

Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

Е.В. Красавин, Е.А. Черепков

НАСТРОЙКА И ИСПОЛЬЗОВАНИЕ СИСТЕМЫ DNS

Методические указания к лабораторной работе
по дисциплине «Операционные системы»

Калуга – 2019


УДК 004.62
ББК 32.972.1
К78

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий».

Методические указания рассмотрены и одобрены:


- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол № 51.4/6 от «20» февраля 2019 г.

Зав. кафедрой ИУ4-КФ

 к.т.н., доцент Ю.Е. Гагарин


- Методической комиссией факультета ИУ-КФ протокол № 9 от «04» 03 2019 г.

Председатель методической
комиссии факультета ИУ-КФ

 к.т.н., доцент М.Ю. Адкин

- Методической комиссией
КФ МГТУ им.Н.Э. Баумана протокол № 5 от «5» 03 2019 г.

Председатель методической комиссии
КФ МГТУ им.Н.Э. Баумана

 д.э.н., профессор О.Л. Перерва

Рецензент:

к.т.н., доцент кафедры ИУ6-КФ

 А.Б. Лачихина

Авторы

к.т.н., доцент кафедры ИУ4-КФ
ассистент кафедры ИУ4-КФ

 Е.В. Красавин
 Е.А. Черепков

Аннотация

Методические указания к выполнению лабораторной работы по курсу «Операционные системы» содержат подробное описание настройки DNS-клиента и DNS-сервера в операционной системе FreeBSD.

Предназначены для студентов 3-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2019 г.
© Е.В. Красавин, Е.А. Черепков, 2019 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ.....	5
КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ	6
НАСТРОЙКА DNS.....	8
ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ	32
КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ	33
ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ	33
ОСНОВНАЯ ЛИТЕРАТУРА.....	34
ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА	34

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Операционные системы» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 3-го курса направления подготовки 09.03.04 «Программная инженерия», содержат подробное описание настройки системы DNS.

Методические указания составлены для ознакомления студентов с настройкой системы DNS в операционной системе FreeBSD. Для выполнения лабораторной работы студенту необходимы минимальные знания по установке операционных систем.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является получение практических навыков работы в среде ОС FreeBSD и ее администрирования.

Основными задачами выполнения лабораторной работы являются:

1. Научиться настраивать DNS-клиент в ОС FreeBSD.
2. Научиться настраивать DNS-сервер в ОС FreeBSD.

Результатами работы являются:

1. Демонстрация настроенной системы DNS.
2. Подготовленный отчет.

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

По умолчанию во FreeBSD используется одна из версий программы BIND (Berkeley Internet Name Domain), являющейся самой распространенной реализацией протокола DNS. DNS - это протокол, при помощи которого имена преобразуются в IP-адреса и наоборот. Например, в ответ на запрос о www.FreeBSD.org будет получен IP-адрес веб-сервера Проекта FreeBSD, а запрос о ftp.FreeBSD.org возвратит IP-адрес соответствующей машины с FTP-сервером. Точно также происходит и обратный процесс. Запрос, содержащий IP-адрес машины, возвратит имя хоста. Для выполнения запросов к DNS вовсе не обязательно иметь в системе работающий сервер имён.

FreeBSD в настоящее время поставляется с сервером DNS BIND9, предоставляющим расширенные настройки безопасности, новую схему расположения файлов конфигурации и автоматические настройки для chroot.

В сети Интернете DNS управляется через достаточно сложную систему авторизованных корневых серверов имён, серверов доменов первого уровня (Top Level Domain, TLD) и других менее крупных серверов имён, которые содержат и кэшируют информацию о конкретных доменах.

На данный момент пакет BIND поддерживается Internet Software Consortium <http://www.isc.org/>.

Используемая терминология

Для изучения данного материала необходимо понимать значения некоторых терминов, связанных с работой DNS, которые указаны в таблице 1.

Таблица 1. Основные термины

Термин	Определение
Прямой запрос к DNS (forward DNS)	Преобразование имён хостов в адреса IP
Ориджин (origin)	Обозначает домен, покрываемый конкретным файлом зоны
named, bind, сервер имён	Общепотребительные названия для обозначения пакета BIND , обеспечивающего работу сервера имён во FreeBSD.
Резолвер	Системный процесс, посредством которого машина обращается к серверу имён для получения информации о зоне
Обратный DNS (reverse DNS)	Операция, обратная прямому запросу к DNS; преобразование адресов IP в имена хостов
Корневая зона	Начало иерархии зон Интернет. Все зоны находятся под корневой зоной, подобно тому, как все файлы располагаются ниже корневого каталога.
Зона	Отдельный домен, поддомен или часть DNS , управляемая одним сервером

Примеры зон:

- . является корневой зоной
- *org.* — домен верхнего уровня (TLD) в корневой зоне.
- *example.org.* является зоной в домене верхнего уровня (TLD) *org.*
- *1.168.192.in-addr.arpa* является зоной, в которую включены все IP-адреса, формирующие пространство адресов 192.168.1.*.

Как можно видеть, уточняющая часть имени хоста появляется слева. Например, *example.org.* более точен, чем *org.*, также, как *org.* более точен, чем корневая зона. Расположение каждой части имени хоста сильно похоже на файловую систему: каталог */dev* расположен в корневой файловой системе, и так далее.

НАСТРОЙКА DNS

Файл настройки ясно документирует настройки DNS-клиента. Администратор может указать до трех серверов имен, два из которых являются резервными - на случай, если не ответит первый сервер. Кроме того, файл содержит имя домена по умолчанию и прочие параметры работы. Файл `resolv.conf` - важнейшая часть настройки службы имен.

`/etc/resolv.conf` - это простой файл, подходящий для чтения людьми. Существуют некоторые вариации команд файла, зависящие от системы. Ниже перечислены записи, поддерживаемые большинством систем:

`nameserver` адрес

Записи `nameserver` указывают IP-адреса серверов, опрашиваемых DNS-клиентом на предмет получения доменной информации. Опрос серверов имен происходит в порядке следования записей `nameserver`. Если от сервера не получен ответ, DNS-клиент посылает запрос следующему серверу по списку, и так до тех пор, пока не будет достигнут последний из серверов. Если файл `resolv.conf` не существует либо в файле отсутствуют записи `nameserver`, все запросы направляются локальному узлу. Однако если файл `resolv.conf` существует и содержит записи `nameserver`, обращение к локальному узлу происходит только в том случае, если присутствует соответствующая запись `nameserver`. Указывайте официальный IP-адрес локального узла (или адрес 0.0.0.0), но не кольцевой адрес. Официальный адрес позволяет избежать проблем, возникающих в некоторых вариантах Unix при использовании кольцевого адреса. Вариант `resolv.conf` для чистого клиента DNS никогда не содержит записи `nameserver`, указывающей на локальный узел.

`domain` имя

Запись `domain` определяет доменное имя по умолчанию. DNS-клиент добавляет доменное имя по умолчанию к любому имени узла, не содержащему точки. Дополненное таким образом имя узла используется в запросе к серверу имен. Например, если DNS-клиент получает имя `crab` (не содержащее точки), он добавляет доменное имя по умолчанию в процессе конструирования запроса. Предположим, значение имени в записи `domain` - `wrotethebook.com`, тогда DNS-клиент создает запрос для `crab.wrotethebook.com`. Переменная среды `LOCALDOMAIN`, будучи установленной, имеет приоритет более высокий, чем запись `domain`, и значение переменной используется для дополнения имени узла.

`search` домен...

Запись `search` определяет ряд доменов, в которых производится поиск, если имя узла не содержит точки. Предположим, файл содержит запись `search essex.wrotethebook.com butler.wrotethebook.com`. Запрос для узла по имени `cookbook` будет преобразован сначала в запрос для имени `cookbook.essex.wrotethebook.com`. Если поиск для такого имени не принес положительных результатов, DNS-клиент создает запрос для `cookbook.butler.wrotethebook.com`. Вновь получив отрицательный результат, клиент DNS прервет процесс поиска для имени узла. Используйте запись `search` либо запись `domain`. Предпочтение отдавайте команде `search`. Никогда не используйте обе команды одновременно. Переменная среды `LOCALDOMAIN` имеет более высокий приоритет, чем запись `search`.

`sortlist` сеть [/маска сети] ...

Адреса, принадлежащие перечисленным в команде `sortlist` сетям, являются для DNS-клиента предпочтительными. Если DNS-клиент получает несколько адресов в ответ на запрос по многосетевому узлу или маршрутизатору, адреса сортируются таким образом, что адреса сетей `sortlist` предшествуют адресам всех прочих сетей. В ином случае

адреса возвращаются приложению в порядке их получения от сервера имен.

Команда `sortlist` используется редко, поскольку затрудняет работу таких серверных механизмов, как распределение нагрузки. Основным исключением является ситуация, когда список сортировки настраивается для предпочтения адресов локальной сети всем прочим адресам. В последнем случае, если клиент DNS состоит в сети 172.16.0.0/16 и один из адресов, полученных в многоадресном ответе, также принадлежит этой сети, адрес сети 172.16.0.0 будет предшествовать всем прочим адресам.

`options` параметр ...

Запись `options` позволяет устанавливать необязательные параметры настройки клиента DNS. Доступны следующие параметры:

- *debug* – включает отладку - печать отладочных сообщений на стандартный вывод/ `debug` работает только в случае, если библиотека DNS-клиента была собрана с ключом - `DDEBUG` (в большинстве случаев это не так).
- *ndots:i* – устанавливает число точек в имени узла, наличие которого служит критерием необходимости использования списка поиска перед отправкой запроса серверу имен. По умолчанию имеет значение 1. Таким образом, к имени узла с одной точкой не добавляется доменное имя перед отправкой серверу имен. Если указать параметр `ndots:2`, к имени узла с одной точкой добавляется домен из списка поиска перед отправкой запроса, но не к имени с двумя или более точками. Параметр `ndots` может пригодиться, если одну из составляющих имени домена можно спутать с доменом высшего уровня, и пользователи постоянно усекают имена из этого домена. В таком случае запросы будут передаваться для разрешения прежде всего корневым серверам имен для поиска в домене верхнего уровня, прежде чем, в конечном итоге, вернуться на локальный сервер имен. Беспokoить корневые серверы по пустякам - очень плохой тон. Используйте `ndots`, чтобы обязать DNS-клиент

принудительно дополнять проблемные имена локальным доменным именем, чтобы разрешение происходило без обращения к корневым серверам.

- *timeout:n* – устанавливает начальный интервал ожидания ответа [DNS](#)-клиентом. По умолчанию интервал ожидания равен 5 секундам для первого запроса к каждому серверу. В пакете BIND для системы Solaris 8 данный параметр имеет синтаксис *retrans: n*.
- *attempts: n* – задает число повторных попыток получить ответ на запрос. По умолчанию имеет значение 2, то есть DNS-клиент дважды повторяет попытку получить ответ для каждого из серверов в списке серверов, прежде чем вернуть приложению сообщение об ошибке. В пакете BIND для системы Solaris 8 данный параметр имеет синтаксис *retry: n* и значение по умолчанию 4.
- *rotate* – включает циклический механизм «round-robin» выбора серверов имен. В обычной ситуации DNS-клиент посылает запрос первому серверу из списка, а следующему серверу - лишь в том случае, если первый сервер не ответил на запрос. Параметр *rotate* предписывает DNS-клиенту распределить нагрузку поровну между всеми серверами имен.
- *no-check-names* – отключает проверку доменных имен на соответствие документу RFC 952, *DOD Internet Host Table Specification* (Спецификация таблицы узлов сети Интернет Министерства обороны). По умолчанию доменные имена, содержащие подчеркивание (`_`), символы не из таблицы ASCII либо управляющие символы ASCII, считаются ошибочными. Воспользуйтесь этим параметром, если существует необходимость работать с именами, содержащими подчеркивание.
- *inet6* – предписывает DNS-клиенту создавать запросы адресов IPv6.

Чаще всего файл настройки [resolv.conf](#) содержит в списке поиска локальное доменное имя, указывает локальный узел в качестве первого сервера имен, а также один или два резервных сервера имен. Пример такой настройки:

```
# Файл настройки клиента DNS
#
search wrotethebook.com
# обратиться, прежде всего, к себе
nameserver 172.16.12.2
# затем к узлу crab
nameserver 172.16.12.1
# и, наконец, к узлу ora
nameserver 172.16.1.2
```

Пример основан на воображаемой сети, поэтому по умолчанию для доменного имени указано имя `wrotethebook.com`. Файл взят с узла `rodent`, который и обозначен в качестве первого сервера имен. В качестве резервных серверов выступают `crab` и `ora`. Настройка не содержит параметров и списка сортировки, поскольку они применяются нечасто. Так выглядит файл настройки обычного DNS-клиента.

Настройка чистого DNS-клиента

Настройки чистого [DNS](#)-клиента очень просты. Они идентичны настройкам обычного клиента, но не указывают локальную систему в качестве сервера имен. Вот пример файла [resolv.conf](#) для системы чистого DNS-клиента:

```
# Файл настройки DNS-клиента
#
search wrotethebook.com
# обратиться к узлу crab
nameserver 172.16.12.1
# затем к узлу ora
nameserver 172.16.1.2
```

Данные настройки предписывают DNS-клиенту передавать все запросы узлу `crab`, а если `crab` не ответил – узлу `ora`. Ни при каких

обстоятельствах запросы не разрешаются локально. Столь простой файл настройки - все, что требуется для работы чистого DNS-клиента.

Причины, по которым вам может понадобиться сервер имён

Сервера имён обычно используются в двух видах: авторитетный сервер имён и кэширующий сервер имён.

Авторитетный сервер имён нужен, когда:

- нужно предоставлять информацию о DNS остальному миру, отвечая на запросы авторизованно.
- зарегистрирован домен, такой, как example.org и в этом домене требуется поставить имена машин в соответствие с их адресами IP.
- блоку адресов IP требуется обратные записи DNS (IP в имена хостов).
- резервный (slave) сервер имён должен отвечать на запросы.

Кэширующий сервер имён нужен, когда:

- локальный сервер DNS может кэшировать информацию и отвечать на запросы быстрее, чем это происходит при прямом опросе внешнего сервера имён.

Например, когда кто-нибудь запрашивает информацию о www.FreeBSD.org, то обычно резолвер обращается к серверу имён вашего провайдера, посылает запрос и ожидает ответа. С локальным кэширующим сервером DNS запрос во внешний мир будет делаться всего один раз. Каждый дополнительный запрос не будет посылаться за пределы локальной сети, потому что информация уже имеется в кэше.

Как это работает

Во FreeBSD даемон [BIND](#), по очевидным причинам, называется [named](#).

Таблица 2. Даемон BIND

Файл	Описание
named	Даемон BIND
mdc	Программа управления даемоном сервера имён
/etc/namedb	Каталог, в котором располагается вся информация о зонах BIND
/etc/namedb/named.conf	Конфигурационный файл для даемона

Файлы зон обычно располагаются в каталоге /etc/namedb и содержат информацию о зоне [DNS](#), за которую отвечает сервер имён.

В зависимости от способа конфигурации зоны на сервере файлы зон могут располагаться в подкаталогах master, slave или dynamic иерархии /etc/namedb. Эти файлы содержат DNS информацию, которую и будет сообщать в ответ на запросы сервер имен.

Запуск BIND

Так как сервер имён BIND устанавливается по умолчанию, его настройка сравнительно проста.

Стандартная конфигурация named запускает простой кэширующий сервер в ограниченной среде chroot. Для одноразового запуска даемона в этой конфигурации используйте команду

```
# /etc/rc.d/named forrestart
```

Чтобы даемон named запускался во время загрузки, поместите в /etc/rc.conf следующую строку:

```
named_enable="YES"
```

Разумеется, существует множество различных конфигураций `/etc/namedb/named.conf`, лежащих за рамками данного документа. Разнообразные опции запуска `named` во FreeBSD описаны в переменных `named_*` файла `/etc/defaults/rc.conf` и странице справочника `rc.conf`.

Конфигурационные файлы

Файлы конфигурации демона [named](#) расположены в каталоге `/etc/namedb` и, за исключением случая, когда вам требуется просто резолвер, требуют модификации.

Использование `make-localhost`

Для создания основной зоны для локального хоста перейдите в каталог [namedb](#)/`/etc/namedb` и выполните команду

```
# sh make-localhost
```

В каталоге `master` должны появиться файлы `localhost.rev` для локальной адресной зоны и `localhost-v6.rev` для для конфигурации IPv6. Ссылки на эти файлы уже содержатся в файле конфигурации `named.conf`.

```
/etc/namedb/named.conf
// $FreeBSD$
//
// If you are going to set up an authoritative server,
// make sure you
// understand the hairy details of how DNS works. Even
// with
// simple mistakes, you can break connectivity for
// affected parties,
// or cause huge amounts of useless Internet traffic.

options {
// All file and path names are relative to the chroot
```

```

// directory,
// if any, and should be fully qualified.
    directory          "/etc/namedb";
    pid-file            "/var/run/named/pid";
    dump-file           "/var/dump/named_dump.db";
    statistics-file     "/var/stats/named.stats";

// If named is being used only as a local resolver,
// this is a safe default. For named to be accessible
// to the network, comment this option, specify
// the proper IP address, or delete this option.

    listen-on { 127.0.0.1; };

// If you have IPv6 enabled on this system, uncomment
// this option for use as a local resolver.
// To give access to the network,
// specify an IPv6 address, or the keyword "any".
// listen-on-v6 { ::1; };

// These zones are already covered by the empty zones
// listed below. If you remove the related empty zones
// below, comment these lines out.

disable-empty-zone "255.255.255.255.IN-ADDR.ARPA";
disable-empty-zone
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
0.0.0.0.0.
IP6.ARPA";
disable-empty-zone
"1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.
0.0.0.0.0.
IP6.ARPA";

// If you've got a DNS server around at your upstream
// provider, enter its IP address here, and enable the

```



```

// line below. This will make you benefit from its
// cache, thus reduce overall DNS traffic in the
// Internet.
/*
forwarders {
127.0.0.1;
};
*/
// If the 'forwarders' clause is not empty the default
// is to
'forward first'
// which will fall back to sending a query from your
// local server if the name servers in 'forwarders'
// do not have the answer. Alternatively you can
// force your name server to never initiate queries of
// its own by enabling the following line:
// forward only;
// If you wish to have forwarding configured
automatically based on
// the entries in /etc/resolv.conf, uncomment the
// following line and
// set named_auto_forward=yes in /etc/rc.conf.
// You can also enable
// named_auto_forward_only (the effect of which is //
described above).
// include "/etc/namedb/auto_forward.conf";

```

Как и говорится в комментариях, если вы хотите получить эффект от использования кэша провайдера, то можно включить раздел forwarders. В обычном случае сервер имён будет рекурсивно опрашивать определённые серверы имён Интернет до тех пор, пока не получит ответ на свой запрос. При включении этого раздела он будет автоматически опрашивать сервер имён вашего провайдера (или тот, который здесь указан), используя преимущества его кэша. наличия нужной информации. Если соответствующий сервер имён провайдера

работает быстро и имеет хороший канал связи, то в результате такой настройки вы можете получить хороший результат.

Предупреждение

127.0.0.1 здесь работать не будет. Измените его на IP-адрес сервера имён провайдера.

```
/*
Modern versions of BIND use a random UDP port for each
outgoing query by default in order to dramatically
reduce the possibility of cache poisoning. All users
are strongly encouraged to utilize this feature, and to
configure their firewalls to accommodate it. AS A LAST
RESORT in order to get around a restrictive firewall
policy, you can try enabling the option below. Use of
this option will significantly reduce your ability to
withstand cache poisoning attacks, and should be
avoided if possible. Replace NNNNN in the example with
a number between 49160 and 65530.
*/
// query-source address * port NNNNN;
};
// If you enable a local name server, don't forget to
// enter 127.0.0.1
// first in your /etc/resolv.conf so this server will
// be queried.
// Also, make sure to enable it in /etc/rc.conf.
// The traditional root hints mechanism. Use this, OR
// the slave zones below.
zone "." { type hint; file "/etc/namedb/named.root"; };
/* Slaving the following zones from the root name
servers has some
significant advantages:
1. Faster local resolution for your users
2. No spurious traffic will be sent from your network
to the roots
3. Greater resilience to any potential root server
failure/DDoS On the other hand, this method requires
more monitoring than the hints file to be sure that an
```

unexpected failure mode has not incapacitated your server. Name servers that are serving many clients will benefit more from this approach than individual hosts. Use with caution. To use this mechanism, uncomment the entries below, and comment the hint zone above. As documented at <http://dns.icann.org/services/axfr/> these zones:

"." (the root), ARPA, IN-ADDR.ARPA, IP6.ARPA, and ROOT-SERVERS.NET

are available for AXFR from these servers on IPv4 and IPv6:

`xfr.lax.dns.icann.org, xfr.cjr.dns.icann.org`

`*/`

`/*`

`zone "." {`

`type slave;`

`file "/etc/namedb/slave/root.slave";`

`masters {`

`192.5.5.241; // F.ROOT-SERVERS.NET.`

`};`

`notify no;`

`};`

`zone "arpa" {`

`type slave;`

`file "/etc/namedb/slave/arpa.slave";`

`masters {`

`192.5.5.241; // F.ROOT-SERVERS.NET.`

`};`

`notify no;`

`};`

`*/`

`/* Serving the following zones locally will prevent any queries for these zones leaving your network and going to the root name servers. This has two significant advantages:`

`1. Faster local resolution for your users`

`2. No spurious traffic will be sent from your network`

```

to the roots
*/
// RFCs 1912 and 5735 (and BCP 32 for localhost)
zone "localhost" { type master; file
"/etc/namedb/master/localhostforward.
db"; };
zone "127.in-addr.arpa" { type master; file
"/etc/namedb/master/
localhost-reverse.db"; };
zone "255.in-addr.arpa" { type master; file
"/etc/namedb/master/
empty.db"; };
// RFC 1912-style zone for IPv6 localhost address
zone "0.ip6.arpa" { type master; file
"/etc/namedb/master/localhostreverse.
db"; };
// "This" Network (RFCs 1912 and 5735)
zone "0.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// Private Use Networks (RFCs 1918 and 5735)
zone "10.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "16.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "17.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "18.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "19.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "20.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "21.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "22.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "23.172.in-addr.arpa" { type master; file

```

```

"/etc/namedb/master/empty.db"; };
zone "24.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "25.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "26.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "27.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "28.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "29.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "30.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "31.172.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "168.192.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// Link-local/APIPA (RFCs 3927 and 5735)
zone "254.169.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// IETF protocol assignments (RFCs 5735 and 5736)
zone "0.0.192.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// TEST-NET-[1-3] for Documentation (RFCs 5735 and
5737)
zone "2.0.192.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "100.51.198.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "113.0.203.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// IPv6 Range for Documentation (RFC 3849)
zone "8.b.d.0.1.0.0.2.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// Domain Names for Documentation and Testing (BCP 32)

```

```
zone "test" { type master; file
"/etc/namedb/master/empty.db"; };
zone "example" { type master; file
"/etc/namedb/master/empty.db"; };
zone "invalid" { type master; file
"/etc/namedb/master/empty.db"; };
zone "example.com" { type master; file
"/etc/namedb/master/empty.db"; };
zone "example.net" { type master; file
"/etc/namedb/master/empty.db"; };
zone "example.org" { type master; file
"/etc/namedb/master/empty.db"; };
// Router Benchmark Testing (RFCs 2544 and 5735)
zone "18.198.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "19.198.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// IANA Reserved - Old Class E Space (RFC 5735)
zone "240.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "241.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "242.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "243.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "244.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "245.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "246.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "247.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "248.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "249.in-addr.arpa" { type master; file
```

```
"/etc/namedb/master/empty.db"; };
zone "250.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "251.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "252.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "253.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "254.in-addr.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// IPv6 Unassigned Addresses (RFC 4291)
zone "1.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "3.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "4.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "5.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "6.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "7.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "8.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "9.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "a.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "b.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "c.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "d.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "e.ip6.arpa" { type master; file
```

```
"/etc/namedb/master/empty.db"; };
zone "0.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "1.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "2.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "3.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "4.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "5.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "6.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "7.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "8.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "9.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "a.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "b.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "0.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "1.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "2.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "3.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "4.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "5.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
```



```

zone "6.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "7.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// IPv6 ULA (RFC 4193)
zone "c.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "d.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// IPv6 Link Local (RFC 4291)
zone "8.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "9.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "a.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "b.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// IPv6 Deprecated Site-Local Addresses (RFC 3879)
zone "c.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "d.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "e.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
zone "f.e.f.ip6.arpa" { type master; file
"/etc/namedb/master/empty.db"; };
// IP6.INT is Deprecated (RFC 4159)
zone "ip6.int" { type master; file
"/etc/namedb/master/empty.db"; };
// NB: Do not use the IP addresses below, they are
faked, and only
// serve demonstration/documentation purposes!
//
// Example slave zone config entries. It can be
// convenient to become a slave at least for the zone
// your own domain is in. Ask your network

```

```

// administrator for the IP address of the responsible
// master name server.
//
// Do not forget to include the reverse lookup zone!
// This is named after the first bytes of the IP
// address, in reverse
//order, with ".IN-ADDR.ARPA" appended, or ".IP6.ARPA"
// for IPv6.
//
// Before starting to set up a master zone, make sure
// you fully understand how DNS and BIND work. There
// are sometimes non-obvious pitfalls. Setting up a
// slave zone is usually simpler.
//
// NB: Don't blindly enable the examples below. :-) Use
// actual names and addresses instead.
/* An example dynamic zone
key "exampleorgkey" {
algorithm hmac-md5;
secret "sf87HJqjkqh8ac87a0211a==";
};
zone "example.org" {
type master;
allow-update {
key "exampleorgkey";
};
file "dynamic/example.org";
};
*/
/* Example of a slave reverse zone
zone "1.168.192.in-addr.arpa" {
type slave;
file "/etc/namedb/slave/1.168.192.in-addr.arpa";
masters {
192.168.1.1;
};

```

```
};*/
```

Это примеры описаний прямой и обратной зон из файла `named.conf` для вторичных серверов.

Для каждого новой зоны, которую будет обслуживать сервер имён, в файл `named.conf` должна быть добавлена запись.

К примеру, самая простая запись для домена `example.org` может выглядеть вот так:

```
zone "example.org" {  
    type master;  
    file "master/example.org";  
};
```

Зона является первичной, что отражается в поле `type`, и информация о зоне хранится в файле [/etc/namedb/master/example.org](#), что указывается в поле `file`.

```
zone "example.org" {  
    type slave;  
    file "slave/example.org";  
};
```

В случае вторичной зоны информация о ней передается с основного сервера имён для заданной зоны и сохраняется в указанном файле. Если и когда основной сервер имён выходит из строя или недостижим, то скачанная информация о зоне будет находиться на вторичных серверах, и они смогут обслуживать эту зону.

Файлы зон

Пример файла зоны `example.org` для основного сервера (распологающийся в файле `/etc/namedb/master/example.org`) имеет такой вид:

```
$TTL 3600 ; 1 hour
```

```

example.org.          IN  SOA  ns1.example.org.
admin.example.org. (
                        2006051501 ;Serial
                        10800      ;Refresh
                        3600       ;Retry
                        604800     ;Expire
                        300        ;Negative Response TTL
); DNS Servers
      IN  NS  ns1.example.org.
      IN  NS  ns2.example.org.
; MX Records
      IN  MX 10  mx.example.org.
      IN  MX 20  mail.example.org.
      IN  A   192.168.1.1
; Machine Names
Localhost  IN  A   127.0.0.1
ns1        IN  A   192.168.1.2
ns2        IN  A   192.168.1.3
mx         IN  A   192.168.1.4
mail       IN  A   192.168.1.5
; Aliases
www        IN  CNAME example.org.

```

Заметьте, что все имена хостов, оканчивающиеся на «.», задают полное имя, тогда как все имена без символа «.» на конце считаются заданными относительно [origin](#). Например, ns1 преобразуется в ns1.example.org. Файл зоны имеет следующий формат:

```

recordname      IN      recordtype  value

```

Наиболее часто используемые записи DNS:

- *SOA* – начало зоны ответственности.
- *NS* – авторитативный сервер имен.
- *A* – адрес хоста.
- *CNAME* – каноническое имя для алиаса.

- *MX* – обмен почтой.
- *PTR* – указатель на доменное имя (используется в обратных зонах DNS).

```
example.org. IN SOA ns1.example.org. admin.example.org. (
    2006051501 ; Serial
    10800      ; Refresh after 3 hours
    3600       ; Retry after 1 hour
    604800     ; Expire after 1 week
    300 )      ; Minimum TTL of 1 day
```

- *example.org* – имя домена, а так же [ориджин](#) для этого файла зоны.
- *ns1.example.org* – основной/авторитативный сервер имён для этой зоны.
- *admin.example.org* – человек, отвечающий за эту зону, адрес электронной почты с символом "@" замененным на точку. (<admin@example.org> становится admin.example.org)
- *2006051501* – последовательный номер файла. При каждом изменении файла зоны это число должно увеличиваться. В настоящее время для нумерации многие администраторы предпочитают формат гтггммддвв. 2006051501 будет означать, что файл последний раз изменялся 15.05.2006, а последнее число 01 означает, что это была первая модификация файла за день.
- *IN NS ns1.example.org* – это NS-запись. Такие записи должны иметься для всех серверов имён, которые будут отвечать за зону.

<i>localhost</i>	<i>IN</i>	<i>A</i>	<i>127.0.0.1</i>
<i>ns1</i>	<i>IN</i>	<i>A</i>	<i>192.168.1.2</i>
<i>ns2</i>	<i>IN</i>	<i>A</i>	<i>192.168.1.3</i>
<i>mx</i>	<i>IN</i>	<i>A</i>	<i>192.168.1.4</i>
<i>mail</i>	<i>IN</i>	<i>A</i>	<i>192.168.1.5</i>

Записи типа A служат для обозначения имён машин. Как это видно выше, имя ns1.example.org будет преобразовано в 192.168.1.2.

```
IN      A      192.168.1.1
```

Эта строка присваивает IP адрес 192.168.1.1 текущему [ориджину](#), в данном случае домену *example.org*.

```
www     IN CNAME  @
```

Записи с каноническими именами обычно используются для присвоения машинам псевдонимов. В этом примере *www* является псевдонимом для "главной" машины, соответствующей ориджину, то есть [example.org](#) (192.168.1.1). Записи CNAME могут использоваться для присвоения псевдонимов именам хостов или для использования одного имени несколькими машинами по очереди.

```
IN      MX      10      mail.example.org.
```

MX-запись указывает, какие почтовые серверы отвечают за обработку входящей электронной почты для зоны. [mail.example.org](#) является именем почтового сервера, а 10 обозначает приоритет этого почтового сервера.

Можно иметь несколько почтовых серверов с приоритетами, например, 10, 20 и так далее. Почтовый сервер, пытающийся доставить почту для *example.org*, сначала попытается связаться с машиной, имеющий MX-запись с самым большим приоритетом (наименьшим числовым значением в поле MX), затем с приоритетом поменьше и так далее, до тех пор, пока почта не будет отправлена.

Для файлов зон [in-addr.arpa](#) (обратные записи DNS) используется тот же самый формат, отличающийся только использованием записей PTR вместо A или CNAME.

```
$TTL 3600
```

```
1.168.192.in-addr.arpa.      IN      SOA      ns1.example.org.  
admin.example.org. (        2006051501      ; Serial
```

```

10800      ; Refresh
3600       ; Retry
604800    ; Expire
300       ) ; Negative Response TTL

IN NS ns1.example.org.
IN NS ns2.example.org.

1 IN PTR example.org.
2 IN PTR ns1.example.org.
3 IN PTR ns2.example.org.
4 IN PTR mx.example.org.
5 IN PTR mail.example.org.

```

В этом файле дается полное соответствие имён хостов IP-адресам в нашем описанном ранее вымышленном домене.

Кэширующий сервер имён

Кэширующий сервер имён - это сервер имён, не отвечающий ни за какую зону. Он просто выполняет запросы от своего имени и сохраняет результаты для последующего использования. Для настройки такого сервера достаточно исключить все описания зон из стандартной конфигурации сервера имён.

Безопасность

Хотя [BIND](#) является самой распространенной реализацией [DNS](#), всегда стоит вопрос об обеспечении безопасности. Время от времени обнаруживаются возможные и реальные бреши в безопасности.

FreeBSD автоматически запускает `named` в ограниченном окружении (`chroot`); помимо этого, есть еще несколько механизмов, помогающих защититься от возможных атак на сервис DNS.

Подсказка

Если возникают проблемы, то наличие последних исходных текстов и свежееоткомпилированного `named` не мешает.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Произвести настройку DNS-клиента и DNS-сервера. Для установки необходимо:

1. Ознакомиться с предложенным материалом для получения базовой информации о DNS в ОС FreeBSD.
 2. Отредактировать файл */etc/resolv.conf*.
 3. Используя команду *ping* проверить правильность настройки.
 4. Отредактировать файл */etc/namedb*.
 5. Настроить кэширующий DNS-сервер (BIND).
 6. Настроить зону прямого отображения для учебной сети FreeBSD.
 7. Настроить зону обратного отображения для учебной сети FreeBSD.
 8. Проверить работоспособность DNS-клиента.
 9. Проверить работоспособность DNS-сервера.
- Ответить на контрольные вопросы и подготовить отчет.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Раскройте значение термина DNS.
2. Раскройте основные термины, связанные с работой DNS.
3. Перечислите причины, по которым может понадобиться сервер имен.
4. Назовите программу в ОС FreeBSD, отвечающую за работу системы DNS.
5. Опишите назначение BIND.
6. Предложите пути запуска BIND.
7. Назовите файл, используемый для настройки DNS клиента.
8. Дайте определение понятию зона в DNS.
9. Перечислите типы зон.
10. Назовите причины использования кэширующего сервера имен.
11. Назовите программу для управления сервером имен.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 2 занятия (4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, ответы на контрольные вопросы, описание процесса выполнения лабораторной работы, выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Вирт, Н. Разработка операционной системы и компилятора. Проект Оберон [Электронный ресурс] / Н. Вирт, Ю. Гуткнехт ; пер.с англ. Борисов Е.В., Чернышов Л.Н.. — Электрон. дан. — Москва : ДМК Пресс, 2012. — 560 с. — Режим доступа: <https://e.lanbook.com/book/39992>

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

2. Крищенко, В.А. Сервисы Windows [Электронный ресурс] : учебное пособие / В.А. Крищенко, Н.Ю. Рязанова. — Электрон. дан. — Москва : МГТУ им. Н.Э. Баумана, 2011. — 47 с. — Режим доступа: <https://e.lanbook.com/book/52416..>

3. Войтов, Н.М. Администрирование ОС Red Hat Enterprise Linux. Учебный курс [Электронный ресурс] : учебное пособие / Н.М. Войтов. — Электрон. дан. — Москва : ДМК Пресс, 2011. — 192 с. — Режим доступа: <https://e.lanbook.com/book/1081>

4. Стащук, П.В. Администрирование и безопасность рабочих станций под управлением Mandriva Linux: лабораторный практикум [Электронный ресурс] : учебно-методическое пособие / П.В. Стащук. — Электрон. дан. — Москва : ФЛИНТА, 2015. — 182 с. — Режим доступа: <https://e.lanbook.com/book/70397>

Электронные ресурсы:

5. Научная электронная библиотека <http://eLIBRARY.RU>
6. Электронно-библиотечная система <http://e.lanbook.com>
7. Losst - Linux Open Source Software Technologies <https://losst.ru>