

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П.КОРОЛЕВА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

КОНФИГУРАЦИЯ СЕТЕВОГО СОЕДИНЕНИЯ

**САМАРА
2013**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

КОНФИГУРАЦИЯ СЕТЕВОГО СОЕДИНЕНИЯ

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Самарский государственный аэрокосмический университет имени академика С. П. Королева (национальный исследовательский университет в качестве методических указаний к лабораторной работе

САМАРА
Издательство СГАУ
2013

УДК СГАУ : 6(075)

Рецензент д-р техн. наук, профессор С. А. П р о х о р о в

Конфигурация сетевого соединения: метод. указания к лабор. работе / сост. А.М. Сухов. – Самара: Изд-во СГАУ, 2013. – 20 с.

Приведен материал, необходимый для выполнения лабораторных работ по дисциплинам «Операционная система Linux на высокопроизводительных кластерах», «Перспективные информационные технологии»

Целью лабораторных работ является изучение основных возможностей по конфигурации сетевого соединения и приобретение навыков работы в операционной системе Linux.

Предназначены для слушателей ФПК СГАУ и студентов специальностей 010500, 010501

УДК СГАУ : 6(075)

© Самарский государственный
аэрокосмический университет, 2013

Учебное издание

КОНФИГУРАЦИЯ СЕТЕВОГО СОЕДИНЕНИЯ

Методические указания к лабораторной работе

Составитель: *Сухов Андрей Михайлович*

Редактор И.И. Спиридонова
Доверстка И.И. Спиридонова

Подписано в печать 5.09.13. Формат 60x84 1/16.
Бумага офсетная. Печать офсетная.
Печ. л. 1,25. Тираж 100 экз. Заказ . Арт. – М8/2013.

Самарский государственный аэрокосмический университет
443086, г. Самара, Московское шоссе, 34.

Издательство Самарского государственного аэрокосмического университета
443086, г. Самара, Московское шоссе, 34.

1. ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

- Освоить основные понятия о конфигурации протокола TCP/IP.
- Уметь самостоятельно конфигурировать сетевое соединение на рабочей станции.
- Иметь представление о конфигурации маршрутизатора под управлением ОС Linux.

2. ОСНОВНЫЕ СВЕДЕНИЯ

Любая компьютерная сеть предполагает присвоение адреса сетевому устройству. В качестве адреса в компьютерных сетях всегда используется набор битов некоторой длины, которая меняется в зависимости от сетевого стандарта. В изучаемой нами версии TCP/IP четвертого поколения (IPv4) длина адреса – 32 бита, разделенных побайтно. Для практических настроек двоичное значение каждого байта переводится в десятичную форму записи, все 4 полученных числа разделяются точками. Но и десятичная запись адреса хоста вызывает у большинства обычных пользователей затруднения, поэтому были специально придуманы доменные имена, приближенные к обычным адресным записям и представляющие совокупность нескольких слов или аббревиатур, выстроенных в иерархическом порядке.

Файлы конфигурации TCP/IP

Для настройки и поддержки работы сети, работающей под управлением протоколов TCP/IP, используется набор файлов конфигурации, расположенных в каталоге */etc*. В этих файлах содержится информация о сети, в частности, имена хост-машин и доменов, IP-адреса и характеристики интерфейсов. Именно в эти файлы вводятся IP-адреса и доменные имена других хост-компьютеров Internet, к которым вы хотите получить доступ. Если в процессе инсталляции системы вы конфигурировали сеть, то вся эта информация в файлах конфигурации уже есть. Ввести конфигурационные данные в эти файлы можно с помощью программы *netcfg* (ее пиктограмма находится на вашем рабочем столе) или с помощью программы *netconfig* (из командной строки). В простейшем случае можно просто отредактировать файлы с помощью простейшего текстового редактора *nano*.

<u>Файл</u>	<u>Функция</u>
<code>/etc/hosts</code>	Связывает хост-имена с IP-адресами.
<code>/etc/networks</code>	Связывает доменные имена с адресами сетей.
<code>/etc/rc.d/init.d/inet</code>	Содержит команды конфигурирования сетевого интерфейса при начальной загрузке.
<code>/etc/HOSTNAME</code>	Содержит хост-имя вашей системы.
<code>/etc/host.conf</code>	Опции конфигурирования.
<code>/etc/resolv.conf</code>	Содержит список серверов доменных имен.

Идентификация хост-имен: файл `/etc/hosts`

Без уникального IP-адреса, которым в сети TCP/IP идентифицируются компьютеры, нужный компьютер найти нельзя. Поскольку IP-адреса трудны для запоминания и работы, вместо них используются доменные имена. Каждому IP-адресу ставится в соответствие доменное имя. Система преобразует доменное имя, по которому пользователь обращается к определенному компьютеру, в соответствующий IP-адрес, и он используется для установления соединения с указанным компьютером.

Вначале ведение списка хост-имен с их IP-адресами было обязанностью всех компьютеров сети. Этот список до сих пор хранится в файле `/etc/host`. Получив от пользователя доменное имя, система ищет в файле `hosts` соответствующий адрес. За ведение этого списка отвечает системный администратор. Вследствие стремительного роста сети Internet и появления все новых очень больших сетей функции преобразования доменных имен в IP-адреса были переданы серверам доменных имен. Тем не менее, файл `hosts` продолжает использоваться для хранения доменных имен и IP-адресов хост-компьютеров, соединения с которыми устанавливаются наиболее часто. Перед тем как обращаться к серверу имен, ваша система всегда будет проверять файл `hosts` и искать в нем IP-адрес заданного ей доменного имени.

Каждая запись в файле `hosts` состоит из IP-адреса, пробела и доменного имени. Для хост-имени можно создавать псевдонимы. В одной строке с записью можно ввести комментарий, который всегда предваряется символом `#`. В файле `hosts` уже имеется запись для локального компьютера `localhost` с IP-адресом `127.0.0.1`. `localhost` - это специальный зарезервированный IP-адрес `127.0.0.1`, которой позволяет пользователям вашей системы связы-

ваться друг с другом в локальном режиме. Он служит для идентификации так называемого закольцовывающего интерфейса.

/etc/hosts

127.0.0.1	turtle.trek.com	localhost
199.35.209.72	turtle.trek.com	
204.32.168.56	pangol.train.com	
202.211.234.1	rose.berkeley.edu	

Инициализация сетевого соединения: файл */etc/rc.d/init.d/inet*

В файле */etc/rc.d/init.d/inet* находятся команды, обеспечивающие конфигурирование сетевого соединения. Многие записи в этом файле автоматически создаются при использовании утилиты *netcfg* и конфигурировании сетевого соединения в процессе инсталляции. Например, здесь находятся команды *ifconfig* и *route*. Кроме того, здесь задаются хост-имя вашей системы, адрес сети и другие необходимые адреса. Непосредственно редактировать этот файл можно лишь в том случае, если вы уверены, что все делаете правильно и обладаете начальными познаниями в области программирования в *shell*.

В других дистрибутивах Linux, например в Slackware, файл инициализации может иметь имя */etc/rc.d/rc-inet1* или просто */etc/rc.inet1*.

Файл */etc/HOSTNAME*

В файле */etc/HOSTNAME* содержится хост-имя вашей системы. Чтобы изменить имя, нужно отредактировать данный файл. Эту задачу можно решить с помощью программы *netcfg*, которая заменяет хост-имя и помещает новое имя в файл */etc/HOSTNAME*. Хост-имя можно узнать не только путем вывода на экран этого файла, но и с помощью команды *hostname*.

```
$ hostname
```

```
turtle.trek.corn
```

Настройка сетевых интерфейсов

Интерфейсом с точки зрения ОС является устройство, через которое система получает и передает IP-пакеты. Роль интерфейса локальной сети может выполнять одно (или несколько) из следующих устройств: Ethernet-карта, ISDN-адаптер или модем, подключенный к последовательному порту. Каждое устройство (не весь компьютер!) имеет свой IP-адрес. Для выхода в локальные сети используется, как правило, Ethernet-карта, что и будет предполагаться в настоящем разделе. Заодно мы рассмотрим и настройку модемного интерфейса, поскольку настраивается он вполне аналогично.

Расположение конфигурационных файлов

Отметим сразу, что все приводимые ниже команды можно выполнять из командной строки, но тогда придется повторять эти операции при каждом перезапуске компьютера. Поэтому может быть удобнее записать их в один из инициализационных файлов, автоматически запускаемых при старте системы. В разных дистрибутивах процесс загрузки организован по-разному.

В «Linux NET-3-HOWTO» приводится следующая таблица 1.

Таблица 1. Расположение конфигурационных файлов в основных дистрибутивах

Дистрибутив	Настройка интерфейса и маршрутизации	Запуск демонов
Debian	/etc/network/interfaces	/etc/init.d/netbase /etc/init.d/netstd_init /etc/init.d/netstd_nfs /etc/init.d/netstd_misc
Slackware	/etc/rc.d/rc.inet1	/etc/rc.d/rc.inet2
RedHat	/etc/sysconfig/network-scripts/ifup-<ifname>	/etc/rc.d/init.d/network

Обратите внимание, что дистрибутивы Debian и Red Hat содержат отдельный каталог для скриптов запуска системных сервисов (хотя сами файлы настроек находятся в других местах, например, в дистрибутиве Red Hat они хранятся в каталоге */etc/sysconfig*).

Для понимания процесса загрузки ознакомьтесь с содержимым файла */etc/inittab* и документацией по процессу *init*.

Команда *ifconfig*

После подключения драйверов вы должны настроить те интерфейсы, которые вы предполагаете использовать. Настройка интерфейса заключается в присвоении IP-адресов сетевому устройству и установке нужных значений для других параметров сетевого подключения. Наиболее часто для этого используется программа *ifconfig* (ее название происходит от «interface configuration»). На лекциях были подробно рассмотрены настройки протокола TCP/IP для локальной сети и формат команды *ifconfig (ip address* для Cisco IOS, *ipconfig* для Windows).

Запустите ее без аргументов (или с единственным аргументом *-a*) и вы узнаете, какие параметры установлены в данный момент для активных сетевых интерфейсов (в частности, для сетевой карты). Кстати, имеет смысл выполнить эту команду еще до подключения модулей: а вдруг у вас поддержка интерфейсов встроена в ядро и необходимые настройки сделаны в процессе инсталляции системы. Тогда вы в ответ можете получить информацию о параметрах вашей Ethernet-карты и так называемого «кольцевого интерфейса» или «обратной петли» - Local Loopback (интерфейс Ethernet при единственной сетевой карте обозначается как *eth0*, а кольцевой интерфейс - как *lo*). Если же по этой команде вы ничего не получите, то надо переходить к подключению модулей и настройке, и начинать надо с кольцевого интерфейса.

Настройка локального интерфейса *lo*

Этот интерфейс используется для связи программ IP-клиентов с IP-серверами, запущенными на той же машине, так что его необходимо настроить даже в том случае, если вы вообще не подключаете никаких сетевых устройств.

Локальный интерфейс настраивается очень просто – командой

```
[root]# /sbin/ifconfig lo 127.0.0.1
```

Теперь, чтобы проверить работоспособность протоколов TCP/IP на вашей машине, дайте команду:

```
[root]# ping 127.0.0.1
```


Настройка интерфейса платы Ethernet локальной сети (*eth0*)

Для того чтобы ваш компьютер вошел в сеть с IP-адресом, полученным вами у администратора (пусть для примера это будет адрес 10.101.1.25), необходимо запустить команду *ifconfig* примерно следующим образом:

```
[root]# /sbin/ifconfig eth0 10.101.1.25 netmask 255.254.0.0 broadcast  
10.101.255.255 up
```

Интерфейс для последовательного порта (*ttyS0*)

Последовательный порт используется для подключения модема, через который осуществляется соединение с сетью по телефонной линии. Для настройки интерфейса этого типа тоже можно использовать программу *ifconfig*. Однако, такие программы как *pppd* и *dip*, используемые для соединения с сетью по модему, способны автоматически конфигурировать сетевой интерфейс, поэтому обычно для этого случая применять *ifconfig* не требуется.

Настройка маршрутизации

Правила маршрутизации определяют, куда перенаправлять IP-пакеты. Данные маршрутизации хранятся в одной из таблиц ядра. Вести таблицы маршрутизации можно статически или динамически.

Статический маршрут – это маршрут, который задается явно с помощью команды *route*. Динамическая маршрутизация выполняется процессом-демоном (*routed* или *gated*), который ведет и модифицирует таблицу маршрутизации на основе сообщений от других компьютеров сети. Для выполнения динамической маршрутизации разработаны специальные протоколы: RIP, OSPF, IGRP, EGP, BGP и т.д.

Динамическая маршрутизация необходима в том случае, если у вас сложная, постоянно меняющаяся структура сети и одна и та же машина может быть доступна по различным интерфейсам (например, через разные Ethernet или PPP интерфейсы).

Маршруты, заданные статически, обычно не меняются, даже если используется динамическая маршрутизация.

Для персонального компьютера, подключаемого к локальной сети, в большинстве ситуаций бывает достаточно статической маршрутизации командой *route*. Прежде чем пытаться настраивать маршруты, просмотрите

таблицу маршрутизации ядра с помощью команды **netstat -n -r**. Вы должны увидеть что-то вроде следующего

```
[root]# netstat -nr
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
10.72.128.101	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
10.72.128.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo
0.0.0.0	10.72.128.254	0.0.0.0	UG	0	0	0	eth0

Если таблица пуста, то вы увидите только заголовки столбцов. Тогда надо использовать **route**. С помощью команды **route** можно добавить или удалить один (за один раз) статический маршрут. Вот ее формат:

```
[root]# /sbin/route [-f] операция [-mun] адресам шлюз [dev] интерфейс
```

Здесь аргумент **операция** может принимать одно из двух значений: **add** (маршрут добавляется) или **delete** (маршрут удаляется). Аргумент **адресат** может быть IP-адресом машины, IP-адресом сети или ключевым словом **default**. Аргумент **шлюз** – это IP-адрес компьютера, на который следует пересылать пакет (этот компьютер должен иметь прямую связь с вашим компьютером).

Команда

```
[root]# /sbin/route -f
```

удаляет из таблицы данные обо всех шлюзах. Необязательный аргумент **тип** принимает значения **net** или **host**. В первом случае в поле адресата указывается адрес сети, а во втором – адрес конкретного компьютера (хоста).

Как правило, бывает необходимо настроить маршрутизацию по упоминавшимся выше трем интерфейсам:

- локальный интерфейс (lo),
- интерфейс для платы Ethernet (eth0),
- интерфейс для последовательного порта (PPP или SLIP).

Локальный интерфейс поддерживает сеть с IP-номером 127.0.0.1. Поэтому для маршрутизации пакетов с адресом 127.... используется команда:

```
[root]# /sbin/route add -net 127.0.0.1 lo
```

Если у вас для связи с локальной сетью используется одна плата Ethernet, и все машины находятся в этой сети (сетевая маска 255.255.255.0), то для настройки маршрутизации достаточно вызвать:

```
[root]# /sbin/route add -net 192.168.36.0 netmask 255.255.255.0 eth0
```

Если же вы имеете несколько интерфейсов, то вам надо определиться с сетевой маской и вызвать команду **route** для каждого интерфейса.

Поскольку очень часто IP-пакеты с вашего компьютера могут отправляться не в одну единственную сеть, а в разные сети (например, при просмотре разных сайтов в Интернете), то в принципе надо было бы задать очень много маршрутов. Очевидно, что сделать это было бы очень сложно, точнее просто невозможно. Поэтому решение проблемы маршрутизации пакетов перекладывают на плечи специальных компьютеров – маршрутизаторов, а на обычных компьютерах задают маршрут по умолчанию, который используется для отправки всех пакетов, не указанных явно в таблице маршрутизации. С помощью маршрута по умолчанию вы говорите ядру «а все остальное отправляй туда».

Маршрут по умолчанию настраивается следующей командой:

```
[root]# /sbin/route add default gw 192.168.1.1 eth0
```

Опция **gw** указывает программе **route**, что следующий аргумент – это IP-адрес или имя маршрутизатора, на который надо отправлять все пакеты, соответствующие этой строке таблицы маршрутизации.

После настройки маршрутизации можно проверить, что у вас получилось. Для этого снова дайте команду

```
[root]# netstat -nr
```

Если вывод команды выглядит так, как это было показано выше, но не содержит строки, которая в графе **Destination** содержит **0.0.0.0**, а в графе **Gateway** указывает на маршрут, используемый для соединений по умолчанию, то вы, вероятно, не задали этот маршрут.

Настройка службы имен

С помощью команды **ifconfig** вы задали IP-адрес вашего компьютера, но он еще не знает своего имени (при инсталляции системы он получил обезличенное имя localhost). Существует команда **hostname**, которая позволяет установить (и узнать действующее в данный момент) имя компьютера и имя домена.

Однако установить только имя и только этой командой еще недостаточно, поскольку эта команда меняет имя только на текущий сеанс работы. Поэтому обычно эта команда вызывается в одном из инициализационных файлов, например, */etc/rc.d/rc* или */etc/rc.d/rc.local*. Вы можете попытаться найти ее там, чтобы изменить должным образом имя компьютера, которое задается в качестве параметра команды **hostname**. В таком случае требуется перезагрузиться для того чтобы изменения вступили в силу.

Другой способ изменения имени компьютера или домена состоит в том, что эти имена прописываются в файле */etc/sysconfig/network* в виде двух строчек примерно следующего вида:

```
HOSTNAME=«new_host_name.localdomain.upperdomain»
```

```
DOMAINNAME=localdomain.upperdomain
```

Тогда в процессе инициализации системы эти имена будут восстанавливаться, потому что файл */etc/sysconfig/network* вызывается из */etc/rc.d/rc.sysinit*.

Кроме того, имя компьютера должно быть прописано в файле */etc/hosts*, который связывает имя компьютера с его IP-адресом. Каждая строка файла */etc/hosts* должна начинаться с IP-адреса, за которым следует имя данного узла. Следом за именем можно записать произвольное число псевдонимов этого узла.

Даже если ваш компьютер не подключен к сети, в файле */etc/hosts* должна быть прописана хотя бы одна строка следующего вида.

127.0.0.1 localhost localhost.localdomain

Если же ваш компьютер подключен к TCP/IP сети, то в этом файле дополнительно нужно прописать строку вида

192.168.0.15 host_name host_name.localdomain

Файл */etc/hosts* используется в механизмах разрешения имен. В больших сетях трудно было бы поддерживать в актуальном состоянии файлы */etc/hosts* на всех компьютерах, если бы это был основной инструмент для определения IP-адресов по именам. Поэтому обычно для разрешения имен используются серверы DNS. Однако файл */etc/hosts* все равно необходим, хотя бы для обращения к серверу DNS. Поэтому в нем имеет смысл указать IP-адреса и соответствующие имена шлюзов и серверов DNS и NIS. А чтобы все приложения использовали этот файл при разрешении имен, должен иметься файл */etc/hosts.conf*, содержащий строку

order hosts,bind

которая говорит, что при разрешении имен сначала должен использоваться файл */etc/hosts*, а затем должно происходить обращение к серверу DNS.

В большинстве случаев в файле */etc/hosts.conf* достаточно иметь две строки:

order hosts,bind

multi on

Эти параметры указывают системе преобразования имен, что надо просмотреть файл */etc/hosts* перед тем, как посылать запрос к серверу, и что следует возвращать все найденные в */etc/hosts* адреса для данного имени, а не только первый.

Но настройка механизма разрешения имен не ограничивается редактированием файлов */etc/hosts* и */etc/hosts.conf*. Необходимо еще указать компьютеру имена серверов DNS. Они прописываются в файле */etc/resolv.conf*. Этот файл имеет весьма простой формат. Это текстовый файл, каждая строка которого задает один из параметров системы преобразования имен.

Как правило, используются три ключевых слова-параметра:

- **domain** – задает имя локального домена.
- **Search** – задает список имен доменов, которые будут добавляться к имени машины, если вы не укажете явно имени домена. Это позволяет ограничить область поиска и избежать некоторых ошибок (например, вы ищете компьютер *linux.msk.ru*, а механизм разрешения имен выведет вас на *linux.spb.ru*).
- **Nameserver** – это параметр, который вы можете указывать несколько раз, задает IP-адрес сервера преобразования имен, на который ваша машина будет посылать запросы. Повторяя этот параметр, вы можете задать несколько серверов.

Если вы не собираетесь заводить поддержку сервиса имен для своей сети (что является довольно сложной организационной и технической проблемой), и доверяете ведение своих имен администратору локальной сети или вашему IP-провайдеру, то вам достаточно задать файл */etc/resolv.conf* примерно следующего вида:

domain abcd.ru

search abcd.ru xyz.edu.ru

nameserver 10.100.100.5

nameserver 10.100.100.8

В этом примере машина находится в домене abcd.ru. Если вы зададите имя машины, не указывая домена, например «pc1», то система преобразования имен попытается сначала найти машину «pc1.abcd.ru», а в случае неудачи – «pc1.xyz.edu.ru». Для преобразования имен ваша машина будет обращаться к серверам по адресам «10.100.100.5 и «10.100.100.8».

Тестирование сетевого соединения

Чтобы проверить, соединяется ли ваш компьютер с сетью, попробуйте дать команду *ping*, указав ей в качестве параметра IP-адрес одного из компьютеров сети. Пусть, например, вам известно (узнайте реальный номер и имя у администратора сети), что в сети есть компьютер с IP-адресом 10.100.100.1 и именем big.smmc. Тогда вы должны дать команду:

```
[user]$ ping 10.100.100.1
```

или (тут вы одновременно проверяете и работу службы DNS)

```
[user]$ ping big.smmc
```

Если соединение с сетью установлено, должны появиться и периодически обновляться строчки примерно такого вида:

```
64 bytes from 192.168.0.2: icmp_seq=0 ttl=32 time=1.2 ms
64 bytes from 192.168.0.2: icmp_seq=1 ttl=32 time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=32 time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=32 time=1.0 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=32 time=1.1 ms
```

Это означает, что сетевое соединение работает. Команда *ping* посылает адресату ICMP-пакет типа ECHO REQUEST, выставляя в нем время и его идентификатор. Ядро машины-получателя отвечает на подобный запрос пакетом ICMP ECHO REPLY. Получив его *ping* измеряет время от момента отправления пакета до его получения. Для того чтобы прервать тестирование сети, нажмите комбинацию клавиш <Ctrl>+<C>.

Настройка сетевого подключения в Linux

Для нормальной работы сетевого подключения в Linux может потребоваться настроить DHCP-клиент таким образом, чтобы отправляемые DHCP-запросы включали идентификатор виртуальной машины. Подробную информацию о настройке вашего DHCP-клиента вы можете найти в его документации.

Например, в гостевых операционных системах Red Hat Linux это делается с помощью редактирования конфигурационного файла DHCP-клиента.

1. Откройте файл *dhclient.conf* и проверьте, присутствуют ли там следующие строки.

```
interface «eth0» {  
send dhcp-client-identifier 1:<MAC address>;  
}
```

2. Если нет, добавьте их и сохраните файл.

При наличии этих строк в файле *dhclient.conf* запросы, отправляемые DHCP-клиентом серверу, будут включать идентификатор виртуальной машины. В ответ на эти запросы DHCP-сервер будет присылать соответствующий IP-адрес.

Совет

Чтобы найти файл *dhclient.conf*, введите:

```
strings /sbin/dhclient | grep etc | grep dhclient.conf
```

или:

```
rpm -ql dhclient
```

Настройка PPPoE-соединения для ОС Linux

Для установки и настройки PPPoE клиента под ОС Linux необходимы:

1. Point-To-Point Protocol Daemon
2. гр-pppoe redirector

Для работы PPPoE *pppd* должен быть версии не ниже 2.3.7.

Узнать версию пакета можно командой:

```
[root@ck sbin]#rpm -q ppp  
ppp-2.4.4-1.fc6
```

Скачать *pppd* можно из используемого Вами дистрибутива Linux.

RP-PPPoE пакет, как правило, поставляется во всех современных версиях дистрибутивов Linux.

Установка

Установки PPPoE клиента из исходников

1. Распаковываем скачанный архив, например, гр-pppoe-3.5.tar.gz

```
# tar zxvf rp-pppoe-3.5.tar.gz
```

2. Заходим в созданную директорию (гр-pppoe-3.5)

```
# cd rp-pppoe-3.5
```

3. Из под *root* пользователя запускаем установочный скрипт

```
#!/go
```

В процессе работы скрипта будет сконфигурировано, скомпилировано и установлено необходимое программное обеспечение. После установки ПО автоматически будет запущен скрипт конфигурирования PPPoE соединения.

Установка PPPoE-клиента из исходников вручную: если по каким-либо причинам не удалось установить PPPoE-клиента следуя пунктам 2.1 и 2.2, Вы можете попробовать установить клиента вручную.

Для этого необходимо:

1. Зайти в директорию «src» находящуюся в каталоге, куда Вы распаковали исходники.

2. Запустить конфигурационный скрипт

```
# ./configure
```

3. Запустить компиляцию исходников

```
# make
```

4. Запустить из под *root* пользователя установку ПО

```
# make install
```

5. Запустить из под *root* пользователя конфигурационный скрипт для настройки PPPoE соединения

```
# adsl-setup
```

Настройка

```
[root@ck ~]#adsl-setup
```

Welcome to the ADSL client setup.

First, I will run some checks on your system to make sure the PPPoE client is installed properly...

```
LOGIN NAME
```

Enter your Login Name (default root): «укажите **Ваш логин на подключение**»

```
INTERFACE
```

Enter the Ethernet interface connected to the ADSL modem

For Solaris, this is likely to be something like */dev/hme0*.

For Linux, it will be *ethX*, where 'X' is a number.

(*default eth0*): **«укажите имя сетевого интерфейса, подключенного к розетке (обычно это eth0)»**

Do you want the link to come up on demand, or stay up continuously?

If you want it to come up on demand, enter the idle time in seconds after which the link should be dropped. If you want the link to stay up permanently, enter 'no' (two letters, lower-case.)

NOTE: Demand-activated links do not interact well with dynamic IP addresses. You may have some problems with demand-activated links.

Enter the demand value (default no): **«укажите yes, если хотите, чтобы соединение подключалось автоматически при запросе. В противном случае укажите no».**

DNS

Please enter the IP address of your ISP's primary DNS server.

If your ISP claims that 'the server will provide dynamic DNS addresses', enter 'server' (all lower-case) here.

If you just press enter, I will assume you know what you are doing and not modify your DNS setup.

Enter the DNS information here: **«укажите IP адрес DNS сервера (217.170.80.83)»**

Please enter the IP address of your ISP's secondary DNS server.

If you just press enter, I will assume there is only one DNS server.

Enter the secondary DNS server address here: **«укажите IP адрес альтернативного DNS сервера (81.222.223.6)»**

PASSWORD

Please enter your Password: **«введите ваш пароль».**

Please re-enter your Password: **«подтвердите ваш пароль».**

USERCTRL

Please enter 'yes' (three letters, lower-case.) if you want to allow normal user to start or stop DSL connection (default yes): **«введите yes, если вы хотите, чтобы позволить обычному пользователю (не root) включать и выключать соединение»**

FIREWALLING

Please choose the firewall rules to use. Note that these rules are very basic. You are strongly encouraged to use a more sophisticated firewall setup; however,

these will provide basic security. If you are running any servers on your machine, you must choose 'NONE' and set up firewalling yourself. Otherwise, the firewall rules will deny access to all standard servers like Web, e-mail, ftp, etc. If you are using SSH, the rules will block outgoing SSH connections which allocate a privileged source port.

The firewall choices are:

- 0 - NONE: This script will not set any firewall rules. You are responsible for ensuring the security of your machine. You are STRONGLY recommended to use some kind of firewall rules.
- 1 - STANDALONE: Appropriate for a basic stand-alone web-surfing workstation
- 2 - MASQUERADE: Appropriate for a machine acting as an Internet gateway for a LAN

Choose a type of firewall (0-2): **«если Вы опытный пользователь, то решайте сами какой тип фаервола Вам подходит, или 1 (стандартный firewall), или 2 (для шлюза) иначе выберете тип 0».**

**** Summary of what you entered ****

Ethernet Interface: eth0

User name: «ВАШ ЛОГИН»

Activate-on-demand: No

Primary DNS: 217.170.80.83

Secondary DNS: 81.222.223.6

Firewalling: NONE

User Control: yes

Accept these settings and adjust configuration files (y/n)? y

Adjusting /etc/sysconfig/network-scripts/ifcfg-ppp0

Adjusting /etc/resolv.conf

(But first backing it up to /etc/resolv.conf.bak)

Adjusting /etc/ppp/chap-secrets and /etc/ppp/pap-secrets

(But first backing it up to /etc/ppp/chap-secrets.bak)

(But first backing it up to /etc/ppp/pap-secrets.bak)

Congratulations, it should be all set up!

Type `'/sbin/ifup ppp0'` to bring up your xDSL link and `'/sbin/ifdown ppp0'` to bring it down.

Type `'/sbin/adsl-status /etc/sysconfig/network-scripts/ifcfg-ppp0'` to see the link status.

СПИСОК КОНТРОЛЬНЫХ ВОПРОСОВ

1. Дайте определение маршрутизации.
2. Как настроить протокол TCP/IP для локальной сети.
3. Как настроить выход за пределы локальной сети?
4. Перечислите файлы конфигурации сети TCP/IP.
5. Какие типы маршрутизации Вы знаете?
6. В чем заключается настройка маршрутизации?
7. Каков порядок задания доменных имен и в каком файле он определяется?
8. Какова последовательность проверки работоспособности сети TCP/IP?

ЗАДАНИЯ ПО ЛАБОРАТОРНОЙ РАБОТЕ

1. Найдите все файлы, задающие настройки сети, перечисленные в лабораторной работе, и просмотрите их содержимое.
2. Отредактируйте файл */etc/hosts*, добавив в него машину своих соседей по лабораторной работе справа.
3. Попробуйте запустить команду *ifconfig* с разными ключами, проанализируйте результаты действия команды.
4. Попробуйте запустить команду *route* с разными ключами, проанализируйте результаты действия команды.
5. Найдите в Вашей системе файл, где записаны адреса DNS серверов.
6. Проверьте работоспособность сети, как описано в лабораторной работе.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Уэлш, М. Руководство по установке и использованию системы Linux. [Текст] / М. Уэлш [и др.]. – М.: ИЛКиРЛ, 1999.
2. Аленичев, Д. ALT Linux снаружи. ALT Linux внутри. [Текст] / Д. Аленичев, О. Власенко. – М.: ДМК пресс, 2006.
3. Собелл, М.Г. Практическое руководство по Red Hat Linux: Fedora Core и Red Hat Enterprise Linux [Текст] / Марк Г. Собелл. – 2-е изд. DVD-ROM. – М.: Вильямс, 2005.
4. Разработка приложений в среде Linux. Программирование для linux. [Текст] / Майкл К. Джонсон, Эрик В. Троан. – 2-е изд. – М.: Вильямс, Диалектика, 2007.
5. Руководство администратора Linux. Установка и настройка. [Текст] / Эви Немец, Гарт Снайдер, Трент Хейн. – 2-е изд. – М.: Вильямс, 2012.
6. Linux. Библия пользователя. [Текст] / Кристофер Негус. – СПб.: Питер, 2011.
7. Linux для чайников. [Текст] / Ди-Анн Лебланк. – 6-е изд. – М.: Диалектика, 2003.
8. Разработка ядра Linux. [Текст] / Роберт Лав. – 6-е изд. – М.: Вильямс, 2008.
9. Библиотека Qt 4. Программирование прикладных приложений в среде Linux. [Текст] / А.В. Чеботарев. – М.: Вильямс, 2006.
10. Red Hat Linux Fedora 4. Полное руководство. [Текст] / Пол Хадсон, Эндрю Хадсон [и др.] – М.: Вильямс, 2006.
11. Искусство программирования для Unix. [Текст] / Эрик С. Реймонд. – М.: Вильямс, 2005.
12. Red Hat Linux. Секреты профессионала. [Текст] / Наба Баркакати. – М.: Вильямс, 2004.
13. Использование Linux, Apache, MySQL и PHP для разработки Web-приложений. [Текст] / Джеймс Ли, Brent Уэр. – М.: Вильямс, 2004.
14. Секреты хакеров. Безопасность сетей – готовые решения. [Текст] / Стюарт Мак-Клар [и др.]. – 4-е изд. – М.: Вильямс, 2002.
15. FreeBSD: полный справочник. [Текст] / Родерик Смит. – М.: Вильямс, 2004.
16. Секреты хакеров. Безопасность Linux – готовые решения. [Текст] / Брайан Хатч [и др.] – 2-е изд. – М.: Вильямс, 2004.

17. Red Hat Linux 8. Библия пользователя. [Текст] / Кристофер Негус. – М.: Диалектика, 2003.
18. Серверы Linux. Самоучитель. [Текст] / К.А. Птицын. – М.: Диалектика, 2003.
19. Безопасность Linux. [Текст] / Скотт Манн [и др.]. – 2-е изд. – М.: Вильямс, 2005.
20. Сетевые средства Linux. [Текст] / Родерик Смит. – М.: Вильямс, 2006.
21. Руководство администратора Linux. [Текст] / Эви Немеет [и др.]. – М.: Вильямс, 2003.
22. Сети TCP/IP. Разработка приложений типа клиент/сервер для Linux/POSIX. [Текст] / Дуглас Камер, Дэвид Л. Стивенс. – Т. 3. – М.: Вильямс, 2002.
23. Программирование для Linux. Профессиональный подход. [Текст] / Марк Митчелл [и др.] – М.: Вильямс, 2003.
24. Использование Linux. Специальное издание. [Текст] / Дэвид Бендел, Роберт Нейпир. – 6-е изд. – М.: Вильямс, 2002.
25. Создание сетевых приложений в среде Linux. [Текст] / Шон Уолтон. – М.: Вильямс, 2001.
26. Освой самостоятельно Linux за 24 часа. [Текст]. – 3-е изд. – М.: Вильямс, 2001.
27. Система электронной почты на основе Linux. Руководство администратора. [Текст] / Ричард Блам. – М.: Вильямс, 2001.
28. Системное администрирование Linux. [Текст] / М. Карлинг [и др.]. – М.: Вильямс, 2000.