши сетевые эксперименты.

Я предполагаю, что вы не находитесь за `firewall` любого типа, который блокирует запрос имен. Если вы все-таки находитесь за `firewall`, то вам необходима специальная настройка, смотрите раздел 8 (Вопросы и ответы).

В Unix задача разрешения имен выполняется программой, называемой `named`. Она является частью пакета "bind", который сопровождается Paul Vixie для Internet Software Consortium. `Named` включен в большинство дистрибутивов Linux, и обычно устанавливается как `/usr/sbin/named`. Если у вас есть `named`, то вы вероятно сможете использовать его; Если у вас его нет, то вы можете получить исполняемые файлы с основных ftp серверов Linux, или взять последнюю и наилучшую версию исходных текстов с `ftp.isc.org:/isc/bind/src/cur/bind-8/`. Этот документ описывает `bind` версии 8. Старые версии этого документа, описывающие `bind` версии 4, еще доступны по адресу `http://www.math.uio.no/~janl/DNS/`, в том случае, если вы используете `bind` 4. Если справочная страница `named` ссылается (в самом конце, в разделе FILES) на файл `named.conf`, то у вас стоит `bind` 8, если она ссылается на `named.boot`, то у вас `bind` 4. Если у вас установлен `bind` версии 4 и вы осознаете необходимость сохранения безопасности, то вы действительно должны обновить вашу версию до версии 8.

DNS является базой данных во всемирной сети. Поэтому заботьтесь о том, что вы помещаете в нее. Если вы будете помещать в нее всякий хлам, то и вы и другие люди также получат всякий

хлам. Храните вашу базу DNS в аккуратности и вы получите хорошее обслуживание с ее стороны. Учитесь как использовать ее, администрировать и отлаживать, и вы будете еще одним хорошим администратором, хранящим сеть от сбоев в результате неумелого управления.

В этом документе я констатирую набор вещей, которые в самом деле не являются полностью правильными (они являются по крайней мере наполовину правильными). Все это сделано в целях упрощения. Эти вещи (вероятно ;-)) будут работать, если вы поверите, тому что я скажу.

**Совет:** Сделайте резервные копии всех файлов, которые я буду советовать изменить, если у вас они уже есть, так что если ничего не будет работать, то вы можете вернуться назад к рабочей конфигурации.

### 3 Настройка кеширующего сервера имен.

**Это первый шаг в настройке DNS, очень полезный для dialup пользователей**

Кеширующий 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.

Для начала вам нужен файл, названный `/etc/named.conf`. Из него `named` читает информацию при старте. Сейчас он должен просто содержать следующие строки:

---

```
// Файл настроек для только кеширующего сервера

options {
    directory "/var/named";

    // Раскомментируйте следующую строку, если вы
    // работаете через firewall и система не работает:

    // query-source port 53;
};

zone "." {
    type hint;
    file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "pz/127.0.0";
};
```

---

Строка `'directory'` задает где искать файлы. Все файлы используемые впоследствии, будут именоваться относительно этой директории. Таким образом `pz` — это директория в директории `/var/named`, т.е., `/var/named/pz`. `/var/named` — это правильная директория согласно *Linux File system Standard (Стандарту файловой системы Linux)*.

Файл названный `/var/named/root.hints` должен находиться в указанной директории. Он должен содержать следующую информацию:

---

```
; Здесь может быть комментарии, если у вас уже есть этот файл.
; Если их нет, то не беспокойтесь.
.                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 изменяется время от времени и эта часть в дальнейшем *должна* сопровождаться. Смотрите [6](#) (раздел по сопровождению) для того, чтобы узнать как хранить эту информацию соответствующей действительности.

Следующий раздел в `named.conf` — это последняя зона. Я объясню как она используется в следующих разделах, сейчас просто создайте файл, названный `127.0.0` в поддиректории `rz`:

```

@                IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                1        ; Serial
                8H       ; Refresh
                2H       ; Retry
                1W       ; Expire
                1D)      ; Minimum TTL
                NS       ns.linux.bogus.
1                PTR     localhost.

```

Далее вам необходимо, чтобы ваш файл `/etc/resolv.conf` выглядел примерно так:

```

search subdomain.your-domain.edu your-domain.edu
nameserver 127.0.0.1

```

Строка `'search'` задает в каких доменах должен идти поиск машин с которыми вы хотите соединиться. Строка `'nameserver'` указывает адрес вашего сервера имен, в нашем случае это ваша собственная машина, поскольку на ней запущен `named` (`127.0.0.1` это правильный адрес, также никаких проблем, если ваша машина имеет другой адрес). Если вы хотите перечислите несколько серверов имен, то поместите их по одному в строку со словом `'nameserver'` для каждого. (Замечание: `Named` никогда не читает этот файл, это делает программа `resolver`, которая использует `named`).

Проиллюстрируем как это работает: Если клиент пытается найти машину с именем `foo`, то сначала программа пытается найти машину с полным именем `foo.subdomain.your-domain.edu`, затем с именем `foo.your-domain.edu`, и в конце концов `foo`. Если клиент пытается найти `sunsite.unc.edu`, то сначала пробуются `sunsite.unc.edu.subdomain.your-domain.edu` (да это глупо, но вот так это работает), затем `sunsite.unc.edu.your-domain.edu`, и в конце концов `sunsite.unc.edu`. Вы можете не помещать слишком много доменов в строку поиска, поскольку поиск в них займет слишком много времени.

Пример предполагает, что вы находитесь в домене `subdomain.your-domain.edu`, и ваша машина вероятно называется `your-machine.subdomain.your-domain.edu`. Строка поиска не должна содержать ваш TLD (Top Level Domain (Домен Верхнего Уровня), 'edu' в нашем случае). Если вам необходимо часто соединяться с машиной в другом домене, то вы можете добавить этот домен в строку поиска, примерно вот так:

---

```
search subdomain.your-domain.edu your-domain.edu other-domain.com
```

---

и так далее. Очевидно, что вам необходимо поместить в эту строку имена настоящих доменов, вместо вышеприведенных. Пожалуйста заметьте отсутствие точки в конце имени домена. Это важно, пожалуйста заметьте отсутствие точки в конце имени доменов.

Далее в зависимости от вашей версии `libc` вам необходимо вносить исправления либо в файл `/etc/nsswitch.conf`, либо в файл `/etc/host.conf`. Если у вас уже есть файл `nsswitch.conf`, то значит мы будем вносить исправления в него, если же его нет, то мы будем вносить изменения в файл `host.conf`.

#### **/etc/nsswitch.conf**

Это длинный файл описывающий как получить разные типы данных, из какого файла или базы данных. В начале он обычно содержит полезные комментарии, которые вы должны учесть при чтении этого файла. После того, как вы найдете строку начинающуюся с 'hosts:', вы должны увидеть:

---

```
hosts:      files dns
```

---

Если в этом файле нет строки начинающейся с 'hosts:', то поместите вышеприведенную строку в файл. Эта строка указывает программам сначала выполнять поиск в файле `/etc/hosts`, а затем просматривать DNS в соответствии с порядком указанным в файле `resolv.conf`.

#### **/etc/host.conf**

Этот файл вероятно содержит разные данные, одна из строк должна начинаться со слова `order` и выглядеть примерно так:

---

```
order hosts,bind
```

---

Если строки с 'order' нет, то вы должны ее добавить. Она заставляет подпрограмму разрешения имен сначала посмотреть в файле `/etc/hosts`, а затем сделать запрос к серверу имен (который в `resolv.conf` указан как машина с адресом `127.0.0.1`).

### **3.1 Запуск named**

После этих приготовлений пришло время запуска `named`. Если вы используете dialup соединение, то сначала производите подключение. Наберите 'ndc start' без опций, и нажмите клавишу return. Если это не работает, то попробуйте следующую команду '/usr/sbin/ndc start'. Если опять попытка не удалась, то смотрите раздел 8 (Вопросы и ответы). Если вы посмотрите в файл сообщений

syslog (обычно названный /var/adm/messages, но может быть другая директория /var/log и другой файл syslog в которые необходимо посмотреть) во время запуска named (выполните команду `tail -f /var/log/messages`), то вы должны увидеть что-то подобное следующему: (строки заканчивающиеся на \ продолжаются на следующей строке)

```
Feb 15 01:26:17 roke named[6091]: starting.  named 8.1.1 Sat Feb 14 \
00:18:20 MET 1998 ^Ijanl@roke.uio.no:/var/tmp/bind-8.1.1/src/bin/named
Feb 15 01:26:17 roke named[6091]: cache zone "" (IN) loaded (serial 0)
Feb 15 01:26:17 roke named[6091]: master zone "0.0.127.in-addr.arpa" \
(IN) loaded (serial 1)
Feb 15 01:26:17 roke named[6091]: listening [127.0.0.1].53 (lo)
Feb 15 01:26:17 roke named[6091]: listening [129.240.230.92].53 (ipp0)
Feb 15 01:26:17 roke named[6091]: Forwarding source address is [0.0.0.0].1040
Feb 15 01:26:17 roke named[6092]: Ready to answer queries.
```

Если есть какие-нибудь сообщения об ошибках, то значит вы что-то сделали неправильно. Named укажет в каком файле ошибка (я надеюсь, что это один из файлов `named.conf` и `root.hints` :-). Завершите выполнение `named` и проверьте файлы конфигурации.

Теперь вы можете протестировать вашу настройку. Запустите `nslookup` для проверки вашей работы.

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1

>
```

Если это выглядит так, то значит вы заставили систему работать. Мы так надеемся. Если что-то другое, то вернитесь назад и все проверьте. Каждый раз когда вы изменяете файл `named.conf`, вам необходимо перезапустить `named`, используя команду `ndc restart`.

Теперь мы можем ввести запрос на поиск информации. Попробуйте найти машину близкую к вам. `pat.uio.no` находится близко от меня, в Университете Осло:

```
> pat.uio.no
Server:  localhost
Address: 127.0.0.1

Name:    pat.uio.no
Address: 129.240.130.16
```

Сейчас `nslookup` попросит ваш `named` посмотреть информацию о машине `pat.uio.no`. Затем он соединится с одним из серверов имен, перечисленных в вашем файле `root.hints`, и запросит у него путь к данной машине. Это может занять какое-то время, до того как вы получите результаты, поскольку система может понадобится попробовать найти заданную машину во всех доменах, перечисленных в вашем файле `/etc/resolv.conf`.

Если вы запросите то же самое, то вы получите такой ответ:

```
> pat.uio.no
Server:  localhost
Address: 127.0.0.1

Non-authoritative answer:
Name:    pat.uio.no
Address: 129.240.2.50
```

Заметим, что мы в это раз получили сообщение 'Non-authoritative answer:'. Это означает, что `named` в этот раз не делал запрос к внешним серверам имен, информация находится в кеше.

Но кешированная информация может быть устаревшей. Так что он вас информирует об этой (весьма незначительной) опасности сообщением 'Non-authorative answer:'. nslookup выдает это сообщение, когда вы второй раз запрашиваете об одной и той же машине — это знак того, что named кеширует информацию и это значит, что он работает правильно. Вы можете завершить работу nslookup дав команду 'exit'.

### 3.2 Как сделать лучше

В больших, хорошо организованных академических сетях или сетях ISP (Internet Service Provider) вы иногда сможете обнаружить, что администраторы сети настроили иерархию серверов DNS, которые помогут облегчить загрузку внутренней сети, также загрузку внешних серверов. Не так легко узнать находитесь ли вы внутри такой сети или нет. Однако это не совсем важно, и используя DNS-сервер вашего сетевого провайдера как "forwarder" вы можете сделать ответы на запросы быстрее и уменьшить загрузку вашей сети. Если вы используете модем, то это может быть большой победой. Для пользы этого примера мы предполагаем, что ваш сетевой провайдер имеет два сервера имен, который вы можете использовать, с сетевыми номерами 10.0.0.1 и 10.1.0.1. Затем в вашем файле named.conf, внутри открывающего раздела под названием "options" вставьте следующие строки:

```
forward first;
forwarders {
    10.0.0.1;
    10.1.0.1;
};
```

Перезапустите ваш сервер имен и протестируйте его используя nslookup. Все должно работать отлично.

### 3.3 Поздравления

Теперь вы знаете как установить кеширующий сервер имен. Возьмите пива, молока или того, что вы предпочитаете и отпразднуйте это.

## 4 Простой домен.

Как установить свой собственный домен

### 4.1 Но сначала некоторое количество сухой теории

До того как мы в *действительности* начнем этот раздел, я хочу дать вам некоторые теоретические сведения о том как работает DNS и пример ее работы. И вы должны читать дальше, потому что это полезно для вас. Если вы не 'хотите' делать это, то вы по крайней мере должны быстро просмотреть его. Остановитесь, когда вы увидите указания о том, что вы должны делать с вашим файлом named.conf.

DNS — это иерархическая, организованная в виде дерева система. Вершина записывается как '.' и произносится как 'root (корень)'. В . существует некоторое количество Доменов верхнего уровня (Top Level Domains, TLDs), наиболее известными из которых являются ORG, COM, EDU и NET, но на самом деле их намного больше. Точно также как и дерево, они имеют корни и ветви. Если у вас есть некоторое образование в области компьютерных наук, то вы сможете думать о DNS как о дереве поиска, и вы сможете находить узлы, листья и связи между вершинами этого дерева.

При поиске машины, запрос обрабатывается рекурсивно, начиная с корня. Если вы хотите найти адрес машины `prep.ai.mit.edu`, то ваш сервер имен должен найти сервер имен, который обслуживает `edu`. Он запрашивает корневой сервер (`.`) (он уже знает корневые `.` сервера — они перечислены в файле `root.hints`), корневой сервер `.` дает список серверов `edu`:

```
$ nslookup
Default Server: localhost
Address: 127.0.0.1
```

Запросите корневой сервер:

```
> server c.root-servers.net.
Default Server: c.root-servers.net
Address: 192.33.4.12
```

Установите тип запроса (Query type) в NS (записи о серверах имен):

```
> set q=ns
```

Запросите его о `edu`:

```
> edu.
```

Заключительная точка `.` очень важна, она сообщает `nslookup`, что мы запрашиваем о `edu` находящемся прямо под корневым сервером `.` (это ускоряет поиск).

```
edu      nameserver = A.ROOT-SERVERS.NET
edu      nameserver = H.ROOT-SERVERS.NET
edu      nameserver = B.ROOT-SERVERS.NET
edu      nameserver = C.ROOT-SERVERS.NET
edu      nameserver = D.ROOT-SERVERS.NET
edu      nameserver = E.ROOT-SERVERS.NET
edu      nameserver = I.ROOT-SERVERS.NET
edu      nameserver = F.ROOT-SERVERS.NET
edu      nameserver = G.ROOT-SERVERS.NET
A.ROOT-SERVERS.NET      internet address = 198.41.0.4
H.ROOT-SERVERS.NET      internet address = 128.63.2.53
B.ROOT-SERVERS.NET      internet address = 128.9.0.107
C.ROOT-SERVERS.NET      internet address = 192.33.4.12
D.ROOT-SERVERS.NET      internet address = 128.8.10.90
E.ROOT-SERVERS.NET      internet address = 192.203.230.10
I.ROOT-SERVERS.NET      internet address = 192.36.148.17
F.ROOT-SERVERS.NET      internet address = 192.5.5.241
G.ROOT-SERVERS.NET      internet address = 192.112.36.4
```

Это дает нам информацию о том, что все `ROOT-SERVERS.NET` обслуживают `edu.`, так что мы можем продолжать опрашивать сервер `C`. Теперь мы хотим знать кто обслуживает следующий уровень имени домена: `mit.edu.`:

```
> mit.edu.
Server: c.root-servers.net
Address: 192.33.4.12

Non-authoritative answer:
mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu
mit.edu nameserver = STRAWB.mit.edu
```

```

Authoritative answers can be found from:
W20NS.mit.edu   internet address = 18.70.0.160
BITSY.mit.edu   internet address = 18.72.0.3
STRAWB.mit.edu  internet address = 18.71.0.151

```

Сервера `steawb`, `w20ns` и `bitsy` все обслуживают `mit`, выберите один из них и запросите у него информацию о `ai.mit.edu`:

```
> server W20NS.mit.edu.
```

Имена машин не зависят от регистра, но я просто использую мышь для вырезания и вставки текста, так что я получаю их такими же, как они написаны на экране.

```

Server:  W20NS.mit.edu
Address: 18.70.0.160

```

```

> ai.mit.edu.
Server:  W20NS.mit.edu
Address: 18.70.0.160

```

```

Non-authoritative answer:
ai.mit.edu      nameserver = ALPHA-BITS.AI.MIT.EDU
ai.mit.edu      nameserver = GRAPE-NUTS.AI.MIT.EDU
ai.mit.edu      nameserver = TRIX.AI.MIT.EDU
ai.mit.edu      nameserver = MUESLI.AI.MIT.EDU
ai.mit.edu      nameserver = LIFE.AI.MIT.EDU
ai.mit.edu      nameserver = BEET-CHEX.AI.MIT.EDU
ai.mit.edu      nameserver = MINI-WHEATS.AI.MIT.EDU
ai.mit.edu      nameserver = COUNT-CHOCULA.AI.MIT.EDU
ai.mit.edu      nameserver = MINTAKA.LCS.MIT.EDU

```

```

Authoritative answers can be found from:
AI.MIT.EDU      nameserver = ALPHA-BITS.AI.MIT.EDU
AI.MIT.EDU      nameserver = GRAPE-NUTS.AI.MIT.EDU
AI.MIT.EDU      nameserver = TRIX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MUESLI.AI.MIT.EDU
AI.MIT.EDU      nameserver = LIFE.AI.MIT.EDU
AI.MIT.EDU      nameserver = BEET-CHEX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MINI-WHEATS.AI.MIT.EDU
AI.MIT.EDU      nameserver = COUNT-CHOCULA.AI.MIT.EDU
AI.MIT.EDU      nameserver = MINTAKA.LCS.MIT.EDU
ALPHA-BITS.AI.MIT.EDU internet address = 128.52.32.5
GRAPE-NUTS.AI.MIT.EDU internet address = 128.52.36.4
TRIX.AI.MIT.EDU  internet address = 128.52.37.6
MUESLI.AI.MIT.EDU internet address = 128.52.39.7
LIFE.AI.MIT.EDU  internet address = 128.52.32.80
BEET-CHEX.AI.MIT.EDU internet address = 128.52.32.22
MINI-WHEATS.AI.MIT.EDU internet address = 128.52.54.11
COUNT-CHOCULA.AI.MIT.EDU internet address = 128.52.38.22
MINTAKA.LCS.MIT.EDU internet address = 18.26.0.36

```

Так что `museli.ai.mit.edu` является сервером имен для `ai.mit.edu`:

```

> server MUESLI.AI.MIT.EDU
Default Server:  MUESLI.AI.MIT.EDU
Address: 128.52.39.7

```

Теперь я изменил тип запроса, поскольку мы нашли сервер имен и можем опрашивать его том, что мы хотим знать о `prep.ai.mit.edu`.

```

> set q=any
> prep.ai.mit.edu.
Server: MUESLI.AI.MIT.EDU
Address: 128.52.39.7

prep.ai.mit.edu CPU = dec/decstation-5000.25    OS = unix
prep.ai.mit.edu
    inet address = 18.159.0.42, protocol = tcp
    ftp telnet smtp finger
prep.ai.mit.edu preference = 1, mail exchanger = gnu-life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = beet-chex.ai.mit.edu
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = mini-wheats.ai.mit.edu
ai.mit.edu      nameserver = trix.ai.mit.edu
ai.mit.edu      nameserver = muesli.ai.mit.edu
ai.mit.edu      nameserver = count-chocula.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
ai.mit.edu      nameserver = life.ai.mit.edu
gnu-life.ai.mit.edu    internet address = 128.52.32.60
beet-chex.ai.mit.edu  internet address = 128.52.32.22
alpha-bits.ai.mit.edu internet address = 128.52.32.5
mini-wheats.ai.mit.edu internet address = 128.52.54.11
trix.ai.mit.edu internet address = 128.52.37.6
muesli.ai.mit.edu    internet address = 128.52.39.7
count-chocula.ai.mit.edu    internet address = 128.52.38.22
mintaka.lcs.mit.edu    internet address = 18.26.0.36
life.ai.mit.edu internet address = 128.52.32.80

```

Так начиная от корня `.`, мы нашли нужные сервера имен для каждого уровня, указанных в имени домена. Если бы вы использовали ваш собственный сервер DNS вместо использования всех перечисленных серверов, то ваш сервер конечно же кешировал бы всю информацию, которую он находил во время прохождения этого пути, и тогда бы он больше не запрашивал эту информацию при повторном обращении.

В аналогии с деревом каждый знак `"."` в имени является точкой ответвления дерева. И каждая часть между знаками `"."` является именем индивидуальной ветви в дереве.

Мы поднимаемся по дереву взяв нужное имя (`prep.ai.mit.edu`), потом находим корень дерева (`.`) и затем ищем следующую ветвь по которой будем подниматься, в нашем случае это `edu`. Однажды найдя эту ветвь, мы будем подниматься по ней, переключившись на сервер, который знает об этой части имени. Затем мы ищем ветвь `mit` в ветви `edu` (объединенное имя становится равным `mit.edu`) и поднимаемся по ней, переключившись на сервер, который знает о `mit.edu`. Далее мы ищем следующую ветвь — `ai.mit.edu` и переключаемся на сервер, который знает об этом домене. Теперь мы достигли нужного сервера в нужной точке ветвления. Последней задачей будет нахождение `prep.ai.mit.edu`, что является простой задачей. В научной литературе мы обычно называем машину `prep` *листом* дерева.

Менее обсуждаемым, но все равно важным является домен `in-addr.arpa`. Он также имеет иерархическую структуру, подобно 'обычным' доменам. `in-addr.arpa` позволяет нам получить имя машины, по ее адресу. Необходимо заметить, то что `ip`-адреса в домене `in-addr.arpa` записаны в обратном порядке. Например, Если у вас есть машина с адресом: `192.128.52.43`, то `named` обрабатывает ее точно также как в примере для `prep.ai.mit.edu`: находит сервера `arpa..` Находит сервера для домена `in-addr.arpa.`, находит сервера `192.in-addr.arpa.`, находит сервера для `128.192.in-addr.arpa.`, сервера для `52.128.192.in-addr.arpa.`. А потом уже находит необходимые записи о `43.52.128.192.in-addr.arpa`. Ловко? (скажите 'да'). Хотя обратный порядок может смущать, но это только первые два года.

Я вам солгал. DNS не работает точно так, как я это описал. Но достаточно близко к описанному процессу.

## 4.2 Наш собственный домен

Теперь определим наш собственный домен. Мы будем делать домен `linux.bogus` и определим машины в нем. Я использую полностью поддельное имя домена, для того чтобы быть уверенным, что мы не побеспокоим никого во внешнем мире.

Одно важное замечание до того как мы начнем: Не все символы разрешено использовать в именах машин. Мы ограничимся символами английского алфавита: a-z, цифрами: 0-9 и символом '-' (тире). Придерживайтесь использования этих символов. Прописные и строчные символы не различаются DNS, так что `pat.uio.no` является равным `Pat.UiO.No`.

Мы уже начали эту часть строкой в `named.conf`:

---

```
zone "0.0.127.in-addr.arpa" {
    type master;
    file "pz/127.0.0";
};
```

---

Заметьте отсутствие '.' в конце имен доменов в этом файле. Это указывает, что мы сейчас будем определять зону `0.0.127.in-addr.arpa`, и что мы будем основным сервером для нее, а также то, что она хранится в файле, названном `pz/127.0.0`. Мы уже сделали этот файл, в нем записано:

---

```
@           IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                1          ; Serial
                8H        ; Refresh
                2H        ; Retry
                1W        ; Expire
                1D)       ; Minimum TTL
                NS        ns.linux.bogus.
1           PTR      localhost.
```

---

Заметьте наличие символа '.' в конце полных имен доменов в этом файле, в противоположность вышеприведенному файлу `named.conf`. Некоторые люди любят начинать каждый файл зон с директивы `$ORIGIN`, но это является излишним. Расположение (origin) (место зоны в иерархии DNS) файла зоны указывается в разделе зон в файле `named.conf`, в данном случае это `0.0.127.in-addr.arpa`. Этот 'файл зоны' содержит 3 'записи ресурсов (resource records)' (RRs): A SOA RR, A NS RR и PTR RR. SOA это сокращение для Начала Полномочий (Start Of Authority). Символ '@' это специальный символ обозначающий расположение, и поскольку в колонке 'домен (domain)' для этого файла записано `0.0.127.in-addr.arpa`, то первая строка на самом деле значит

```
0.0.127.in-addr.arpa.  IN      SOA ...
```

NS это RR для сервера имен (Name Server). В начале строки символ '@' не указывается, это подразумевается, поскольку предыдущая строка начиналась с символа '@'. Это экономит нам несколько нажатий на клавиши. Так что строка NS в действительности читается как

```
0.0.127.in-addr.arpa.  IN      NS      ns.linux.bogus
```

Эта строка сообщает DNS, что машина является сервером имен домена `0.0.127.in-addr.arpa`, это `ns.linux.bogus`. 'ns' традиционное имя для серверов имен, но как и для web-серверов, чьим традиционным именем является `www.что-нибудь`, данное имя может быть любым.

И в в окончание - запись PTR гласит, что машина с адресом 1 в подсети 0.0.127.in-addr.arpa, например, 127.0.0.1 называется localhost.

Запись SOA находится в преамбуле каждого из файлов зон. Она описывает зону — откуда она появляется (машина, названная ns.linux.bogus), кто отвечает за содержимое зоны (hostmaster@linux.bogus, вы должны вставить здесь свой адрес электронной почты), какая версия файла зоны текущая (serial: 1), и другие вещи, которые надо сделать для кеширующих и вторичных серверов DNS. Для полей refresh, retry, expire и minimum используйте числа приведенные в этом документе и вы должны быть в безопасности, используя их.

Затем перезапустите ваш named (команда `ndc restart`) и используйте программу `nslookup` для проверки того, что сделано:

```
$ nslookup

Default Server: localhost
Address: 127.0.0.1

> 127.0.0.1
Server: localhost
Address: 127.0.0.1

Name: localhost
Address: 127.0.0.1
```

мы видим, что named работает и можно получить данные о localhost из домена 127.0.0.1, это очень хорошо. Теперь приступим к нашей основной задаче, домену linux.bogus, вставим новый раздел '(zone)' в файл named.conf:

---

```
zone "linux.bogus" {
    notify no;
    type master;
    file "pz/linux.bogus";
};
```

---

Заметим, что мы продолжаем опускать завершающий символ '.' в имени домена в файле named.conf.

В файле зоны linux.bogus мы поместим некоторые поддельные данные:

---

```
;
; Файл зоны для linux.bogus
;
; Полный файл зоны
;
@      IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                                199802151      ; serial, todays date + todays serial #
                                8H              ; refresh, seconds
                                2H              ; retry, seconds
                                1W              ; expire, seconds
                                1D )            ; minimum, seconds
;
                                NS              ns              ; Internet адрес сервера имен
                                MX              10 mail.linux.bogus      ; Основной почтовый сервер
                                MX              20 mail.friend.bogus.   ; Дополнительный почтовый сервер
;
```

---

```
localhost      A      127.0.0.1
ns              A      192.168.196.2
mail           A      192.168.196.4
```

---

Необходимо упомянуть две вещи о записи SOA. `ns.linux.bogus` *должен* быть настоящей машиной с записью A. Не разрешается указывать машину с записью CNAME в записи SOA. Это имя не обязательно должно быть 'ns', оно может быть любым правильным именем машины. Далее, `hostmaster.linux.bogus` должен читаться как `hostmaster@linux.bogus`, это должен быть почтовый псевдоним или почтовый ящик для человека сопровождающего DNS и читающего почту достаточно часто. Любая почта, относительно домена будет посылаться на адрес указанный здесь. Имя не обязательно должно быть 'hostmaster', это может быть любой правильный адрес электронной почты, но адрес с именем 'hostmaster' как ожидается будет работать.

В этом файле приведен еще один новый тип записи о ресурсах (RR) — MX, или запись ресурса Почтовый Сервер (Mail eXchanger). Она сообщает почтовой системе куда посылать почту адресованную `someone@linux.bogus`, а именно серверам `mail.linux.bogus` или `mail.friend.bogus`. Число перед каждым именем машины — это приоритет записи MX RR. Запись ресурса с наименьшим номером (10) — это машины куда почта должна посылаться, если это возможно. Если происходит ошибка, то почта может быть послана на машину с большим номером, вторичному почтовому серверу, например, `mail.friend.bogus` для которого приоритет установлен равным 20.

Перезапустите `named` с помощью команды `ndc restart`. Проверьте результаты работы используя команду `nslookup`:

```
$ nslookup
> set q=any
> linux.bogus
Server: localhost
Address: 127.0.0.1

linux.bogus
  origin = ns.linux.bogus
  mail addr = hostmaster.linux.bogus
  serial = 199802151
  refresh = 28800 (8 hours)
  retry = 7200 (2 hours)
  expire = 604800 (7 days)
  minimum ttl = 86400 (1 day)
linux.bogus      nameserver = ns.linux.bogus
linux.bogus      preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
linux.bogus      preference = 20, mail exchanger = mail.friend.bogus
linux.bogus      nameserver = ns.linux.bogus
ns.linux.bogus   internet address = 192.168.196.2
mail.linux.bogus internet address = 192.168.196.4
```

При внимательном тестировании вы обнаружите ошибку. Строка

```
linux.bogus      preference = 10, mail exchanger = mail.linux.bogus.linux.bogus
```

является полностью неправильной. Она должна выглядеть следующим образом

```
linux.bogus      preference = 10, mail exchanger = mail.linux.bogus
```

Я сознательно сделал ошибку, чтобы вы смогли получить некоторый опыт —:) Глядя в файл зоны мы обнаружим, что в строке

```
MX      10 mail.linux.bogus      ; Основной почтовый сервер
```

отсутствует точка. Или лишний раз написано 'linux.bogus'. Если имя машины не заканчивается на символ точки в файле зоны, то к концу этого имени добавляется текущее расположение (origin), вызывая в итоге дублирование текста linux.bogus.linux.bogus. Так запись

---

```
MX      10 mail.linux.bogus.      ; Основной почтовый сервер
```

---

или

---

```
MX      10 mail                    ; Основной почтовый сервер
```

---

является правильной. Я предпочитаю последнюю форму, поскольку надо меньше набирать на клавиатуре. Существуют пользователи bind, которые не согласны с этим подходом, но есть и те, которые согласны с этим. В файле зоны имя домена должно быть написано и заканчиваться на символ '.' или домен не должен быть указан, в этом случае по умолчанию доменом будет текущее расположение (origin) машины.

Я должен подчеркнуть, что в файле named.conf *не должно* быть символа '.' после имен доменов. У вас может не быть понятия про символ '.' — это слишком часто или наоборот слишком редко заполняет разные вещи и смущает много людей.

Так что опираясь на мою точку зрения мы напишем новый файл зоны, с некоторой дополнительной информацией.

---

```
;
; Файл зоны для linux.bogus
;
; Полный файл зоны
;
@      IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                        199802151      ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        1W              ; expire, seconds
                        1D )            ; minimum, seconds
;
                        TXT      "Linux.Bogus, your DNS consultants"
                        NS       ns              ; Internet адрес сервера имен
                        NS       ns.friend.bogus.
                        MX       10 mail.linux.bogus      ; Основной почтовый сервер
                        MX       20 mail.friend.bogus.    ; Дополнительный почтовый сервер

localhost      A       127.0.0.1

gw
              A       192.168.196.1
              HINFO   "Cisco" "IOS"
              TXT     "The router"

ns
              A       192.168.196.2
              MX      10 mail
              MX      20 mail.friend.bogus.
              HINFO   "Pentium" "Linux 2.0"

www
              CNAME   ns

donald
              A       192.168.196.3
```

---

```

MX      10 mail
MX      20 mail.friend.bogus.
HINFO   "i486" "Linux 2.0"
TXT     "DEK"

mail    A      192.168.196.4
        MX    10 mail
        MX    20 mail.friend.bogus.
        HINFO "386sx" "Linux 1.2"

ftp     A      192.168.196.5
        MX    10 mail
        MX    20 mail.friend.bogus.
        HINFO "P6" "Linux 2.1.86"

```

---

Здесь присутствует несколько новых записей о ресурсах (RR): запись HINFO (Информация о машине, Host INFOrmation) имеет две части, хорошей привычкой является заключение каждой из этих частей в кавычки. Первая часть — это информация об оборудовании машины, а вторая часть описывает программное обеспечение и операционную систему данной машины. Машина, названная 'ns', имеет процессор Pentium и работает под управлением Linux 2.0. CNAME (Каноническое имя, Canonical NAME) — это способ присвоить каждой машине несколько имен. Так, что www является алиасом для ns.

Использование записи CNAME является немного неоднозначным. Но безопасным способом будет следовать правилу, что записи MX, CNAME или SOA *никогда* не должны ссылаться на имя, указанное как запись CNAME, они должны ссылаться на имя определенное записью A, так что будет неправильно записать

---

```
foobar    CNAME    www                ; NO!
```

---

но вместо этого необходимо записать

---

```
foobar    CNAME    ns                ; Yes!
```

---

Также лучше считать, что запись CNAME не является настоящим именем машины для использования в адресе электронной почты: адрес `webmaster@www.linux.bogus` является неправильным адресом электронной почты. Вы можете ожидать, что некоторые администраторы электронной почты во внешнем мире следят за соблюдением этого правила, даже если у вас все работает нормально. Для того, чтобы избежать этого используйте запись A (и возможно также некоторые другие записи, такие как MX) вместо:

---

```
www      A      192.168.196.2
```

---

Некоторые из разработчиков архитектуры bind (arch-bind-wizards), рекомендуют *не* использовать запись CNAME. Так, что ее использование надо рассматривать серьезно.

Но как вы видите, этот документ также как и множество других серверов не следуют этому правилу. Загрузите новую базу данных выполнив команду `ndc reload`, это заставит named перечитать файлы зон заново.

```

$ nslookup
Default Server: localhost
Address: 127.0.0.1

> ls -d linux.bogus

```

Это означает, что должны быть перечислены все записи в данном домене. В результате получится следующее:

```
[localhost]
$ORIGIN linux.bogus.

                                1D IN NS      ns
                                1D IN NS      ns.friend.bogus.
                                1D IN TXT     "Linux.Bogus, your DNS consultants"
                                1D IN MX     10 mail
                                1D IN MX     20 mail.friend.bogus.
gw                                1D IN A     192.168.196.1
                                1D IN HINFO  "Cisco" "IOS"
                                1D IN TXT     "The router"
mail                              1D IN A     192.168.196.4
                                1D IN MX     10 mail
                                1D IN MX     20 mail.friend.bogus.
                                1D IN HINFO  "386sx" "Linux 1.0.9"
localhost                         1D IN A     127.0.0.1
www                                1D IN CNAME ns
donald                            1D IN A     192.168.196.3
                                1D IN MX     10 mail
                                1D IN MX     20 mail.friend.bogus.
                                1D IN HINFO  "i486" "Linux 1.2"
                                1D IN TXT     "DEK"
ftp                                1D IN A     192.168.196.5
                                1D IN MX     10 mail
                                1D IN MX     20 mail.friend.bogus.
                                1D IN HINFO  "P6" "Linux 1.3.59"
ns                                 1D IN A     192.168.196.2
                                1D IN MX     10 mail
                                1D IN MX     20 mail.friend.bogus.
                                1D IN HINFO  "Pentium" "Linux 1.2"
@                                  1D IN SOA   ns hostmaster (
                                        199802151      ; serial
                                        8H             ; refresh
                                        2H             ; retry
                                        1W             ; expiry
                                        1D )           ; minimum
```

Это хорошо. Как вы видите, это выглядит почти как сам файл зоны. Теперь проверим какой будет ответ только для машины с именем www:

```
> set q=any
> www.linux.bogus.
Server: localhost
Address: 127.0.0.1

www.linux.bogus canonical name = ns.linux.bogus
linux.bogus     nameserver = ns.linux.bogus
linux.bogus     nameserver = ns.friend.bogus
ns.linux.bogus  internet address = 192.168.196.2
```

Другими словами, настоящим именем для www.linux.bogus является ns.linux.bogus, и он также дает некоторую дополнительную информацию о машине ns, достаточная, чтобы соединиться с ней. Теперь мы находимся на середине пути.

### 4.3 Обратная (reverse) зона

Теперь программы могут преобразовывать имена машин в домене linux.bogus в адреса, по которым они могут связаться с этими машинами. Но также, кроме этого, требуется обратная зона, которая дает возможность DNS преобразовывать адреса в имена машин. Эти имена используются достаточным количеством серверов различного рода (FTP, IRC, WWW и другими) для того, чтобы решить, хотят ли они с вами общаться или нет, и даже иногда имя машины используется для того, чтобы решить какой приоритет вам дать. Обратная зона требуется для полного доступа к различным сервисам в Internet.

Поместите следующие строки в файл named.conf:

---

```
zone "192.168.192.in-addr.arpa" {
    notify no;
    type master;
    file "pz/192.168.196";
};
```

---

Эти строки похожи на описание зоны 0.0.127.in-addr.arpa, а файл зоны также имеет сходное содержание:

---

```
@      IN      SOA      ns.linux.bogus. hostmaster.linux.bogus. (
                                199802151 ; Serial, todays date + todays serial
                                8H      ; Refresh
                                2H      ; Retry
                                1W      ; Expire
                                1D)     ; Minimum TTL
                                NS      ns.linux.bogus.

1      PTR      gw.linux.bogus.
2      PTR      ns.linux.bogus.
3      PTR      donald.linux.bogus.
4      PTR      mail.linux.bogus.
5      PTR      ftp.linux.bogus.
```

---

Теперь перезапустите ваш named (ndc restart) и снова проверьте его работу с помощью программы nslookup:

---

```
> 192.168.196.4
Server: localhost
Address: 127.0.0.1
```

```
Name: mail.linux.bogus
Address: 192.168.196.4
```

---

так, это выглядит нормально, теперь выдайте полный список машин в домене, для того чтобы проверить правильность информации:

---

```
> ls -d 192.168.192.in-addr.arpa
[localhost]
$ORIGIN 192.168.192.in-addr.arpa.
@              1D IN SOA      ns.linux.bogus. hostmaster.linux.bogus. (
```

```

199802151      ; serial
8H             ; refresh
2H             ; retry
1W             ; expiry
1D )           ; minimum

1D IN NS      ns.linux.bogus.
1 1D IN PTR   gw.linux.bogus.
2 1D IN PTR   ns.linux.bogus.
3 1D IN PTR   donald.linux.bogus.
4 1D IN PTR   mail.linux.bogus.
5 1D IN PTR   ftp.linux.bogus.
@ 1D IN SOA   ns.linux.bogus. hostmaster.linux.bogus. (
199802151      ; serial
8H             ; refresh
2H             ; retry
1W             ; expiry
1D )           ; minimum

```

Выглядит хорошо! Если вывод вашей команды выглядит не так, то посмотрите сообщения об ошибках в вашем syslog, я объяснял как это сделать в самом начале главы.

#### 4.4 Предупреждение

Существует несколько вещей, которые я должен добавить здесь. Сетевые адреса, используемые здесь взяты из одного из блоков 'личных (private) сетей', их не разрешается публично использовать в internet. Так, что их можно спокойно использовать как пример в этом документе. Вторая вещь это строка `notify no;`. Она заставляет named не извещать вторичные (ведомые) сервера, когда вы обновляете один из файлов зон. В bind-8, named может уведомлять другие сервера, перечисленные в записях NS в файлах зон, когда конкретная зона обновляется. Это является удобным для обычного использования, но для личных экспериментов с зонами эта возможность должна быть отключена, мы же не хотим, чтобы наш эксперимент загрязнял internet?

И конечно, этот домен является фиктивным, также как и все адреса в нем. Для настоящего примера действующего домена смотрите следующий раздел.

#### 4.5 Почему не работает поиск в обратной зоне

Существует набор "gotchas", которые обычно избегаются в процессе поиска имен, что часто появляются при установке обратных зон. До того как вы продолжите продвигаться вперед вам необходимо обеспечить работу обратного поиска ваших машин на вашем сервере имен. Если эта часть не работает, то вернитесь назад и исправьте это до того, как продолжите читать.

Я буду обсуждать два сбоя обратного поиска так, как они видятся извне вашей сети:

##### 4.5.1 Обратная зона не делегируется

Когда вы просите вашего сетевого провайдера о предоставлении диапазона сетевых адресов и имени домена, то имя домена обычно делегируется. Делегирование является клеем, который связывает записи NS, которые помогают вам переходить от одного сервера имен к другому, как это описывалось в теоретическом разделе. Вы прочитали его, не так ли? Если ваша обратная зона не работает, то вернитесь назад и прочитайте нужный раздел. Прямо сейчас.

Обратную зону также надо делегировать. Если вы получили от вашего провайдера сеть 192.168.196 с доменом linux.bogus, то вам необходимо поместить запись NS в вашу обратную зону, точно так же как и для прямой зоны. Если вы проследуете по цепочке с `in-addr.arpa`

и поднимитесь до своей сети то вы скорее всего обнаружите разорванную цепочку. Скорее всего в районе вашего сетевого провайдера. Обнаружив разрыв цепочки свяжитесь с вашим сетевым провайдером и попросите его исправить ошибку.

#### 4.5.2 Вы получили бесклассовую сеть.

Это может показаться сложной темой, но бесклассовые подсети в настоящее время являются часто используемыми и у вас вероятно есть такая сеть, если вы не являетесь компанией средних размеров.

Бесклассовая сеть — это то, что сохраняет работу Internet в наши дни. Несколько лет назад было много разговоров о том, что ip-номера слишком коротки. Умные люди в IETF (Internet Engineering Task Force, они сохраняют Internet в рабочем состоянии) объединились вместе и решили данную проблему. За цену. Цена заключается в том, что вы получаете подсеть меньше подсети класса "C" и некоторые вещи могут не работать. Пожалуйста посмотрите страницу *Ask Mr. DNS at <http://www.acmebw.com/askmrdns/00007.htm>* для подробного объяснения об этом и как с этим работать.

Вы прочитали эту информацию? Я ничего не буду объяснять здесь, так что пожалуйста прочитайте. Первая часть проблемы заключается в том, что ваш провайдер должен понимать технологию, описанную Mr. DNS. Не все маленькие провайдеры имеют работающее решение данной проблемы. Если это так, то вы должны объяснить это провайдеру и быть настойчивы. Но сначала убедитесь, что вы сами это понимаете ;-). Они смогут установить правильную обратную зону на своем сервере, который вы должны проверить на работоспособность используя команду nslookup.

Вторая и последняя часть этой проблемы заключается в том, что вы должны понять применяемую технологию. Если вы не уверены в этом, то вернитесь назад и заново прочитайте все материалы. Затем вы сможете настроить обратную зону своей собственной бесклассовой сети, как это описано Mr. DNS.

Существует другая возможность для скрытой ошибки. Старые программы разрешения имен не смогут следовать приему с CNAME в цепочке разрешения имен и будут давать сбой при выяснении имени вашей машины. В результате этого различные сервисы будут относить вас к классу неправильного доступа, запрещать доступ или делать что-то подобное. Если вы столкнулись с таким сервисом, то решение (которое я знаю) заключается только в том, чтобы ваш провайдер вставил вашу запись PTR прямо в фиктивный файл бесклассовой зоны вместо фиктивной записи CNAME. Некоторые провайдеры будут предлагать другие способы решения этой проблемы, подобные основанным на Web формам ввода ваших записей обратной зоны или другие автоматические системы.

## 5 Пример реального домена

### Мы здесь перечислим несколько настоящих файлов зон

Пользователи предложили, чтобы я включил пример настроек действующего домена как учебное пособие.

Я использую этот пример с разрешения David Bullock из LAND-5. Эти файлы содержат информацию, которая являлась реальной на 24 Сентября 1996 года, и были отредактированы мною для соответствия правилам синтаксиса bind-8 и использовали некоторые мои расширения. Так что, то что вы увидите здесь будет отличаться от того, что вы увидите сделав запрос на настоящий сервер имен LAND-5.

### 5.1 /etc/named.conf (или /var/named/named.conf)

Здесь мы обнаружим два раздела основных зон для двух необходимых обратных зон: для сети 127.0.0, а также для домена LAND-5 с номером 206.6.177. А записи для основной зоны домена land-5.com. Также заметим, что вместо размещения файлов в директории названной rz, как я делал здесь до этого, администратор поместил эти файлы в директорию названную zone.

---

```
// Загрузочный файл для сервера имен LAND-5

options {
    directory "/var/named";
};

zone "." {
    type hint;
    file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
    type master;
    file "zone/127.0.0";
};

zone "land-5.com" {
    type master;
    file "zone/land-5.com";
};

zone "177.6.206.in-addr.arpa" {
    type master;
    file "zone/206.6.177";
};
```

---

Если вы поместите эти данные в ваш файл `named.conf` для того, чтобы поэкспериментировать с ними, то **ПОЖАЛУЙСТА** поместите строку `notify no;` в раздел зон, принадлежащих `land-5`, для того чтобы избежать инцидентов.

## 5.2 /var/named/root.hints

Запомните, что это файл не является постоянным, и данные одного из перечисленных здесь серверов являются устаревшими. Вам лучше использовать вместо приведенного файла, файл сделанный перед этим помощью программы `dig`, как это объяснялось ранее.

---

```
; <<>> DiG 8.1 <<>> @A.ROOT-SERVERS.NET.
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr aa rd; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 13
;; QUERY SECTION:
;;      ., type = NS, class = IN

;; ANSWER SECTION:
.          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.

;; ADDITIONAL SECTION:
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

;; Total query time: 215 msec
;; FROM: roke.uio.no to SERVER: A.ROOT-SERVERS.NET. 198.41.0.4
;; WHEN: Sun Feb 15 01:22:51 1998
;; MSG SIZE sent: 17 rcvd: 436

```

### 5.3 /var/named/zone/127.0.0

Как основа файла обязательными записями являются запись SOA, и запись, которая объявляет 127.0.0.1 как localhost. Требуется указать обе эти записи. Больше ничего не должно быть в этом файле. Его скорее всего никогда не надо будет обновлять, до тех пор пока не изменится адрес сервера имен или ответственного за машину (hostmaster).

```

@           IN          SOA      land-5.com. root.land-5.com. (
                                199609203      ; Serial
                                28800      ; Refresh
                                7200      ; Retry
                                604800      ; Expire
                                86400) ; Minimum TTL
                                NS       land-5.com.

1           PTR         localhost.

```

### 5.4 /var/named/zone/land-5.com

Здесь мы увидим обязательную запись SOA, и необходимые записи NS. Мы можем видеть, что имеется дополнительный сервер имен расположенный по адресу ns2.psi.net. *Vserga* необходимо иметь дополнительный сервер имен за пределами домена в качестве резерва. Мы также можем видеть, что этот домен имеет основной сервер, названный land-5, который заботится о множестве

разных сервисов Internet, это сделано используя записи CNAME (как альтернатива использованию записей A).

Как вы видите из записи SOA, файл зоны расположен в домене land-5.com, ответственным лицом является root@land-5.com. hostmaster — это другой часто используемый адрес для ответственного за эту работу человека. Серийный номер записан в привычном формате ууууммдд и дополнен серийным номером для текущего дня; это примерно 6-я версия файла зоны на 20 сентября 1996. Помните, что серийный номер *должен* увеличиваться монотонно, здесь только *одна* цифра для серийного номера текущего дня, так что после 9 поправок мы должны ждать завтрашнего дня, для того чтобы дальше продолжить редактировать файл. Рассмотрите возможность использования двух цифр для номера вместо одной.

---

```

@      IN      SOA      land-5.com. root.land-5.com. (
199609206      ; serial, todays date + todays serial #
8H            ; refresh, seconds
2H            ; retry, seconds
1W            ; expire, seconds
1D )          ; minimum, seconds
      NS      land-5.com.
      NS      ns2.psi.net.
      MX      10 land-5.com. ; Основной почтовый сервер

localhost    A      127.0.0.1

router       A      206.6.177.1

land-5.com.  A      206.6.177.2
ns           A      206.6.177.3
www         A      207.159.141.192

ftp          CNAME   land-5.com.
mail         CNAME   land-5.com.
news        CNAME   land-5.com.

funn        A      206.6.177.2
@           TXT     "LAND-5 Corporation"

;
;      Рабочие станции
;
ws-177200   A      206.6.177.200
            MX      10 land-5.com. ; Основная почтовая машина
ws-177201   A      206.6.177.201
            MX      10 land-5.com. ; Основная почтовая машина
ws-177202   A      206.6.177.202
            MX      10 land-5.com. ; Основная почтовая машина
ws-177203   A      206.6.177.203
            MX      10 land-5.com. ; Основная почтовая машина
ws-177204   A      206.6.177.204
            MX      10 land-5.com. ; Основная почтовая машина
ws-177205   A      206.6.177.205
            MX      10 land-5.com. ; Основная почтовая машина
; {много повторяющихся определений удалено}
ws-177250   A      206.6.177.250

```

	MX	10	land-5.com.	; Основная почтовая машина
ws-177251	A		206.6.177.251	
	MX	10	land-5.com.	; Основная почтовая машина
ws-177252	A		206.6.177.252	
	MX	10	land-5.com.	; Основная почтовая машина
ws-177253	A		206.6.177.253	
	MX	10	land-5.com.	; Основная почтовая машина
ws-177254	A		206.6.177.254	
	MX	10	land-5.com.	; Основная почтовая машина

Если вы поработаете с сервером имен домена land-5, то вы обнаружите, что имена машин имеют форму `ws_номер`. С последних версий bind 4 `named` начал ограничивать то, какие символы могут быть использованы в именах машин. Так, что эти имена нельзя было использовать с bind-8, и я подставил символ '-' (тире) вместо символа '\_' (подчеркивание), для того, чтобы этот пример можно было привести в документе.

Другая вещь, которую необходимо заметить, это то, что все рабочие станции не имеют индивидуальных имен, а вместо этого состоят из префикса за которым следует 2 последних числа из IP-адреса. Используя такое соглашение вы можете значительно упростить работу по сопровождению, но это может быть достаточно безлично и в действительности может быть источником недовольства со стороны ваших клиентов.

Мы также видим, что имя `funn.land-5.com` это алиас для `land-5.com`, но используется запись A, а не запись CNAME. Это хорошая тактика, как было упомянуто выше.

### 5.5 /var/named/zone/206.6.177

Я прокомментирую это файл после этого листинга.

```
@           IN      SOA      land-5.com. root.land-5.com. (
                                199609206      ; Serial
                                28800      ; Refresh
                                7200      ; Retry
                                604800    ; Expire
                                86400)    ; Minimum TTL
                                NS       land-5.com.
                                NS       ns2.psi.net.
;
;       Servers
;
1       PTR      router.land-5.com.
2       PTR      land-5.com.
2       PTR      funn.land-5.com.
;
;       Рабочие станции
;
200     PTR      ws-177200.land-5.com.
201     PTR      ws-177201.land-5.com.
202     PTR      ws-177202.land-5.com.
203     PTR      ws-177203.land-5.com.
204     PTR      ws-177204.land-5.com.
205     PTR      ws-177205.land-5.com.
; {много повторяющихся определений удалено}
250     PTR      ws-177250.land-5.com.
251     PTR      ws-177251.land-5.com.
```

---

252	PTR	ws-177252.land-5.com.
253	PTR	ws-177253.land-5.com.
254	PTR	ws-177254.land-5.com.

---

Обратная зона является той частью настройки, которая кажется способной вызвать большое горе (печаль, grief). Она используется для того, чтобы найти имя машины по ее IP-адресу. Пример: у вас есть IRC-сервер и он принимает соединения от IRC-клиентов. Однако он находится в Норвегии и хочет принимать соединения только от клиентов из Норвегии и других скандинавских стран. Когда вы соединяетесь с клиентом, то с помощью библиотеки языка C вы можете узнать адрес соединяющейся с сервером машины, поскольку этот адрес содержится во всех пакетах, передаваемых по сети. Теперь сервер может вызвать функцию названную `gethostbyaddr`, которая ищет имя машины по заданному IP-номеру. `gethostbyaddr` запросит сервер DNS, который выполнит поиск заданной машины в DNS. Допустим клиент соединяется с машины `ws-177200.land-5.com`. IP-номер, который библиотека C передает IRC-серверу, равен `206.6.177.200`. Для того, чтобы найти имя машины нам необходимо найти домен `200.177.6.206.in-addr.arpa`. DNS-сервер сначала найдет сервера домена `arpa.`, затем найдет сервера `in-addr.arpa.`, следуя дальше через сервера группы 206, затем 6 и в конце концов найдет сервер для зоны `177.6.206.in-addr.arpa` в домене `land-5`. От которого он в конце концов может получить ответ, что для `200.177.6.206.in-addr.arpa` у нас есть запись "PTR `ws-177200.land-5.com`", означающая что имя машины с адресом `206.6.177.200` равно `ws-177200.land-5.com`. Как и объяснение того, как мы искали `prep.ai.mit.edu`, этот пример вымышлен.

Вернемся к нашему примеру с IRC-сервером. IRC-сервер принимает соединения только из скандинавских стран, например, `*.no`, `*.se`, `*.dk`, имя `ws-177200.land-5.com` явно не соответствует этому правилу и сервер запретит соединение. Если бы не было обратного мапирования адреса `206.2.177.200` с помощью зоны `in-addr.arpa`, то сервер не смог бы вообще найти имя машины и должен был бы сравнивать адрес `206.2.177.200` с заданными масками — `*.no`, `*.se` и `*.dk`, которым этот адрес явно не соответствует.

Некоторые люди скажут, что обратное преобразование адресов важно только для серверов, и не важно для остальной работы. Это не так: многие `ftp`, `news`, IRC и даже некоторые `http` (WWW) сервера не принимают соединения от машин для которых они не могут найти имена. Так что в действительности обратное преобразование для машин является *обязательным*.

## 6 Сопровождение

### Содержание в рабочем состоянии.

Существует только одна задача по сопровождению `named`, кроме содержания его запущенным на машине. Это регулярное обновление файла `root.hints`. Наиболее легкий путь — это использование программы `dig`. Сначала запустите `dig` без аргументов и вы получите файл `root.hints`, соответствующий вашим данным. Затем запросите один из перечисленных корневых серверов, выполнив команду `dig @rootserver`. Вы заметите, что вывод этой команды выглядит ужасно похоже на файл `root.hints`. Сохраните выводимые данные в файл (с помощью команды `dig @e.root-servers.net . ns >root.hints.new`) и замените этим файлом старый файл `root.hints`.

Помните, что необходимо перезапустить `named` после замены файла.

Al Longyear послал мне этот скрипт, который может быть автоматически запущен для того, чтобы обновлять файл `root.hints`, создайте запись в `crontab` для запуска его раз в месяц и забудьте про него. В скрипте предполагается, что почтовая система работает и что определен почтовый алиас 'hostmaster'. Вы должны подправить этот скрипт для соответствия вашим настройкам.

---

```
#!/bin/sh
#
```

```
# Обновляет информацию кеша сервера имен раз в месяц.
# Он автоматически запускается раз в месяц с помощью cron.
#
# Original by Al Longyear
# Updated for bind 8 by Nicolai Langfeldt
# Miscelanious error-conditions reported by David A. Ranch
# Ping test suggested by Martin Foster
#

(
echo "To: hostmaster <hostmaster>"
echo "From: system <root>"
echo "Subject: Automatic update of the named.conf file"
echo

PATH=/sbin:/usr/sbin:/bin:/usr/bin:
export PATH
cd /var/named

# Мы подключены? проверка подключения к серверу вашего ISP
case `ping -qnc some.machine.net` in
*'100% packet loss'*)
    echo "The network is DOWN. root.hints NOT updated"
    echo
    exit 0
    ;;
esac

dig @rs.internic.net . ns >root.hints.new 2>&1

case `cat root.hints.new` in
*NOERROR*)
    # It worked
    ;;
*)
    echo "The root.hints file update has FAILED."
    echo "This is the dig output reported:"
    echo
    cat root.hints.new
    exit 0
    ;;
esac

echo "The named.conf file has been updated to contain the following information:"
echo
cat root.hints.new

chown root.root root.hints.new
chmod 444 root.hints.new
rm -f root.hints.old
mv root.hints root.hints.old
mv root.hints.new root.hints
```

```

ndc restart
echo
echo "The nameserver has been restarted to ensure that the update is complete."
echo "The previous root.hints file is now called
/var/named/root.hints.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0

```

Некоторые из вас заметили, что файл `root.hints` также доступен по ftp с сервера Internic. Пожалуйста *не* используйте ftp для обновления файла `root.hints`, вышеприведенный метод более дружелюбен по отношению к сети.

## 7 Преобразование настроек named из версии 4 в версию 8

Раньше это был раздел об использовании bind 8, написанный David E. Smith ([dave@bureau42.ml.org](mailto:dave@bureau42.ml.org)). Я немного изменил его для того, чтобы соответствовать имени раздела.

В нем совсем немного. За исключением использования файла `named.conf` вместо `named.boot`, все остальное тоже самое. И `bind8` поставляется со скриптом на perl, который преобразует файлы настроек со старым синтаксисом в файл с новым синтаксисом. Пример файла `named.boot` (старый стиль) для кеширующего сервера имен:

```

directory /var/named
cache      .                root.hints
primary 0.0.127.IN-ADDR.ARPA 127.0.0.zone
primary localhost          localhost.zone

```

В командной строке, находясь в директории `bind8/src/bin/named` (этот пример предполагает, что у вас имеется исходный код пакета `bind8`. Если у вас бинарный пакет, то скрипт вероятно должен быть где-то рядом, хотя я не уверен, в том где точно он будет находиться), наберите:

```
./named-bootconf.pl < named.boot > named.conf
```

Эта команда создаст соответствующий файл `named.conf`:

```

// generated by named-bootconf.pl

options {
    directory "/var/named";
};

zone "." {
    type hint;
    file "root.hints";
};

zone "0.0.127.IN-ADDR.ARPA" {
    type master;
    file "127.0.0.zone";
};

zone "localhost" {

```

```
    type master;
    file "localhost.zone";
};
```

---

Этот скрипт работает для всего, что могло быть записано в файле `named.boot`, хотя он не добавляет все новые расширения и опции настройки, которые разрешено использовать в `bind8`. Здесь я приведу более полный файл `named.conf`, который делает те же самые вещи, но немного более эффективно.

---

```
// Это файл конфигурации для named (для BIND 8.1 или более поздних).
// Он обычно устанавливается как /etc/named.conf.
// Отличие от `шаблонного' файла named.conf (кроме этого комментария :)
// в том, что строка <tt/directory/ разкомментирована, поскольку
// у меня уже есть файлы зон в директории /var/named.
```

```
options {
    directory "/var/named";
    datasize 20M;
};

zone "localhost" IN {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "127.0.0.zone";
};

zone "." IN {
    type hint;
    file "root.hints";
};
```

---

В директории `bind8/src/bin/named/test` исходных текстов `bind8` содержится этот файл, а также копии файлов зон, которые многие люди могут посмотреть и использовать для быстрого старта с нуля.

Форматы файлов зон и файлов `root.hints` одинаковы, также как и команды для их обновления.

## 8 Вопросы и ответы

Пожалуйста прочитайте этот раздел до того, как соберетесь писать мне о чем либо.

1. Мой `named` требует наличия файла `named.boot`

Вы прочитали неправильный HOWTO. Пожалуйста посмотрите старую версию этого документа, которая описывает `bind 4`. Ее можно найти по адресу <http://www.math.uio.no/~janl/DNS/>

2. Как использовать DNS изнутри сети, защищенной `firewall`?

Совет: вам также может понадобиться строка

---

```
query-source port 53;
```

в разделе "options" файла `named.conf`, как предполагается в примере, приведенном в разделе 3 (Кеширующий сервер).

3. Как могу сделать круговорот DNS для определенного сервиса, например `www.busy.site`, среди доступных для него адресов, для того, чтобы получить эффект баланса нагрузки, или чего-то подобного.

Сделайте несколько разных записей `A` для `www.busy.site` и используйте `bind` версии 4.9.3 или старше. При этом `bind` при ответах будет смещаться по кругу между заданными адресами. Этот метод *не* будет работать с более ранними версиями `bind`.

4. Я хочу установить DNS в закрытой корпоративной сети (intranet). Что мне надо сделать?

Вам необходимо убрать файл `root.hints` и просто использовать файлы зон. Это также означает, что вам никогда не нужно будет обновлять его.

5. Как я могу установить дополнительный (ведомый) сервер имен?

Если основной/ведущий сервер имен имеет адрес `127.0.0.1`, то вам необходимо поместить примерно такую строку в файл `named.conf` дополнительного сервера имен:

```
zone "linux.bogus" {
    type slave;
    file "sz/linux.bogus";
    masters { 127.0.0.1; };
};
```

Вы можете перечислить несколько альтернативных основных серверов для зоны, которые могут копироваться, внутри списка `masters`, разделенного знаком `';` (точка с запятой).

6. Я хочу иметь работающий `bind`, в то время, когда я не подключен к сети.

Есть три подхода к решению этой проблемы:

- Я получил это письмо от Ian Clark <ic@deakin.edu.au>, где он объяснял свой способ решения данной проблемы:

У меня запущен `named` на моей машине с 'Masquerading'. У меня есть два файла `root.hints` -- один, названный `root.hints.real`, который содержит настоящий список имен корневых серверов и второй, названный `root.hints.fake`, который содержит...

```
----
; root.hints.fake
; этот файл не содержит информации
----
```

Когда я отключаюсь от провайдера, я копирую файл `root.hints.fake` в файл `root.hints` и перезапускаю `named`.

Когда я снова подключаюсь к провайдеру, то я копирую файл `root.hints.real` в `root.hints` и опять перезапускаю `named`.

Это делается из скриптов `ip-down` и `ip-up` соответственно.

Сначала я делаю запрос об имени домена, о котором named не имеет информации, и он помещает примерно такую строку в файл messages..

```
Jan 28 20:10:11 hazchem named[10147]: No root nameserver for class IN
```

Кажется это работает в моем случае. Я могу использовать сервер имен для локальных машин в то время, когда я отключен от сети, без каких либо задержек по тайм-ауту для внешних доменов, и сервер имен работает нормально с внешними доменами, в то в время, когда я подключен к сети.

- Я также получил информацию от Karl-Max Wanger о том, как bind взаимодействует с NFS и portmapper на большинстве машин, не имеющих доступа к внешней сети.

У меня запущен named на всех машинах, которые только время от времени подключены к Internet с помощью модема. Сервер имен только кеширует информацию, он не авторизует никакую информацию и запрашивает обо всем сервера перечисленные в файле root.cache. Как обычно в Slackware, он запускается до nfsd и mountd.

На одной из моих машин (портативный компьютер Libretto 30) у меня была проблема с тем, что иногда я мог примонтировать ее диск с другой машины, подключенной к моей локальной сети, но в остальное время эта операция не удавалась. У меня был один и тот же эффект, вне зависимости от использования PLIP, PCMCIA ethernet карты или PPP по последовательному интерфейсу.

После некоторого времени, проведенного в предположениях и экспериментах, я нашел, что named несомненно вносит беспорядок в процесс регистрации nfsd и mountd, что выполняется с помощью portmapper при запуске (я как обычно запускаю эти демоны во время загрузки). Запуск named после nfsd и mountd полностью устранил эту проблему.

Поскольку не ожидается никакого ущерба от такой измененной последовательности загрузки, то я рекомендую всем сделать также как сделал я, для предотвращения потенциальных осложнений.

- В заключение, существует HOWTO об этом по адресу *Ask Mr. DNS at <http://www.acmebw.com/askmrdns/#linux-ns>*. Хотя это о bind 4, вы можете это применить и для bind 8.

7. Где кеширующий сервер имен хранит свой кеш? Есть ли способ контролировать размер кеша.

Кеш полностью хранится в памяти, он *никогда* сохраняется на на диск. Каждый раз, когда вы прекращаете выполнение named содержимое кеша теряется. Кеш не контролируется ни одним из способов. named обслуживает его, согласно нескольким простым правилам. Вы не можете контролировать кеш или его размер по любым причинам. Если вы хотите, то вы можете "исправить" это, подправив named. Однако это не рекомендуется.

8. Сохраняет ли named свой кеш между перезапусками? Как я могу заставить named сохранять его?

Нет, named *не* сохраняет кеш при завершении. Это означает, что кеш должен быть построен заново каждый раз при перезапуске named. *Нет* способа, который заставил бы named сохранять кеш в файле. Если вы хотите, то вы можете "исправить" это, подправив named. Однако это не рекомендуется.

9. Как я могу получить домен? Я хочу настроить свой домен, названный (например) `linux-rules.net`. Как можно назначить этот домен для меня?

Свяжитесь с вашим провайдером интернет. Он сможет вам помочь в этом вопросе. Заметьте, что в большинстве частей мира вам необходимо заплатить деньги за получение домена.

## 9 Как стать более знающим администратором DNS.

### Документации и утилиты.

Существует настоящая документация. Доступная в электронной и в печатной формах. Чтение некоторых из этих руководств требуется для того, чтобы сделать шаг от простого администратора DNS к крупному администратору. В печатном виде стандартной книгой является *DNS and BIND* авторов С. Liu и Р. Albitz, выпущенная О'Reilly & Associates, Sebastopol, CA, ISBN 0-937175-82-X. Я читал ее, она великолепна. Также существует раздел о DNS в книге *TCP/IP Network Administration*, написанной Craig Hunt и изданной О'Reilly..., ISBN 0-937175-82-X. Также для администрирования DNS (или всего, относящегося к этому) должна быть хорошей книга *Zen and the Art of Motorcycle Maintenance* автора Robert M. Prsig :-), доступна как ISBN 0688052304, а также другие книги.

В электронной форме вы можете найти материалы на <http://www.dns.net/dnsrd/> (Каталог ресурсов DNS), <http://www.isc.org/bind.html>; FAQ, справочные материалы (BOG; Bind Operations Guide), а также статьи и определения протоколов и дополнения (hacks) к DNS (эти, и большинство, если не все, RFC, упомянутые ниже, они также идут в поставке bind). Мне не нужны большинство из них, но я не являюсь крупным DNS администратором. Arnt Gulbrandsen читал BOG и находится в экстазе от его :-). В группе новостей *comp.protocols.tcp-ip.domains* происходят дискуссии о DNS. В добавление к этому, здесь перечислены некоторые RFC о DNS, самыми важными из которых вероятно являются следующие:

### RFC 2052

A. Gulbrandsen, P. Vixie, *A DNS RR for specifying the location of services (DNS SRV)*, October 1996

### RFC 1918

Y. Rekhter, R. Moskowitz, D. Karrenberg, G. de Groot, E. Lear, *Address Allocation for Private Internets*, 02/29/1996.

### RFC 1912

D. Barr, *Common DNS Operational and Configuration Errors*, 02/28/1996.

### RFC 1912 Errors

B. Barr *Errors in RFC 1912*, this is available at <http://www.cis.ohio-state.edu/~barr/rfc1912-errors.html>

### RFC 1713

A. Romao, *Tools for DNS debugging*, 11/03/1994.

### RFC 1712

C. Farrell, M. Schulze, S. Pleitner, D. Baldoni, *DNS Encoding of Geographical Location*, 11/01/1994.

### RFC 1183

R. Ullmann, P. Mockapetris, L. Mamakos, C. Everhart, *New DNS RR Definitions*, 10/08/1990.

### RFC 1035

P. Mockapetris, *Domain names - implementation and specification*, 11/01/1987.

### RFC 1034

P. Mockapetris, *Domain names - concepts and facilities*, 11/01/1987.

**RFC 1033**

M. Lottor, *Domain administrators operations guide*, 11/01/1987.

**RFC 1032**

M. Stahl, *Domain administrators guide*, 11/01/1987.

**RFC 974**

C. Partridge, *Mail routing and the domain system*, 01/01/1986.