

Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий
Кафедра технологий программирования

Методические указания
к выполнению лабораторной работы №4

по дисциплине «**Компьютерные системы и сети**»
для специальности 1-40 01 01 Программное обеспечение информационных
технологий

на тему «**Настройка службы DHCP на Linux Ubuntu Server**»

Новополоцк, 2018 г.

Название: «Настройка службы DHCP на Linux Ubuntu Server».

Цель работы: Установить Linux Ubuntu Server. Изучить консольные команды работы с файлами и каталогами в Linux. Ознакомиться с сетевыми настройками ОС Linux Ubuntu. Настроить DHCP сервер.

Теоретическая часть

Установка операционной системы Linux

Установка Ubuntu Server начинается с выбора языка меню загрузки. Для установки рекомендуется выбрать English. В загрузочном меню выбираем «Install Ubuntu Server» – установка операционной системы заново. Затем появится окно выбора языка установки (рекомендуется выбрать English). Появляется меню выбора раскладки клавиатуры, которое позволяет выбрать раскладку либо автоматически (через выбор знакомых символов), либо выбрать из списка (рекомендуется USA).

Меню конфигурации сети предоставляет возможность настроить системы по умолчанию, вручную задать ip-адрес, маску подсети, шлюз, адрес DNS-сервера. Обязательными параметрами являются имя компьютера, имя домена.

Меню выбора диска для установки операционной системы предлагает выбрать:

- Авто – использовать весь диск.
- Авто – использовать весь диск и настроить LVM (менеджер логических томов).
- Авто – использовать весь диск с шифрованным LVM (менеджер логических томов).
- Вручную (Manual).

Выбираем ручную настройку.

Далее следует ввести полное имя нового пользователя, логин пользователя, пароль пользователя и повторение пароля. Отказываемся от выбора прокси и автоматического обновления. Устанавливаем загрузчик GRUB в главную загрузочную запись.

Файловая система Linux

С точки зрения пользователя, файловая система – это логическая структура каталогов и файлов. В отличие от Windows, где каждый логический диск хранит отдельное дерево каталогов, во всех UNIX-подобных системах эта древовидная структура растет из одного корня: она начинается с корневого каталога, родительского по отношению ко всем остальным, а физические файловые системы разного типа, находящиеся в разных разделах и даже на удаленных машинах, представляются как ветви этого дерева. В Linux используются следующие файловые системы:

ext – Расширенная файловая система, разработанная для ОС на ядре Linux.

ext2 – Второе расширение файловой системы, разработанной для ОС на ядре Linux.

ext3 – Так называется журналируемая версия ext2.

ext3cow – Форма файловой системы ext3 с поддержкой снапшотов (Снимок файловой системы или снапшот – моментальный снимок, копия файлов и директорий файловой системы на определённый момент времени.).

ext4 – Новая версия файловой системы ext3, с поддержкой экстенгов.

Имена файлов и каталогов

Имена файлов и каталогов могут иметь длину до 255 символов. Символы «/» (слэш) и символ с кодом 0 запрещены. Кроме того, ряд символов имеет специальное значение для командного интерпретатора, и их использование не рекомендуется. Это символы: ~!@#&%*()[]{}"'\;,>< N пробел. Если вам все-таки нужно употребить один из этих символов в имени файла, то при указании его имени в команде этот символ нужно экранировать знаком «\» (обратный слэш) или заключать все имя в двойные кавычки. Например, вы хотите вывести на экран командой cat содержимое текстового файла с именем my file, содержащим символ пробела:

```
$cat my file # неправильно  
$cat: my: No such file or directory  
$cat: file: No such file or directory  
$cat "my file" # правильно  
$cat my\file # правильно
```

Знак # – это символ комментария для командного интерпретатора. Командный интерпретатор игнорирует все символы от этого знака до конца введенной строки. Возможность комментировать не слишком полезна при вводе команд с консоли, но неоценима при написании командных сценариев. Заметьте, что точки среди специальных символов нет, и имена вроде this.is.a.text.file.containing.the.famous.string.hello.world допустимы и широко распространены.

Часто последняя отделенная точкой часть имени используется подобно «расширению имени» в Windows, обозначая файл определенного типа, но это обозначение несет смысл только для человека. Так, человеку имя файла ivan_home.tar.gz подсказывает, что это домашний каталог пользователя ivan, упакованный архиватором tar и сжатый компрессором gzip. Если имя файла начинается с точки, то этот файл считается скрытым: некоторые команды его «не видят». Например, введя в своем домашнем каталоге команду просмотра содержимого каталога ls с ключом -a, означающим «показывать скрытые файлы», вы увидите больше файлов, чем введя ту же команду без ключей.

Linux различает регистр символов в именах файлов: так, в одном каталоге могут находиться два разных файла README и Readme.

Имена каталогов строятся по точно тем же правилам, что и имена файлов.

Полным именем файла (или путем к файлу) называется список вложенных друг в друга каталогов, заканчивающийся собственно именем файла. Начинаться он может с любого каталога, потому что в древовидной структуре между любыми двумя узлами существует путь. Если этот список

начинается с корневого каталога, то путь называется абсолютным. Если с любого другого – то относительным (по отношению к этому каталогу). Корневой каталог обозначается символом «/» (слэш), и этим же символом разделяются имена каталогов в списке. Таким образом, абсолютным именем файла `README`, в домашнем каталоге пользователя **den** будет **`/home/den/README`**. В каждом каталоге существуют два особых «подкаталога» с именами «две точки» и «точка». Первый из них служит указанием на однозначно определенный родительский каталог, а второй – на сам данный каталог. Для корневого каталога, у которого нет родителя, оба эти «подкаталога» указывают на корневой каталог. С помощью этих имен образуются относительные имена файлов. Так, именем вышеупомянутого файла `README` относительно домашнего каталога **`/home/ivan`** пользователя **ivan** будет **`../den/README`**.

Назначение основных системных каталогов

В системных каталогах находятся файлы, необходимые для управления и сопровождения системы, а также стандартные программы. Их имена, расположение и содержание почти одинаковы почти во всех ОС Linux, поэтому эти каталоги называют также стандартными. Впрочем, на данный момент эпитет «стандартные» отражает скорее благие пожелания, чем действительность: иерархия каталогов одинакова только для дистрибутивов, связанных единством происхождения, а исторически сложившиеся различия создают опасность несовместимости разных дистрибутивов.

Краткое описание основных каталогов линии Red Hat и Fedora Core сведено в **Таблица 1** – Стандартные каталоги.

Таблица 1 – Стандартные каталоги

Каталог	Назначение
<code>/sbin</code>	Команды для системного администрирования, а также программы, выполняемые в ходе загрузки.
<code>/home</code>	Домашние каталоги пользователей, кроме root.
<code>/etc</code>	Файлы настроек: стартовые сценарии, конфигурационные файлы графической системы и различных приложений.
<code>/lost+found</code>	Восстановленные после аварийного размонтирования части файловой системы.
<code>/mnt</code>	Временные точки монтирования жестких дисков. Использовать этот каталог необязательно: подмонтировать файловую систему можно к любому другому каталогу.
<code>/proc</code>	Виртуальная файловая система, дающая доступ к информации ядра (например, выведите на экран файл <code>/proc/cpuinfo</code>). Другие файлы в этом каталоге в каждый момент времени содержат информацию о выполняющихся в этот момент программах.
<code>/tmp</code>	Временные файлы.
<code>/usr</code>	Практически все остальное: программы, исходные коды, документация. Сюда по умолчанию устанавливаются новые

Каталог	Назначение
	программы.
/bin	Основные программы, необходимые для работы в системе: командные оболочки, файловые утилиты и т.п.
/boot	Файлы, необходимые для загрузки системы (образ ядра).
/dev	Файлы устройств.
/lib	Системные библиотеки, необходимые для основных программ, и модули ядра.
/media	Сюда обычно монтируются съемные носители: компакт-диски, flash-накопители.
/opt	Дополнительные пакеты программ. Если программа, установленная сюда, больше не нужна, то достаточно удалить ее каталог без обычной процедуры деинсталляции.
/root	Домашний каталог суперпользователя. Домашние каталоги всех остальных могут находиться на отдельном разделе, но /root должен быть в корневой файловой системе, чтобы администратор всегда мог войти в систему для ремонтных работ.
/var	Часто меняющиеся данные: системные журналы и протоколы приложений, замки, почтовые ящики, очереди печати и т.п.

Типы файлов

С точки зрения UNIX-подобных ОС, файл представляет собой поток или последовательность байтов. Такой подход позволяет распространить понятие файла на множество ресурсов не только локального компьютера, но и удаленного, связанного с локальной сетью любого рода. Доступ к любому такому ресурсу осуществляется через универсальный интерфейс, благодаря чему запись данных в файл, отправка их на физическое устройство или обмен ими с другой работающей программой происходит аналогично. Это очень упрощает организацию данных и обмен ими. В ОС Linux можно выделить следующие типы файлов:

- обычные файлы – последовательность байтов (текстовые документы, исполняемые программы, библиотеки и т.п.);
- каталоги – именованные наборы ссылок на другие файлы;
- файлы физических устройств, подразделяющихся на файлы блочных устройств, драйверы которых буферизуют ввод-вывод с помощью ядра и файлы байт-ориентированных, или символьных, устройств, позволяющих связанным с ними драйверам выполнять буферизацию собственными средствами;
- символические ссылки (symlink, symbolic link);
- именованные каналы (named pipes);
- гнезда (sockets).

Команды работы с файлами и каталогами

Командой в ОС Linux называется все, что может быть выполнено: исполняемый файл, встроенная команда текущей программы и даже формируемая на ходу последовательность символов.

Синтаксис обычной команды:

имя_команды [короткие_ключи] [длинные_ключи] [аргументы]

где в квадратные скобки взяты необязательные данные. Ключи можно указывать в любом порядке, разделяя их пробелами. Аргументы тоже отделяются друг от друга пробелами.

Имена коротких ключей, или опций, состоят из одной буквы, перед которой стоит символ «-» (дефис). У ключа может быть свой аргумент. Короткие ключи, у которых нет аргументов, можно соединять под одним дефисом: так, команда

Is -I -a -d эквивалентна команде **Is -lad**.

Имена длинных ключей – это осмысленные слова, перед которыми стоят два символа «-» (дефис). Большинство команд понимают ключи **-help** и **-usage**, требующие вывести краткую справку об использовании команды. Необязательный ключ «-» сигнализирует об окончании списка ключей и начале аргументов.

Справку о ключах и аргументах команды можно получить по команде *man*.

Переход к учетной записи root

Команды, изменяющие системные каталоги или влияющие на работу системы, могут потребовать особые права. Для получения доступа к этим правам можно использовать суперпользователя (**root**). Для перехода к учетной записи root (суперпользователь) используется команда **sudo -i**. После ввода команды необходимо ввести пароль для входа в root. При выполнении одиночных команд от имени root, возможно воспользоваться конструкцией **sudo имя_команды аргументы_команды**, например:

sudo ifconfig eth0 192.168.1.1 netmask 255.255.255.0

Смена текущего каталога

Текущий каталог – это каталог, от которого отсчитываются относительные пути. В каждый момент времени с каждой работающей программой, в том числе с командной оболочкой, связан единственный такой каталог. Узнать, какой каталог сейчас является текущим, можно с помощью команды **pwd** без аргументов.

Сменить текущий каталог можно командой

cd [новый_каталог]

Для смены текущего каталога на домашний каталог пользователя можно вместо имени нового каталога указать специальный символ **~**: **cd ~**. Разрешается переходить также в псевдоподкаталоги «.» и «..». В первом случае ничего не изменится, а во втором текущим каталогом станет родительский.

Просмотр содержимого каталога

Команда **ls** *имя_каталога* выводит его содержимое на экран. Если не указывать имя каталога, команда выведет содержание текущего каталога.

Вместо имени каталога можно указать шаблон имен файлов: например,

ls *my** покажет все файлы и подкаталоги, имена которых начинаются с «my».

Подробнее об аргументах и ключах команды **ls** (как, впрочем, и любой другой команды) можно узнать у справочной системы, набрав команду **man ls**.

Создание и удаление файла и каталога

Создать пустой файл можно командой:

touch *имя_файла*

Вообще-то она предназначена для того, чтобы для всех заинтересованных программ (например, утилиты сборки проекта **make**) файл выглядел новее, чем на самом деле: она меняет время последнего изменения файла на текущее время. Но если файла с таким именем не существует, то она его создаст.

Текстовые файлы можно создавать, вводя текст с консоли:

\$cat > *hello.world*

Команду **cat** можно заставить не только выводить файл на консоль, но и вводить с нее. Это достигается перенаправлением ввода-вывода. В таком режиме команда **cat** считает своими входными данными поток байтов, поступающий с клавиатуры, и выводит его в указанный файл. Иными словами, в файл записывается все, что вы после этой команды введете с клавиатуры. Чтобы закончить ввод, нажмите **Ctrl+D**.

Если файл с указанным именем существует, то команда **cat** перепишет его. Чтобы вместо этого добавить данные в конец файла, перенаправьте ее вывод с помощью символов **>>**:

\$cat >> *hello.world*

Каталог создается командой **mkdir** <имя_каталога>.

Чтобы операция создания файла или подкаталога прошла успешно, вы должны иметь право записи в каталог, в котором вы его создаете.

Удалить пустой каталог можно командой **rmdir** <имя_каталога>.

Удаляется файл командой **rm** <имя_файла>.

При этом наличие прав на запись и даже чтение этого файла не требуется: достаточно иметь право на запись в каталог, где находится этот файл. Ключи команды **rm**:

-i: требует подтверждения удаления для каждого удаляемого файла. Подтвердите удаление каждого файла или откажитесь от него, введя символ **Y** или **N** соответственно;

-f: не запрашивать подтверждения, не выводить сообщений об ошибках. Если указаны оба ключа -i и -f, то срабатывает последний указанный;

-r: рекурсивное удаление каталога со всеми его подкаталогами. Не пустой каталог можно удалить только так.

Копирование и перемещение файла

Файл копируется командой **ср**. Формат этой команды:

ср [*ключи*] <исходный_файл> {<файл_назначения> | <каталог_назначения>}

Полезные ключи команды **ср**:

-i: требовать подтверждения перед перезаписью существующего файла;

-f: не требовать подтверждения;

-r: рекурсивно копировать каталог со всеми подкаталогами;

-a: сохранять атрибуты файла;

-d: копировать символические ссылки вместо файлов, на которые они указывают;

-s: создавать символические ссылки вместо копирования (-l – жесткие);

-u: не переписывать существующий файл, если он модифицирован позже;

-x: игнорировать подкаталоги, расположенные в других файловых системах.

Команда **mv** перемещает или переименовывает файлы. Формат команды:

mv [*ключи*] <исходный_файл> {<файл_назначения> | <каталог_назначения>}

Ключи **-i**, **-f** имеют тот же смысл, что для команд **ср** и **rm**.

Просмотр текстовых файлов

Перед тем, как просматривать файл, неплохо было бы убедиться, что он действительно является текстовым, то есть содержит только печатные ASCII-символы. Для проверки типа файла служит команда

file имя_файла

Если ее ответ содержит слово «text», то файл можно безопасно вывести на терминал. Вывод двоичного файла может сбить кодировку так, что вместо набираемых на клавиатуре символов вы увидите, черт знает что. Если это все же случилось, введите вслепую команду

\$ tput reset

Для вывода файла на терминал служат команда

more имя_файла

и ее улучшенный вариант **less**.

Команда-фильтр **more** разбивает поток своего вывода на порции размером в экран и ожидает ввода пользователя для отображения следующей порции. Чтобы увидеть следующую строку, нажмите <Ввод>; чтобы увидеть следующий экран, нажмите < пробел >; чтобы прервать работу команды, нажмите <Q> или Ctrl+C.

Утилита **less** позволяет листать выводимый файл не только вперед, но и назад (клавишами PgDn и PgUp), перемещаться к указанному месту в файле, искать по образцу и дает еще много полезных возможностей. В общем, **man less**.

Если интересующая вас информация находится в конце файла (например, вы хотите просмотреть журнал системных сообщений `messages` в каталоге `/var/log`, чтобы узнать, что именно только что пошло наперекосяк), то вам поможет команда

tail [-N] имя_файла,

где N – число выводимых строк файла, считая от последней.

Командой

head [-N] имя_файла

можно просмотреть, наоборот, только первые N его строк. Значение N по умолчанию равно 10.

Редактирование текстовых файлов

В качестве текстовых редакторов используются `vi` и `nano`. Формат вызова команд для редактирования файлов:

***vi* имя_файла**

***nano* имя_файла**

Для выполнения команд в редакторе **nano** используются сочетания клавиш `Ctrl` + клавиша.

Текстовый редактор **vi** существует уже очень давно. Он отличается от привычных графических редакторов, так как не обладает графическим интерфейсом, а общается с пользователем через систему команд.

Редактор при старте находится в режиме просмотра. Вход в режим редактирования осуществляется при выборе одной из команд. Возвращение в командный режим по клавише `<Esc>`. Для работы с редактором используется система команд.

РЕЖИМ ВСТАВКИ.

Включение режима вставки:

- `i` в текущей позиции курсора;
- `I` перед первым непробельным символом в текущей строке;
- `w` в новой строке, добавленной после текущей;
- `W` в новой строке, добавленной перед текущей.

Выключение режима вставки:

- `<Esc>`

Команды режима вставки:

- `Ctrl+a` повторить предыдущую вставку;
- `Ctrl+y` вставить символ, находящийся над курсором (в предыдущей строке);
- `Ctrl+e` вставить символ, находящийся под курсором (в следующей строке).

КОМАНДНЫЙ РЕЖИМ.

Удаление (здесь и далее N – это число):

- `N x N` символов под курсором и справа от него;
- `N X N` символов слева от курсора;
- `N dd N` строк;
- `D` до конца текущей строки;
- `N D` до конца текущей строки и еще N-1 строку.

Копирование и вставка строк:

- N уу взять в буфер N строк от текущей и ниже;
- р вставить содержимое буфера после текущей строки;
- Р вставить содержимое буфера перед текущей строкой.

Поиск и переход:

- N G перейти к строке с номером N;
- \$ G перейти к последней строке файла;
- /< образец > искать образец вниз от курсора;
- ?< образец > искать образец вверх от курсора;
- n повторить поиск в том же направлении;
- N (буквально «N»): повторить поиск в обратном направлении.

Сохранение и выход:

- :w сохранить текущий файл;
- :w <имя> сохранить под новым именем, если файл <имя> еще не существует;
- :w! <имя> сохранить под новым именем, переписав существующий файл;
- :q выйти;
- :q! принудительно выйти без сохранения;
- :wq сохранить и выйти.

Разное полезное:

- N и отменить последние N изменений;
- N Ctrl+г вернуть последние N отмененных изменений;
- U отменить изменения в последней строке;
- N г < символ > заменить N следующих символов на < символ >;
- N > > добавить отступ (Tab) в N следующих строк;
- N < < удалить один отступ (Tab) из N следующих строк;
- :sh временно выйти в оболочку (вернуться — exit);
- :!
<команда> выполнить команду оболочки.

Поиск файлов

Быстрый поиск имени программы можно выполнить прямо из командной строки: для этого введите первые буквы нужной вам команды и нажмите <Tab>. Если введенные вами буквы однозначно определяют команду или исполняемый файл, то ее имя появится в командной строке. Эта функция называется автозаполнением командной строки.

Если не появилось ничего, нажмите <Tab> еще раз для вывода списка всех доступных команд, начинающихся со введенных букв. Если таких команд окажется больше сотни, у вас попросят подтверждения того, что вы действительно хотите увидеть их все.

В разных каталогах может оказаться несколько исполняемых файлов с одинаковыми именами. Какой из них будет исполнен? На этот вопрос отвечает команда **which**. Она просматривает каталоги, перечисленные в переменной окружения PATH, в поисках исполняемого файла с указанным именем, и выводит абсолютное имя первого встреченного из них.

Команда **whereis** ищет не только исполняемый файл, но и его справочные страницы и исходный код.

Команда **locate** ищет файл по образцу имени, опираясь на свою базу данных о файловой системе. Ее вариант с повышенной безопасностью **slocate** сохраняет данные о правах доступа к файлам, так что пользователь не увидит тех файлов, на которые у него нет прав.

Команда **find** ищет файл по его атрибутам в указанном каталоге и его подкаталогах на заданную глубину. Например:

```
find /media -name rc* /media/cdrecorder/Fedora/RPMS/mc-4.6.1-0.8.i386.rpm
```

Эта команда осуществит поиск **mc-4.6.1-0.8.i386.rpm** на компакт диске.

Изменение прав доступа к файлу

Как многопользовательская операционная система, ОС Linux содержит механизм разграничения доступа к данным, позволяющий как защитить данные одного пользователя от нежелательного вмешательства других, так и разрешить другим доступ к этим данным для совместной работы.

Как уже сказано, любой ресурс компьютера под управлением ОС Linux представляется как файл, поэтому мы будем говорить только о правах доступа к файлу.

По отношению к файлу пользователь может входить в одну из трех категорий: владелец, член группы владельца, все остальные. Для каждой из этих категорий есть свой набор прав доступа.

Первым владельцем файла становится его создатель. Далее файл можно передать другому владельцу или в другую группу командой

```
chown ключи новый_пользователь:новая_группа файл  
или
```

```
chgrp ключи новая_группа файл
```

В некоторых реализациях Linux передать файл другому владельцу имеет право только суперпользователь, а в других — также его текущий владелец.

Набор прав доступа состоит из прав на чтение, запись и исполнение файла. В символьном представлении он выглядит как строка «гwx», где вместо любого символа может стоять дефис. Буква означает наличие права (r — чтение, w — запись, x — исполнение), дефис — его отсутствие.

Очевидно, что эти три бита могут быть записаны еще и как восьмеричное число. Так, права доступа r-x (чтение и исполнение без записи) понимаются как три двоичные цифры 101 или как восьмеричная цифра 5. Численное представление прав доступа называется абсолютным, или двоичной маской.

Полная строка прав доступа в символьном представлении устроена так: **<права_владельцаправа_группныхправа_остальных>**.

В абсолютном представлении права владельца являются старшим разрядом восьмеричного числа, права группы — вторым и права остальных — третьим. Так, права гwxг-x-x выглядят как число 111101 001, или 751.

Команда изменения прав доступа **chmod** понимает, как абсолютное, так и символическое указание прав.

Назначим файлу /home/den/README права rw-r, то есть разрешим себе чтение и запись, группе только чтение, остальным пользователям — ничего:

```
$cd ~ # переход в свой домашний каталог
$chmod 640 README # 110 100 000 == 640
$ls -l README
-rw-r--r-- 1 den users 0 Feb 14 19:08 /home/den/README
```

В символическом представлении можно явно указывать, кому какое право мы хотим добавить, отнять или присвоить. Добавим право на исполнение файла README группе и всем остальным:

```
$chmod go+x README
$ls -l README
-rw-r-x--x 1 den users 0 Feb 14 19:08 /home/den/README
```

Формат символического режима:

chmod <категория действий набор_прав> <файл >

Команды работы с сетевыми интерфейсами и сетью

В Linux есть целый ряд команд, позволяющих осуществлять поиск неисправностей в сети, конфигурировать сетевые интерфейсы и выполнять задачи, связанные с маршрутизацией. Наиболее важные из них рассматриваются в данной главе: ifconfig, route, ping, traceroute, arp, netstat. Все они входят в большинство дистрибутивов Linux

Команда ifconfig

Команда **ifconfig** обслуживает сетевые интерфейсы на нижних уровнях модели OSI, позволяя задавать тип передающей среды, скорость работы интерфейса и сетевые адреса.

Обычно команда находится в каталоге /sbin, и запускать ее могут непривилегированные пользователи. Но менять конфигурацию интерфейсов разрешено только суперпользователю(root).

Функции, выполняемые командой **ifconfig**, можно разделить на две группы: выдача статистики работы интерфейсов и конфигурирование интерфейсов.

Отчет о состоянии интерфейсов. Команда **ifconfig** часто используется при поиске неисправностей в сети, так как позволяет получить отчет о состоянии сетевых интерфейсов:

Ifconfig -a

Получение статистики по интерфейсу eth0:

Ifconfig eth0

Конфигурирование сетевых интерфейсов

Команда **ifconfig** позволяет менять параметры сетевых интерфейсов. Она обычно выполняется во время начальной загрузки системы, но может применяться и для динамического изменения конфигурации сети. Такие изменения будут иметь силу до того момента, пока не произойдет перезагрузка системы. Чтобы установки сохранялись постоянно,

соответствующие команды нужно поместить в сценарии начальной загрузки (этот процесс здесь не показан). Будьте осторожны: в случае ошибки компьютер может оказаться отключенным от сети. Стандартные имена интерфейсов в Ubuntu начинаются на **eth** и заканчиваются номером (например, eth0, eth1).

Подключение и отключение интерфейсов. Чаще всего, команда **ifconfig** применяется для активизации и отключения интерфейсов:

Отключение интерфейса eth0:

ifconfig eth0 down

Включение интерфейса eth0:

ifconfig eth0 up

С помощью команды **ifconfig** можно назначить интерфейсу новый IP-адрес, например:

ifconfig eth0 192.168.1.253

Разрешается также менять сетевую маску:

ifconfig eth0 192.168.1.253 netmask 255.255.255.248

При перезагрузке ОС все параметры, выставленные данной командой, пропадут, настройка сети будет производиться согласно начальной сетевой конфигурации (17).

Команда route

Команда **route** предназначена для доступа к маршрутным таблицам ядра. Она может работать в трех режимах: отображение, добавление и удаление. Будучи вызванной без параметров, команда **route** отображает содержимое таблицы маршрутизации.

Возможные флаги перечислены в **Таблица 2** – Флаги команды **route**.

Таблица 2 – Флаги команды **route**

Флаг	Описание
!	Маршруты, полученные из данной сети, отвергаются
A	Маршрут, добавленный протоколом конфигурирования адресов IPv6
C	Кэшированная запись
D	Маршрут, созданный демоном или в результате переадресующего ICMP-запроса
G	Трафик, идущий по данному маршруту, проходит через ШЛЮЗ, указанный в столбце Gateway
M	Маршрут, модифицированный демоном или В результате переадресующего ICMP-запроса
u	Маршрут, созданный и используемый ядром

Команда **route** без аргументов эквивалентна команде **netstat -r**

Добавление маршрутов. Основное достоинство команды **route** заключается в возможности добавления статических маршрутов в таблицу ядра. Команда, добавляющая маршрут, имеет такой формат:

route add -net адрес-получателя netmask маска gw ШЛЮЗ.

Команда **ping**

Команда **ping** сообщает о том, доступен ли заданный узел. Она посылает эхо-запрос ICMP удаленному узлу, который должен вернуть эхо-ответ ICMP. Поскольку с помощью данной команды можно легко выяснить структуру сети, многие сетевые фильтры и брандмауэры удаляют ICMP-пакеты в целях безопасности.

С помощью опции **-d** можно получить дополнительную отладочную информацию, выдаваемую командой **ping**. Будут, например, показываться ICMP-сообщения о недоступности сети.

Опцию **-f** нужно использовать осторожно, так как она заставляет команду **ping** посылать столько эхо-запросов, сколько позволяет пропускная способность сети. С помощью этой опции можно имитировать состояние перегруженности сети.

Обычно команда **ping** ждет одну секунду, прежде чем посылать следующий эхо-запрос. Опция **-i** позволяет задать больший интервал ожидания (в секундах).

Посредством опции **-s** задается размер ICMP-пакетов, который по умолчанию составляет 64 байта.

Опция **-I** интерфейс заставляет команду **ping** посылать эхо-запросы через указанный интерфейс. Это полезно, если в системе имеется несколько интерфейсов и, таким образом, существует несколько маршрутов к получателю.

Команда **traceroute**

Команда **traceroute** чрезвычайно широко применяется при поиске неисправностей в сетях. Она показывает путь от исходного узла (Linux-машины, на которой была введена команда) к адресату.

Алгоритм работы команды таков. Она посылает по указанному адресу UDP-пакет, время жизни которого (поле TTL) равно 1. Первый же маршрутизатор, получивший такой пакет, уменьшит значение TTL на 1 и обнаружит, что пакет устарел. Тогда он вернет отправителю ICMP-сообщение *time exceeded* и удалит пакет. Суть в том, что в ICMP-сообщении при водится адрес маршрутизатора. Команда **traceroute** отображает этот адрес и формирует следующий пакет, в котором поле TTL равно уже 2.

Первый маршрутизатор уменьшает значение TTL на единицу и пересылает пакет дальше. Второй маршрутизатор удаляет пакет и генерирует описанное выше ICMP-сообщение. Так продолжается до тех пор, пока не будет достигнут заданный адресат. При этом поле TTL будет равно числу переходов.

Опция **-d** включает отображение дополнительной отладочной информации.

Опция **-f** **t**tl заставляет команду **traceroute** начать отсчет поля TTL с указанного значения, а не с единицы. Это может понадобиться, если брандмауэр не отвечает на пакеты команды **traceroute**, но пропускает их во внешний мир.

Опция **-F** запрещает осуществлять фрагментацию дейтаграмм (пакеты отправляются с установленным флагом DF). Маршрутизатор, получивший

такой пакет, удалит его, если размер пакета окажется больше, чем параметр MTU сетевого интерфейса, через который он должен быть переслан

Команда netstat

Команда **netstat** отображает информацию о состоянии сетевого программного обеспечения на локальном узле. Чаще всего она используется с аргументом **-r**, задающим отображение таблицы маршрутов.

Опция -a заставляет команду netstat отобразить информацию обо всех сокетах, открытых на локальном узле. Учитываются также все демоны, прослушивающие сетевые порты.

Настройка сетевых параметров

Имя машины

Имя машины содержится в двух файлах: **/etc/hosts** и **/etc/hostname**.

/etc/hosts — текстовый файл, содержащий базу данных доменных имен и используемый при их трансляции в сетевые адреса узлов. Запрос к этому файлу имеет приоритет перед обращением к DNS-серверам. В отличие от DNS, содержимое файла контролируется администратором компьютера.

На любой Linux системе имя компьютера можно изменить с помощью команды **hostname**. Команда, вызванная без аргументов, выведет на экран текущее имя системы:

hostname

Чтобы изменить имя компьютера, достаточно использовать ту же команду, но с аргументом:

hostname имя_компьютера

Новое имя компьютера будет установлено сразу после выполнения и будет активно вплоть до очередной перезагрузки системы. Чтобы новое имя стало постоянным, необходимо изменить конфигурационный файл **/etc/hostname**.

Система, при загрузке читает файл **/etc/hostname**, чтобы установить имя компьютера. После чего, прочитанное имя устанавливается во время работы init-скрипта **/etc/init.d/hostname** (или **/etc/init.d/hostname.sh**).

Чтобы задать постоянное имя компьютеру, необходимо отредактировать этот файл, указав новое имя машины. Затем необходимо перезапустить службу **hostname**:

start hostname

Либо просто перезагрузить машину. Кроме этого, необходимо проверить файл **/etc/hosts** на предмет упоминания в нем старого имени компьютера. Если оно там встречается (а оно там наверняка встречается), то его так же необходимо заменить на новое имя.

После этого необходимо перезапустить службу сети:

sudo /etc/init.d/networking restart

Либо, опять же, просто перезагрузить компьютер.

Настройка DNS-клиента

Для организации работы Linux-сервера в качестве клиента системы DNS понадобятся три файла. Это файлы */etc/resolv.conf*, */etc/hosts* и */etc/host.conf*.

Файл */etc/resolv.conf* используется для конфигурации DNS-клиента на машине. Параметр *nameserver* определяет какому DNS-серверу должны направляться DNS-запросы. Можно указать до трех DNS-серверов. Второй и третий серверы из списка будут использоваться в качестве резервных, если ответ не будет получен от первого (главного) сервера. Если в вашей сети имеется локальный DNS-сервер, то его нужно использовать в качестве главного. Теперь, если нужно получить IP-адрес для хоста *fred.smallorg.org*, можно просто указать имя хоста *fred*, и ОС Linux будет автоматически добавлять к нему имя домена *smallorg.org*. К сожалению, это может работать и против вас. Дело в том, что система DNS будет добавлять имя домена *smallorg.org* ко всем хостам, для которых она производит преобразование имен. Так, если осуществляется соединение с узлом *www.linux.org*, сначала ведется поиск узла *www.linux.org.smallorg.org*. Если найти его не удалось, то производится поиск уже *www.linux.org*. Пример файла */etc/resolv.conf* для клиента DNS на базе ОС Linux.

```
search smallorg.org  
nameserver 10.0.0.1  
nameserver 10.0.0.2  
nameserver 10.0.0.3
```

В строке 1 задается поиск домена по умолчанию, который следует использовать во всех DNS-запросах. Помните, что это может замедлить обработку запросов для хостов, не принадлежащих к вашему домену, так как текст по команде *search* будет добавляться ко всем запросам. В строках 2–4 указаны главный, вторичный и третичный DNS-серверы, которые обслуживают данного клиента.

Еще один метод преобразования имен заключается в использовании локальной базы данных имен хостов, подобно тому, как это делалось на заре сети Internet. Файл */etc/hosts* содержит список имен хостов с соответствующими IP-адресами.

Во всех компьютерах с ОС Linux осуществляется поддержка имени *localhost*. Этому имени всегда соответствует IP-адрес 127.0.0.1, который присваивается специальному сетевому устройству типа «петля». Эти имя и адрес позволяют внутренним процессам взаимодействовать с другими процессами в этой же системе по сетевым протоколам. Многие программы даже сконфигурированы на использование имени *localhost*. Изменение этого имени или соответствующего ему IP-адреса может повлиять на работу этих программ.

В файле */etc/host.conf* определяются методы и порядок преобразования имен ОС Linux. Пример файла */etc/host.conf*.

```
order hosts, bind  
multi on
```


При получении параметров от DHCP-сервера, файл `/etc/resolv.conf` переписывается с учетом DNS-суффикса и адреса DNS-сервера.

Настройка сетевых интерфейсов

Настройка сетевых интерфейсов системы производится в файле `/etc/network/interfaces`.

По умолчанию интерфейсы неактивированы. Для активации интерфейса необходимо добавить в конфигурацию следующую запись:

auto имя_интерфейса

Тип ip-адреса интерфейса описывается параметрами:

iface имя_интерфейса ***inet*** тип_адреса

где *тип_адреса* может принимать значения ***dhcp*** (получает ip-адрес с DHCP-сервера) и ***static*** (статический ip-адрес). Для типа ***static*** необходимо указать ip-адрес:

address ip_адрес

Маска подсети описывается строкой:

netmask маска_подсети

Адрес широковещательной рассылки описывается строкой:

broadcast ip_адрес_широковещательной_рассылки

Адрес шлюза (маршрутизатора) описывается строкой:

gateway ip_адрес_шлюза

Адрес DNS-сервера описывается строкой:

dns-nameservers ip_адрес_dns_сервера

По умолчанию в данном файле содержится запись об аппаратной петле:

auto lo

iface lo inet loopback

Аппаратная петля (localhost) (127.0.0.1 — 127.255.255.255) — зарезервированный диапазон IP-адресов для обозначения так называемого «локального хоста», то есть для сети, состоящей только из одного компьютера. Как правило, используется всего один адрес — 127.0.0.1, который устанавливается на специальный сетевой интерфейс «внутренней петли» (англ. loopback) в сетевом протоколе TCP/IP. В Unix-подобных системах данный интерфейс обычно именуется «loN», где N — число, либо просто «lo». При установке соединений в этой вырожденной «сети» присутствует только один компьютер, при этом сетевые протоколы выполняют функции протоколов межпроцессного взаимодействия.

Использование адреса 127.0.0.1 позволяет устанавливать соединение и передавать информацию для программ-серверов, работающих на том же компьютере, что и программа-клиент, независимо от конфигурации аппаратных сетевых средств компьютера (не требуется сетевая карта, модем, и прочее коммуникационное оборудование, интерфейс реализуется при помощи драйвера псевдоустройства в ядре операционной системы). Таким образом, для работы клиент-серверных приложений на одном компьютере не требуется изобретать дополнительные протоколы и дописывать программные модули.

Обычно адресу 127.0.0.1 однозначно сопоставляется имя хоста «localhost» и/или «localhost.localdomain».

Перезапуск сетевых интерфейсов производится командой
/etc/init.d/networking restart

По умолчанию установщик конфигурирует сетевые интерфейсы на получение настроек по DHCP.

Ход работы

Перед началом установки пакетов необходимо указать репозиторий. Чтобы иметь возможность установить пакеты с диска необходимо, отредактировать файл */etc/apt/sources.list* и указать путь к репозиторию с пакетами для установки. По умолчанию в этом файле стоит комментарий перед ссылкой на cdrom с исходниками.

Строка, указывающая путь к репозиторию на диске начинается с:

deb cdrom:.....

Установка и настройка DHCP в Ubuntu 11.04

DHCP (англ. Dynamic Host Configuration Protocol – протокол динамической конфигурации узла) – это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Для этого компьютер обращается к специальному серверу, называемому сервером DHCP. Сетевой администратор может задать диапазон адресов, распределяемых среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве крупных сетей TCP/IP.

Установка DHCP сервера

Для запуска DHCP сервера необходимо установить службу *isc-dhcp-server*, находящуюся на установочном диске Ubuntu. Команда установки выглядит следующим образом:

apt-get install isc-dhcp-server

Запуск службы DHCP-сервера выполняется командой:

/etc/init.d/isc-dhcp-server start

Остановка службы DHCP-сервера выполняется командой:

/etc/init.d/isc-dhcp-server stop

Перезапуск службы DHCP-сервера выполняется командой:

/etc/init.d/isc-dhcp-server restart

Настройка DHCP сервера

Настройка DHCP-сервера производится путём редактирования двух файлов:

/etc/default/isc-dhcp-server и ***/etc/dhcp/dhcpd.conf***.

Файл ***/etc/default/isc-dhcp-server*** определяет с каких интерфейсов будет выполняться раздача сетевых настроек. Список интерфейсов описывается строчкой:

INTERFACES="eth0 eth1" – раздача с двух сетевых интерфейсов **eth0** и **eth1**.

Файл ***/etc/dhcp/dhcpd.conf*** определяет сетевые настройки, которые будут раздаваться клиентам.

Для описания подсети используется конструкция

subnet адрес_подсети netmask маска_подсети

{

параметры;

}

Наиболее широко используемые параметры:

range начальный_адрес_пула конечный_адрес_пула;

option netbios-name-servers ip_адреса_NetBios_серверов;

option domain-name-servers ip_адреса_DNS_серверов;

option domain-name "имя_домена";

option broadcast-address адрес_широковещательных_запросов;

option routers ip_адрес_шлюза;

option subnet-mask маска_подсети;

При указании нескольких подсетей необходимо в каждой подсети указать параметр ***subnet-mask***.

Для указания статического ip-адреса определенному компьютеру используется следующая запись:

host bla3 {

hardware ethernet mac_адрес;

fixed-address ip_адрес;

}

Примечание: Для запуска службы DHCP-сервера необходимо убедиться в указании ip-адреса соответствующему сетевому интерфейсу.

Практическая часть

Содержание задания

В рамках задания требуется установить Ubuntu Server и настроить на получение статического ip-адреса. Установить и настроить DHCP-сервер. Настроить клиент на динамическое получение сетевых адресов и сетевых параметров. По результатам работы составить отчет.

Порядок выполнения работы

- 1 Ознакомиться с теоретической частью.
- 2 Установить Ubuntu Server.
- 3 Настроить конфигурацию сетевых интерфейсов сервера на получение статического ip-адреса (*Значение последнего октета ip-адреса должно соответствовать номеру студента в списке группы*).
- 4 Установить и настроить DHCP-сервер (*Пул адресов не должен превышать десять значений, и начинаться с цифр равных последним двум цифрам в вашем студенческом билете*).
- 5 Настроить клиент на динамическое получение сетевых адресов и сетевых параметров.
- 6 Проверить функционирование сети.
- 7 Составить отчёт о проделанной работе.
- 8 Показать выполненную работу и отчёт преподавателю.

Содержание отчёта

- 1 Титульный лист.
 - 2 Цель работы.
 - 3 Краткие теоретические сведения.
 - 4 Основные результаты по каждому пункту хода выполнения работы.
 - 5 Описание проверок функционирования служб со скриншотами результатов.
 - 6 Выводы о проделанной работе.
- Защита работ проводится индивидуально.