

Министерство образования и науки Российской
Федерации
Кубанский государственный университет

Факультет математики и компьютерных наук

Кафедра **информационных образовательных технологий**
Кафедра **вычислительной математики** и информатики

Алексеев Е.Р., Грушевский С.П.,
Дога К.В., Обухова Ю.Н.

Учебное пособие «Основы работы в Linux»

Краснодар, 2023

УДК
ББК
А

Рецензенты:

Доктор физико-математических наук, профессор, профессор
Санкт-Петербургского Политехнического Университета

А.Н. Васильев

Кандидат технических наук, доцент Вятского
Государственного Университета

В.Б. Чернявский

Алексеев Е.Р., Грушевский С.П., Дога К.В., Обухова Ю.Н. Основы работы в Linux: учеб. пособие / Е.Р. Алексеев, С.П. Грушевский, К.В. Дога, Ю.Н. Обухова. – Краснодар: Кубанский гос. ун-т, 2023. –165 с. – 50 экз.

ISBN

Содержит основные сведения, необходимые для освоения операционной системы семейства Linux. Пособие состоит из двух частей. В первой части авторы знакомят читателя с базовыми понятиями операционных систем, построенных на ядре Linux. Вторая часть посвящена сетевым возможностям ОС Linux. Пособие содержит тексты одиннадцати лабораторных работ.

Адресуется студентам математических, физико-технических, ИТ-специальностей.

УДК
ББК ISBN

© Кубанский
государственный университет, 2023
© Е.Р. Алексеев, 2023

Оглавление

Введение	4
Работа на компьютере под управлением ОС Линукс	6
1 Свободное программное обеспечение	8
2 Знакомство с операционной системой Linux	10
3 Файловая система ОС Linux	13
4 Терминал Linux и основные команды	16
5 Создание нового пользователя. Изменение прав доступа	26
6 Пакеты в ОС Linux	35
7 Жёсткие и символичные ссылки	39
8 Особенности настройки АльтЛинукс после установки	43
9 Установка Linux на компьютер пользователя	55

10 Лабораторный практикум «Основы работы в Linux»	70
Сетевые возможности ОС Linux	82
11 Общие сведения о сетевых протоколах	84
12 Общий доступ к каталогам с помощью Samba	88
13 Лабораторная работа №9. Локальная сеть. Общий доступ к каталогу. Работа с програм- мой Samba	99
14 Клиент и сервер SSH	103
15 Лабораторная работа №10. Протокол SSH	123
16 Настройка и использование FTP-сервера в Linux	127
17 Лабораторная работа №11. FTP-сервер	158
Литература	160

Введение

Данное пособие написано для студентов первого и второго курсов факультета математики и компьютерных наук Кубанского Государственного университета.

Цель этой работы - познакомить читателя с фундаментальными понятиями в области операционных систем (ОС), построенных на базе ядра Linux. Авторы надеются, что у читателя есть большое желание изучить операционные системы, которые являются альтернативой проприетарной ОС, доминирующей на рынке персональных компьютеров. В пособии нет привязки к конкретному дистрибутиву и подробного описания графического интерфейса. Большое внимание уделяется базовым навыкам, которыми должен овладеть пользователь для дальнейшего самостоятельного изучения операционных систем. В доступной форме разъясняются такие понятия, как операционная система, дистрибутив, права доступа к файлу, жёсткая ссылка и т.д.

Пособие состоит из двух частей. В первой читатель знакомится с базовыми понятиями операционных систем, построенных на базе ядра Linux. Во второй - с сетевыми возможностями ОС Linux. Для отработки практических навыков приводятся тексты одиннадцати лабораторных работ.

В первой главе читатель познакомится с двумя моделями разработки программного обеспечения: свободной и проприетарной. Особое внимание уделяется описанию принципов свободного программного обеспечения. Вторая глава содержит информацию об операционных системах, дистрибутивах. Последующие несколько глав (3-6) содержат фундамент для

освоения Linux на уровне продвинутого пользователя (права доступа к файлам, дерево каталогов, роль администратора в системе, основные команды терминала, необходимые для управления файловой системой компьютера, работающего под управлением unix-подобных ОС). В седьмой главе читатель познакомится с такими важными понятиями, как жёсткие и символьные ссылки. Отдельная глава посвящена особенностям работы в операционных системах семейства АЛЪТ. Их настройка отличается и от дистрибутивов на базе Debian, и от дистрибутивов на базе rpm пакетов. В девятой главе излагаются общие принципы установки операционных систем семейства Linux на компьютер пользователя. Десятая глава содержит задания лабораторного практикума, который выполняют студенты для практического освоения операционных систем семейства Linux. Во второй части пособия читатель знакомится с работой в локальной сети под управлением ОС Linux. Одиннадцатая и двенадцатая главы посвящены организации общего доступа к файлам и каталогам в локальной сети по протоколу Samba. В тринадцатой главе приведён текст лабораторной работы. Четырнадцатая и пятнадцатая главы содержат практические сведения, необходимые для организации взаимодействия в Сети с помощью протокола SSH. Последние две главы содержат информацию для организации взаимодействия в сети по протоколу FTP.

Лабораторные работы, приведенные в пособии, разрабатывались и апробировались более десяти лет на базе Донецкого национального Технического Университета (г. Донецк, ДНР), Вятского Государственного Университета (г. Киров), Кубанского Государственного Университета (г. Краснодар).

Авторы надеются, что пособие будет востребовано за пределами Кубанского Государственного Университета.

Россия, Краснодар, апрель 2023 г.

Работа на компьютере под управлением ОС Линукс

Глава 1

Свободное программное обеспечение

Современное программное обеспечение (ПО) по методам разработки и идеологии использования можно разделить на два больших класса: **свободное** и **проприетарное**.

Программное обеспечение называется **свободным**, если оно удовлетворяет следующим четырём условиям свободы.

Свобода 0. Вы можете запускать свободное программное обеспечение на своём (и не только на своём) компьютере. Возможность запускать программу не зависит от того, как Вы её получили.

Свобода 1. Вы можете распространять свободное программное обеспечение (делиться программами с друзьями, коллегами, знакомыми).

Свобода 2. Вы можете изменять свободное программное обеспечение.

Свобода 3. Вы можете распространять модифицированное Вами свободное программное обеспечение.

Чтобы Вы могли изменять свободное программное обеспечение, должен быть доступен исходный код программ. Доступность исходного кода является самым важным свойством свободного программного обеспечения.

Обратите внимание, критерии свободы программного обеспечения никаким образом не затрагивают вопросы стоимости компьютерных программ или авторских прав.

Зачастую свободные программы являются бесплатными, но это не является обязательным требованием.

Автором программы является её разработчик (в большинстве случаев).

Программное обеспечение, которое не удовлетворяет всем четырём свободам, называется **проприетарным**.

Свобода программного обеспечения открывает новые возможности для его разработки, модификации и тестирования.

Можно выделить следующие широкоизвестные свободные программы.

1. Операционные системы семейства **Linux**.
2. Браузер **Mozilla Firefox**.
3. Интерпретатор языка **Python** 3.
4. Офисный пакет **LibreOffice**.
5. Математические пакеты **Scilab**, **GNU Octave**, **Maxima**.

Глава 2

Знакомство с операционной системой Linux

Начнём с базового понятия в данной теме – понятие операционной системы.

Операционная система (ОС) – комплекс взаимосвязанных программ, предназначенных для работы с вычислительным устройством и организации взаимодействия с пользователем.

В основе понятия ОС лежит понятие ядра. Ядро – это низкоуровневая программа, которая управляет физическими ресурсами вычислительного устройства и другими программами.

В отличие от Windows, Linux – это не конкретная операционная система, которую можно установить на устройство. Linux – это ядро операционной системы, на базе которого создаются различные дистрибутивы.

Дистрибутив Linux – это система комплектации ядра ОС с комплексом утилит и прикладных программ.

Современные дистрибутивы:

- Debian, Ubuntu, Mint;
- RedHat, Fedora, OpenSUSE, CentOS;
- Arch, Manjaro, Gentoo;

- Русские дистрибутивы Linux (ROSA, Calculate, Astra, AltLinux).

Кроме разделения дистрибутивов по производителям, существует деление по типам использования:

1. LiveCD – дистрибутив, при использовании которого загрузка операционной системы (рабочего стола) происходит с CD (или USB). После этого с операционной системой можно работать, как с ОС, запущенной с жесткого диска (винчестера).
2. Установочные дистрибутивы – дистрибутивы, с помощью которых осуществляется установка ОС на жесткий диск.

Термин «дистрибутив» применим не только к операционным системам, речь может идти и о прикладном программном обеспечении (ПО).

Дистрибутив ПО – это набор файлов, который включает в себя вспомогательные инструменты для автоматической или автоматизированной (частично автоматической) начальной настройки ПО. Другими словами, установщик.

При использовании дистрибутива все необходимые файлы устанавливаются таким образом, чтобы их правильно видела операционная система. Кроме того, конфигурируются начальные параметры, язык, способ подключения к Интернету и т. д.

В ОС Linux дистрибутивы программ называются пакетами, которые бывают нескольких видов. Самые популярные из них:

1. deb (в Debian, Ubuntu, Linux Mint и др.)
2. rpm (openSUSE, CentOS, Mageia и др.)

Desktop enviroment (DE) - это окружение рабочего стола, в которое входит:

- Графическая подсистема;
- Оконный менеджер;
- Файловый менеджер;
- Эмулятор терминала;
- Меню запуска приложений;
- Браузер и т. д.

В Windows существует всего одно окружение рабочего стола. Оно стандартно, пользователь видит его сразу после установки ОС и изменить не может. Однако, один и тот же дистрибутив Linux может иметь несколько вариаций с разным графическим окружением, например:

1. KDE 5 (Plasma).
2. GNOME 3 (MATE, CINAMON).
3. XFCE.
4. LXDE.
5. LXQT.
6. Openbox.

Сейчас существует множество различных дистрибутивов Linux, между которыми много общего. Чаще всего поправки (в основном графические) вносятся в инсталляцию, изменяется список включаемого ПО. При этом ядро, как правило, остается без изменений. В качестве графической оболочки почти везде используется KDE или GNOME. В случае отсутствия, их всегда можно установить, и тогда вне зависимости от основного дистрибутива, у всех будет одинаковый графический интерфейс.

Глава 3

Файловая система ОС Linux

Основные понятия в файловой системе ОС Linux – файл и каталог.

Файл – поименованная область данных, расположенная на внешнем носителе. Исходя из определения, файл характеризуется содержимым, именем и месторасположением.

Имя файла может иметь расширение. Большинство графических интерфейсов связывают программу обработки файла данного типа с его расширением.

Каталог - поименованный перечень файлов и подкаталогов (вложенных каталогов). Он позволяет облегчить поиск данных и систематизировать их на устройстве. В большинстве современных файловых систем используется иерархическая структура. В любой файловой системе Linux существует единственный общий для всех файлов каталог - корневой. Обозначается "/" ("слэш").

В нём могут содержаться не только файлы, но и подкаталоги первого уровня вложенности. В каталогах первого уровня вложенности тоже могут существовать файлы и подкаталоги (по отношению к корневому, это будут каталоги второго уровня вложенности) и так далее. Эта система называется деревом каталогов.

Положение любого файла в этой системе описывается при помощи полного пути, который начинается от «/» (кор-

нового каталога), а затем через разделитель «/» перечисляются все подкаталоги согласно порядку вложенности. Заканчивается полный путь названием искомого файла. Например, `/home/user/pas/abc.pas` – файл `abc.pas`¹, который находится в каталоге `/home/user/pas`².

Древо устроено так, что среди подкаталогов конкретного каталога не может находиться он сам. Придумано это для того, чтобы избежать заикливания. Благодаря такому принципу путь к файлу или каталогу определяется однозначно.

Текущий каталог – это каталог, с которым пользователь (или программа) работает в настоящий момент времени. С текущего каталога начинается относительный путь к файлу, и в этом существенное его отличие от полного пути.

Кроме того, в Linux у каждого пользователя обязательно есть свой собственный каталог, который становится текущим сразу после регистрации в системе. Он называется домашним и предназначен для хранения пользовательских данных.

Родительский каталог – такой каталог, в котором содержится данный. Для корневого каталога родительским является он сам.

Обязательные подкаталоги корневого (см. рис. 3.1):

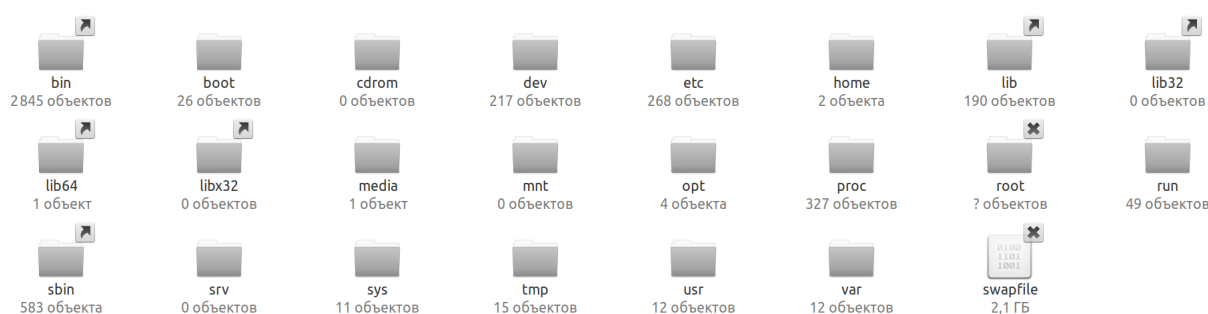


Рис. 3.1: Содержимое корневого каталога

¹.**pas** – расширение файлов программ на языке Pascal.

²`/home/user` – домашний каталог пользователя `user`.

- `/bin` и `/sbin` - каталог для хранения исполняемых файлов системных программ;
- `/boot` («**Boot**» – загрузка системы) – файлы, необходимые для самого первого этапа загрузки: загрузки ядра и, обычно, само ядро;
- `/etc` - каталог для конфигурационных файлов;
- `/dev` - каталог для файлов внешних устройств;
- `/lib` ("libraries"– "библиотеки") - каталог для хранения разделяемых библиотек;
- `/var` и `/tmp` - каталоги для всякого рода регулярно изменяемых или временных данных;
- `/usr` - каталог для пользовательских программ со всеми их файлами. Например, с документацией;
- `/home` - место пользовательских каталогов с данными.
- `/mnt` (`/media`) – каталог для монтирования (от англ. "mount") для временного подключения файловых систем;
- `/root` домашний каталог администратора системы – пользователя **root**.

Глава 4

Терминал Linux и основные команды

Некоторые действия в Linux могут осуществляться только в режиме терминала, а иногда работать в нем проще, чем в графическом режиме.

Терминал – это программа, которая используется для взаимодействия пользователя с операционной системой.

Терминал необходим для решения административных задач, таких как установка пакетов, управление пользователями, действия с файлами т. д.

Запуск осуществляется командой **Системные - Терминал**¹. После чего в окне терминала можно вводить команды вида

команда параметры

Однако, ряд действий можно осуществить только с правами администратора, для чего вводится команда:

sudo команда параметры

После этого система спросит пароль администратора.

Текстовый редактор nano

Для редактирования небольших текстов можно исполь-

¹Во многих дистрибутивах Linux терминал можно вызвать комбинацией клавиш **Ctrl+Alt+T**

зовать текстовый редактор **nano**, который входит в набор программ GNU и является стандартным консольным редактором для многих дистрибутивов Linux. Справка по возможностям nano доступна после его запуска с помощью комбинации клавиш Ctrl+G (F1). Рассмотрим некоторые комбинации клавиш в nano.

- ^G – справка;
- ^O – сохранить данные в файле;
- ^X – закрыть nano;
- ^W – поиск текста;
- ^\ – замена текста;
- ^B – пометка начала и конца выделяемого фрагмента текста, для выделения текста можно использовать клавиши управления курсором;
- ^K – вырезать текст в буфер;
- M^ – копировать текст в буфер;
- ^U – вставить текст из буфера;
- M U - отмена последнего действия;
- ^Y – переход к первой строке;
- ^V – переход к последней строке;
- ^R – прочитать файл.

Здесь ^ – клавиша Ctrl, M – клавиша Alt. Более подробное описание возможностей редактора nano можно найти тут <https://losst.ru/tekstovyyj-redaktor-nano-v-linux-dlya-novichkov>

или тут <https://darksf.ru/2018/07/25/rukovodstvo-po-ispolzovaniyu-teksto>.

Рассмотрим основные команды Linux, которые могут понадобиться рядовому пользователю.

Команды получения справочной информации

ОС Linux позволяет получить справочную информацию по самой системе. Современные дистрибутивы содержат тысячи страниц документации, представленной в электронном виде. Выводить эту информацию можно с помощью страниц интерактивного руководства `man`.

Команда `man`

Команда **`man`** имеет вид

`man name`

Здесь `name` - название команды или другого объекта, о котором нужна информация. Например, команда

`man pwd`

выведет на экран информацию о команде **`pwd`**. Для управления процессом вывода используются следующие клавиши:

- `Q` – выход из программы;
- `Enter` – просмотр строка за строкой;
- `Space` – вывод следующего экрана информации;
- `B` – вернуться к предыдущему экрану;
- `/` за которой следует строка символов и `Enter` – поиск введенной строки символов;
- `N` – повторение предыдущего поиска.

Если необходимо отпечатать копию страницы, то можно воспользоваться командой

```
man name | lpr
```

для postscript-принтера

```
man -t name | lpr
```

В русскоязычных дистрибутивах часть man'ов переведены на русский язык, однако в основном справочная информация англоязычна.

Команда info

Команда info является альтернативой команде man. Она имеет вид

```
info name
```

здесь name – имя команды.

Например,

```
info pwd
```

Информация, которую вы увидите, в большинстве случаев несколько отличается от той, которую дает команда man. Но самое существенное отличие заключается в том, что выдаваемая info информация представлена в гипертекстовом формате. В силу этого появляется возможность просматривать различные разделы помощи. Работая в тестовом режиме, вы можете запустить info в одной из альтернативных консолей. Переход от графического режима к текстовым альтернативным консолям осуществляется комбинациями клавиш: **Ctrl+Alt+F2**, **Ctrl+Alt+F3**, возврат в графический – **Ctrl+Alt+F7**.

Команда help

Если ввести в командной строке help без параметров, то будет выведен список всех встроенных команд. Если ввести команду

```
help name
```

где name – имя одной из этих команд, то выведется очень краткая справка о применении этой команды.

Команды для работы с файлами

Для работы с файлами и каталогами чаще всего применяются следующие команды:

- `touch file` создание пустого текстового файла с именем `file`;
- `cat file` просмотр файла с именем `file`;
- `tac file` просмотр текстового файла с именем `file` в обратном порядке, от последней строки к первой;
- `cp file1 file2` копирование `file1` в `file2`; `mv file1 file2` перемещение `file1` в `file2`;
- `rm file` удаление файла с именем `file`;
- `rm -r dir` рекурсивное удаление каталога `dir` (вместе с содержимым);
- `rmdir dir` удаление пустого каталога `dir`;
- `locate file` быстрый поиск файла;
- `which file` вывод каталога, в котором находится программа, если она установлена;
- `less file` используется для просмотра файла с возможностью скроллинга;
- `mkdir dir` в текущем каталоге создает каталог с именем `dir`;
- `cd dir` переход в каталог с именем `dir`;
- `ls dir` вывод содержимого каталога `dir`.

Вместо имен каталогов можно использовать:

- `.` – текущий каталог;
- `..` – родительский каталог (на уровень выше);
- `~` – домашний каталог пользователя;

Рассмотрим примеры нескольких связанных между собой команд работы с файлами и каталогами.

```
touch ./my.txt
```

В текущем каталоге создаем пустой текстовый файл `my.txt`.

```
mkdir dir1
```

В текущем каталоге создаем каталог `dir1`.

```
nano ./my.txt
```

Вызываем текстовый редактор `nano` и редактируем в нем файл `./my.txt`.

```
cp ./my.txt ./dir1/my2.txt
```

Копируем файл `my.txt` из текущего каталога в созданный в нем каталог `dir1` под именем `my2.txt`.

```
cp ./my.txt ./dir1/my3.txt
```

Копируем файл `my.txt` из текущего каталога в подкаталог `dir1` под именем `my3.txt`.

```
sudo mv ./dir1/my3.txt /tmp/3.txt
```

Переносим файл `./dir1/my3.txt` в каталог `/tmp` под именем `3.txt`. Для выполнения команды нужны права суперпользователя.

```
rm ./my.txt
```

Удаляем файл `my.txt` из текущего каталога.

```
cd ./dir1
```

Переходим в каталог `./dir1`, который теперь будет текущим.

```
cp ./my2.txt ./my2.txt
```

Копируем файл `./my2.txt` из текущего каталога (теперь это каталог `dir1`) под именем `my2.txt` в этом же каталоге.

```
ls .
```

Вывод на экран содержимого текущего каталога.

```
cd ..
```

Переход на уровень выше.

```
rm -r dir1
```

Удаляем каталог `dir1` вместе с его содержимым.

Поиск файлов. Команда `find`

Команда `find` имеет вид:

```
find путь -опции
```

Здесь путь – полный путь к каталогу, в котором следует произвести поиск. Опции

- `-name` (поиск файлов по имени);
- `-user` (поиск файлов, принадлежащих указанному пользователю);
- `-group` (поиск файлов, принадлежащих указанной группе);
- `-perm` (поиск файлов с указанным режимом доступа);
- `-type` (поиск файлов определенного типа);
- `-size ±n` (поиск файлов с размером `n` единиц (`b` - блоки 512 байт, `c` - байт, `k` - килобайт, `M` - мегабайт, `G` - гигабайт));
- `-mtime ±n` (поиск файлов, содержание которых изменялось менее чем «-» или более чем «+» дней назад)

Примеры

```
find / -name config -print
```

Искать файлы с именем config (-name config), начиная с каталога / (первый параметр команды find) и выводить имена файлов на экран (аргумент -print). Поиск рекурсивный.

```
find . -type f -name "~*" -print
```

Найти в текущем каталоге обычные файлы (не каталоги), имя которых начинается с символа ~.

```
find . -newer file.bak -type f -print
```

Найти в текущем каталоге файлы, измененные позже, чем файл file.bak.

```
find . -type f -empty
```

Найти в текущем каталоге пустые файлы.

Перенаправление информации в файл

Основные операторы перенаправления представлены в таблице. Пример:

Оператор перенаправления	Назначение
>	Данные на выходе команды выводятся не в консоль, а в файл или на другое устройство (принтер)
>>	Выходные данные дописываются в конец файла, при этом сохраняется уже существующая в файле информация
<	Считывание данных не с клавиатуры, а из файла
	Выходные данные одной команды считываются и записываются, как входные данные другой команды

```
cat>file1
Hello world!
```

(для завершения ввода текста нажмите сочетание клавиш Ctrl+D).

В file1, который автоматически был создан в домашнем каталоге пользователя, записана фраза «Hello world!»

```
cat file1
```

В терминале отобразится строка из файла file1 «Hello world!»

```
cat>file2
some other text
```

В file2, который так же автоматически был создан в домашнем каталоге пользователя, записана фраза «some other text»

```
cat<file1 | cat>>file2
```

Содержимое file1 было считано и перенаправлено на ввод в file2, при этом данные дописываются в конец файла, сохраняя уже существующую строку «some other text»

```
cat file2
```

На экран выводится содержимое file2. А именно, строка «some other text», и «Hello world!», которая была скопирована из file1.

При перенаправлении информации несколько файлов можно объединить в один.

Пример:

Запишем в file1 текст "Строка1". В file2 - "Строка2". В file3 - "Строка3". Затем содержимое file1 и file2 допишем в конец файла file3

```
cat > file1
Строка1
cat > file2
Строка2
```



```
cat > file3
```

Строка3

```
cat file1 file2 » file3
```

```
cat file3
```

В результате получим три надписи в следующем порядке:

Строка3

Строка1

Строка2

Глава 5

Создание нового пользователя. Изменение прав доступа

Команды управления пользователями

Для создания нового пользователя можно воспользоваться одной из следующих команд

```
sudo adduser nameuser  
sudo useradd nameuser
```

- **useradd** – команда, которая создает пользователя, не имеющего домашнего каталога (если не указан ключ `m`) и каких-либо дополнительных настроек.
- **adduser** – более безопасная команда, которая создает пользователя вместе с его домашним каталогом и другими пользовательскими настройками.
- **sudo passwd nameuser** – задать пароль пользователю `nameuser`.

Параметры:

- **-d, --home homedir** новый домашний каталог учётной записи

- `-e, --expiredate date` установить дату окончания действия учётной записи в `date`
- `-f, --inactive inactivity` установить период неактивности пароля после устаревания учётной записи равным `inactivity`
- `-h, --help` показать данное сообщение и закончить работу
- `-l, --login newname` новое значение имени учётной записи
- `-L, --lock` заблокировать учётную запись
- `-m, --move-home` переместить содержимое домашнего каталога в новое место (использовать только вместе с `-d`)
- `-o, --non-unique` разрешить создание учётной записи с уже имеющимся `UID`
- `-p, --password pass` задать новый шифрованный пароль для учётной записи

Сменить пользователя можно несколькими способами:

1. После загрузки системы на экране появляется список пользователей. Требуется выбрать того пользователя, под которым нужно войти в систему, и ввести пароль.
2. После нажатия стандартного сочетания клавиш **Ctrl+Alt+L** блокируется экран устройства. При попытке входа в систему на экране появляется кнопка «переключить пользователя». Требуется выбрать того пользователя, под которым нужно войти в систему, и ввести пароль.

3. В терминале смена пользователя осуществляется командой `su - nameuser` (где `nameuser` – это имя пользователя)

Права доступа в ОС Linux

В Windows можно определить права доступа для каталога, и они автоматически распространяются на все файлы и подкаталоги. В ОС Linux права не наследуются, они у каждого файла свои. Чтобы увидеть права доступа к файлам текущего каталога, достаточно набрать в терминале `ls -l`. В первом столбце выведены права доступа. Первый символ - это тип файла, остальные девять - права.

```
$ ls -l
-rw-r--r-- 1 evg evg 212 2008-09-12 19:38 1.tex~
-rw-r--r-- 1 evg evg 195 2008-09-12 19:42 2~
-rw-r--r-- 1 evg evg 223 2008-09-12 20:00 2.tex~
drwxr-xr-x 2 evg evg 4096 2008-09-14 13:03 Example
-rw-r--r-- 1 evg evg 1647594 2008-09-12 22:59 Ex.zip
-rw-r--r-- 1 evg evg 1585850 2008-09-13 19:30 ll.pdf
-rw-r--r-- 1 evg evg 6229739 2008-09-13 19:33 lv.pdf
-rw-r--r-- 1 evg evg 878 2008-09-12 19:09 nf.lyx~
drwxr-xr-x 2 evg evg 4096 2008-09-14 12:40 primer
drwxr-xr-x 2 evg evg 4096 2008-09-14 13:03 proba
drwxr-xr-x 2 evg evg 4096 2008-09-10 13:42 Regul
-rw----- 1 evg evg 342861 2008-09-12 23:45 Reg.zip
-rw-r--r-- 1 evg evg 23112 2008-09-13 17:15 edit.odt
```

Типы файлов:

- `b` – специальный блочный файл;
- `d` – каталог;
- `c` – специальный символьный файл;

- `f` – обычный файл, который может обозначаться и через тире;
- `l` – символическая ссылка;
- `p` – именованный канал;
- `s` – сокет.

Права доступа делятся на три группы:

<code>r w x</code>		<code>r w x</code>		<code>r w x</code>
<code>user</code>		<code>group</code>		<code>other</code>

- `user` – права пользователя;
- `group` – права группы, к которой принадлежит пользователь;
- `other` – права всех остальных пользователей системы.

Права доступа к файлам и каталоге имеют различное значение. Для файлов:

- `r` – право на чтение данных из файла;
- `w` – право на изменение содержимого файла (запись);
- `x` – право на исполнение файла.

Право `w` не дает права на удаление файла, только на изменение содержимого. Если у файла установлено право `x`, его можно запустить на выполнение.

Права доступа к каталога интерпретируются по-другому:

- `r` – право на чтение (просмотр) содержимого каталога (команда терминала `ls`);
- `w` – право на изменение содержимого каталога – создание и удаление объектов в этом каталог;

- x – право на «вход» в каталог (команда терминала cd).

Для упрощения записи у каждого типа доступа есть числовое представление, для используется двоичная система счисления. Единица – есть право, 0 – нет права. Таким образом запись прав rwx r-x r-x в бинарном виде будет выглядеть следующим образом: 111 101 101. Но двоичное представление не очень удобно, поэтому используют восьмеричное. 111 в двоичной системе – это 7 в восьмеричной, а 101 – это 5, таким образом 111 101 101 - это 755. Итак, получаем:

000		001		010		011		100		101		110		111
---		--x		-w-		-wx		r--		r-x		rw-		rwX
0		1		2		3		4		5		6		7

Основные команды для работы с правами доступа

Команда **chgrp** позволяет изменить группу владельца файла. Если эту команду использует простой пользователь, то он, во-первых, должен быть владельцем файла, а во-вторых, он должен быть членом группы, которой он хочет дать права. Пользователь не должен быть членом группы, которая владеет файлом. Синтаксис следующий:

```
chgrp newgroup files
```

можно использовать как имя, так и ID группы (GID).

```
$ ls -l
-rw-r-r- 1 adm root 0 2007-09-03 15:33 file.txt
$ chgrp users file.txt
$ ls -l
-rw-r-r- 1 adm users 0 2007-09-03 15:33 file.txt
```

Команду можно использовать с ключом **-R**, что приводит к рекурсивному обходу всех файлов в каталоге и его подкаталогах.

Команда **chown** используется для изменения как владельца, так и группы. В большинстве систем, только суперпользователь имеет право пользоваться этой командой. У команды следующий синтаксис:

```
chown newuser:newgroup files
```

Как и в команде **chgrp**, вы можете использовать как имена пользователей и групп, так и их идентификаторы **GID** и **UID**. Команда не проверяет пользователей и группы на существование, так что можно задать и несуществующие.

```
$ ls -l
-rw-r-r- 1 adam users 0 2007-09-03 15:33 file.txt
root@laptop:~# chown root:root file.txt
adam@laptop:~$ ls -l
-rw-r-r- 1 root root 0 2007-09-03 15:33 file.txt
```

Смена группы и пользователя

```
chown student file.txt
```

Смена владельца на пользователя **student**, группа остается без изменений.

```
chown :university file.txt
```

Смена группы на **university**, владелец без изменений. Рассмотрим команду **chmod** для смены прав доступа:

```
chmod access rights files
```

Перед выполнением команды необходимо выбрать класс пользователей, для кого мы хотим изменить права: **u**, **g** или **o** (**User**, **Group**, **Others**). Есть так же дополнительная группа **a** (**all**, все). Затем выбираем оператор: **+** (дать права), **-** (убрать права) и **=** (присвоить права). И в конце выбрать сами права (**rxw**, **r-x** и т.д.).

```
chmod u+x skrypt.sh
```

Пользователю (**u**) добавляется (+) право на выполнение (**x**) файла **skrypt.sh**.

```
chmod go-r raport.odt
```

У группы и остальных (**go**) отбирается (-) право на чтение (**r**).

```
chmod a=w finansse.ods
```

Всем (**a**) присваивается (=) право на запись (**w**), а остальные права стираются (=).

```
chmod u+rw, g+rw, o+x trurl.py
```

Пользователю прибавляются права на все, группе тоже, а остальным прибавляется право на выполнение.

Существует и второй синтаксис команды **chmod**

```
chmod -R ugo file
```

Здесь

- **u** – цифра в восьмеричной системе счисления, определяющая права доступа для владельца файла;
- **g** – цифра в восьмеричной системе счисления, определяющая права доступа для группы, к которой принадлежит владелец файла;
- **o** – цифра в восьмеричной системе счисления, определяющая права доступа для всех остальных пользователей;
- **file** – имя файла или каталога;
- **-R** – параметр, позволяющий выполнить команду рекурсивно, если речь идёт об изменении прав каталога (под рекурсией тут понимается распространение действия команды на все файлы и каталоги, которые находятся внутри указанного в команде); без этого ключа права меняются только для указанного каталога.

Примеры:

```
chmod 710 ./progr
```

После выполнения этой команды

- владелец файла **progr** может просматривать, редактировать и исполнять файл **progr**, который находится в текущем каталоге (**7 – 111**);
- члены группы могут только запускать файл **progr** на выполнение (**1 – 001**);
- у всех остальных пользователей нет никаких прав для работы с этим файлом **0 – 000**

```
chmod 710 ./progr
```

Каталогу **abc** из домашнего каталога пользователя и всем файлам и каталогам, которые находятся внутри **abc** установить следующие права

- у владельца каталога есть права чтения, модификации и запуска (**7 – 111**);
- у членов группы, которой принадлежит владельца каталога, есть право чтения (**4 – 100**);
- у всех остальных, есть право чтения (**4 – 100**).

Пришло время разобраться, какая информация хранится в системе о файле. Для того, чтобы увидеть всю информацию о файлах и каталогах необходимо использовать команду **ls** с ключами **-li**.

Пример:

```
ls -li .
итого 32
61997811 drwxrwxrwx 2 aer aer 4096 апр 26 2020 bin
61997770 drwxrwxrwx 3 aer aer 4096 апр 26 2020 etc
61997773 drwx-wxrw 3 aer aer 4096 апр 26 2020 include
61870058 drwxrwx-wx 3 aer aer 4096 апр 26 2020 lib
61869170 drwxrwxrwx 2 aer aer 4096 апр 26 2020 libexec
61870062 -rwxrwx--- 1 aer aer 5218 апр 14 2020 LICENSE.md
61869172 drw-rwxrwx 8 aer aer 4096 апр 26 2020 share
```

Первый столбец – значение дескриптора

Второй столбец (десять символов) определяют тип файла (1 символ) и права доступа (9 символов).

Типы:

d – каталог,

- – обычный файл.

Права доступа: первые три символа – права владельца, затем права членов группы, последние три – права всех остальных.

Число в **третьем столбце** определяет:

- для файлов – количество жёстких ссылок, подробнее о них в седьмой главе;
- для каталогов – количество вложенных каталогов +2 (добавляются два каталога, текущий и родительский каталоги).

Четвертый столбец – владелец файла (каталога).

Пятый столбец – группа, в которой состоит владелец файла.

Шестой столбец – размер файла, для каталога размером является число 4096.

Седьмой столбец – дата создания (модификации) файла.

Седьмой столбец – имя файла.

Параметр **итого (total)** определяет размер каталога в блоках.

Глава 6

Пакеты в ОС Linux

Ранее уже было сказано, что в ОС Linux приложения распространяются в виде пакетов. В дистрибутивах на базе Debian (Ubuntu, Linux Mint, Astra Linux) – это deb пакеты, в Red Hat, OpenSuse, Alt Linux в репозитории - это rpm пакеты. И те и другие хранятся в репозиториях.

Репозиторий – место хранения и поддержки каких-либо данных. Эти данные чаще всего имеют вид файлов, доступных для дальнейшего распространения по сети.

Репозиторий дистрибутива – хранилище программ дистрибутива. Все описанные ниже утилиты требуют прав суперпользователя.

Утилита установки deb пакетов `dpkg`

Основные ключи утилиты `dpkg`

- `dpkg -i file.deb` – установка пакета (обновление, если пакет установлен);
- `dpkg -r file.deb` – удаление пакета;
- `dpkg -l file` – поиск установленных пакетов (`file` - маска поиска пакета, можно использовать *, ?);
- `dpkg -L file.deb` – вывод списка файлов из установленного пакета;

- `dpkg -p file.deb` – вывести информацию об установленном пакете;
- `dpkg -unpack file.deb` – распаковать, но не устанавливать пакет.

Универсальной командой для работы с пакетами `rpm` является утилита `rpm` <https://losst.ru/ustanovka-rpm-paketov-v-linux>.

Утилита `rpm`

Синтаксис утилиты `rpm`

`rpm -режим-опции пакет`

Режимы:

- `-q`, `--query` - запрос, получение информации (часто используется для проверки установлен ли пакет);
- `-i`, `--install` - установка;
- `-V`, `--verify` - проверка пакетов;
- `-U`, `--upgrade` - обновление;
- `-e`, `--erase` - удаление.

Опции программы:

- `-v` – показать подробную информацию;
- `-h` – выводить статус-бар;
- `--percent` – выводить информацию в процентах о процессе распаковки;
- `--force` – выполнять действие принудительно;

- `-i` – получить информацию о пакете;
- `-l` – список файлов пакета;
- `-R` – вывести список пакетов, от которых зависит текущий.

Простейшая команда установки:

```
rpm {-}i имяпакета.rpm
```

Для вывода более подробную информацию о процессе установки можно воспользоваться опцией `-v`:

```
rpm {-}iv имяпакета.rpm
```

Для включения статус бара в процессе установки служит опция `-h`:

```
rpm {-}ihv имяпакета.rpm
```

Для установки пакетов из репозитория используются утилиты `apt` и `apt-get` ¹.

Утилита `apt`

Основные команды утилиты `apt`(`apt-get`)

- `apt update` (`apt-get update`) – обновление баз данных доступных программных пакетов;
- `apt upgrade` (`apt-get upgrade`) – проверяет наличие обновлений для всех установленных пакетов и предлагает загрузить и установить их;
- `apt install package` (`apt-get install package`) – загрузка и установка программного пакета по заданному названию (если он найден в базах данных);

¹Описанные ниже утилиты используются в дистрибутивах на базе Debian, а также в российских дистрибутивах «Альт» <https://www.basealt.ru>

- `apt -f install` (`apt-get -f install`) – поиск неработающих программных пакетов и попытка отладить те из них, которые отмечены сообщением «unmet dependency» (неудовлетворенные зависимости);
- `apt autoclean` (`apt-get autoclean`) – удаляет не полностью загруженные или ещё не установленные пакеты;
- `apt autoremove` (`apt-get autoremove`) – удаляет не используемые в системе библиотеки;
- `apt remove package` (`apt-get remove package`) – удаление пакета, оставляя конфигурационные файлы;
- `apt purge package` (`apt-get purge package`) – удаление пакета вместе с конфигурационными файлами;
- `apt search package` – поиск пакета (по названию и краткому описанию);
- `apt show package` (`apt-get show package`) – подробная информация о пакете;
- `apt deb package` (`apt-get deb package`) – установка скачанного локального deb-пакета.

Отличие `apt` и `apt-get` можно прочитать тут <https://www.alv.me/Upravlenie-deb-paketami-Utilita-apt-bez-izl/> или тут <https://losst.ru/apt-vs-apt-get-v-chem-raznitsa>.

Глава 7

Жёсткие и символьные ссылки

Понятие ссылки

Каждый файл представляет собой область данных на диске.

Жесткая ссылка на файл – это ссылка, с помощью которой связывается имя файла и область данных (на диске), где располагается его содержимое. Количество таких ссылок не ограничено.

При просмотре содержимого каталога с ключом `-i` (`ls -i`) можно увидеть не только имя файла, но и его дескриптор (уникальное числовое значение, которое присваивается файлу при его создании).

Жёсткая ссылка

Пользователь Linux может создать жёсткую ссылку на файл при помощи утилиты `ln` (сокращение от англ. «link» – «соединять, связывать»). Первый параметр – это имя файла, на который нужно создать ссылку, второй – имя новой ссылки (жёсткой).

По умолчанию ссылка будет создана в текущем каталоге.

```
nano first.txt  
ls -li
```

итого 4

```
9190023 -rw-rw-r-- 1 aer aer 9 окт 25 10:48 first.txt
```

Здесь 9190023 – дескриптор – уникальный числовой идентификатор, связанный с физическим местом на диске. Создадим жёсткую ссылку на файл.

```
ln first.txt second.txt
```

Создадим каталог 3, и в нём создадим ещё одну одну ссылку на файл `first.txt`.

```
mkdir 3
```

```
ln first.txt 3/third.txt
```

```
ls -li -R
```

```
9190023 -rw-rw-r-- 1 aer aer 9окт 25 10:48 first.txt
```

```
9190023 -rw-rw-r-- 1 aer aer 9окт 25 10:48 second.txt
```

```
3:
```

итого 4

```
9190023 -rw-rw-r-- 1 aer aer 9 окт 25 10:48 third.txt
```

```
ln first.txt second.txt
```

Таким образом мы имеем три файла, указывающих на одну и ту же область данных. У всех трёх файлов один и тот же дескриптор (9190023). Если мы удалим первый файл (`first.txt`), то файлы `second.txt`, `third.txt` будут по прежнему указывать на ту же область данных.

```
rm first.txt
```

```
ls -li *.* -R
```

```
9190023 -rw-rw-r-- 1 aer aer 9окт 25 10:48 second.txt
```

```
3:
```

итого 4

```
9190023 -rw-rw-r-- 1 aer aer 9 окт 25 10:48 third.txt
```


Как мы видим, одна и та же область памяти на диске может иметь несколько имён (жёстких ссылок).

У жёстких ссылок есть два существенных ограничения:

- жёсткая ссылка не может указывать на каталог; это может привести к бесконечной циклической ссылке на один и тот же каталог;
- жёсткая ссылка может указывать на файл только в рамках одной файловой системы.

Причина последнего ограничения в том, что номер дескриптора уникален только в рамках одной файловой системы. В разных файловых системах могут оказаться два разных файла с одинаковым номером дескриптора, в результате будет невозможно установить, на какой из них указывает жёсткая ссылка.

Символьная ссылка

Символьная ссылка – файл, в котором хранится полное имя другого файла или каталога. Аналогом символьной ссылки является понятие ярлыка. Дескриптор символьной ссылки отличается от дескриптора файла (каталога), на который она ссылается.

При создании символьной ссылки используется команда `ln` с ключом `-s`. Рассмотрим предыдущий пример, но изменим тип создаваемой ссылки. Будем создавать символьную ссылку. Предполагается, что файл `first.txt` создается в домашнем каталоге.

```
nano first.txt
ln -s first.txt second.txt
mkdir 3
ln -s ~/first.txt 3/third.txt
ls -li -R
rm first.txt
ln -li -R
```

Перед удалением файла `first.txt`.

```
9190023 -rw-rw-r-- 2 aer aer 9 окт 25 10:48 first.txt first.txt
9190026 lrwxrwxrwx 1 aer aer 9 окт 25 11:31 second.txt -> first.txt
3:
итого 0
9190028 lrwxrwxrwx 1 aer aer 9 окт 25 11:34 third.txt -> first.txt
```

После удаления файла `first.txt`.

```
итого 4
9190026 lrwxrwxrwx 1 aer aer 9 окт 25 11:31 second.txt -> first.txt
3:
итого 0
9190028 lrwxrwxrwx 1 aer aer 9 окт 25 11:34 third.txt -> first.txt
```

После удаления файла `first.txt` символические ссылки `second.txt` и `third.txt` указывают на несуществующий файл.

Глава 8

Особенности настройки АльтЛинукс после установки

Под первоначальной настройкой будем понимать нестандартную установку необходимого программного обеспечения, а не настройку визуальных эффектов. Мы рассмотрим все возможные способы установки программ в АльтЛинукс. Но всё ниже описанное должно работать в системе, построенной на базе любого дистрибутива Альт 10 платформы. В коммерческих дистрибутивах (и в Симпли Линукс) часть из описанного ниже уже выполнена автоматически.

Обновление системы

Обновление системы выполняется от имени администратора. Давайте рассмотрим основные возможности управления пользователями в дистрибутивах семейства Альт десятой платформы.

Управление пользователями в Альт. Администратор системы

Смена пользователя в простейшем случае осуществляется командой

```
su [—] [user]
```

Здесь `user` – имя пользователя, от имени которого осуществляется вход в систему. Если имя пользователя отсутствует, то речь идёт об администраторе (пользователь `root`).

После ввода команды необходимо ввести пароль пользователя `user`. Особую роль играет символ `-` (минус).

Если после команды `su` ввести символ минус (`su - user`), то вход в систему будет произведён от имени пользователя `user`, и домашним каталогом будет `/home/user`. Все настройки системы в этом случае хранятся в этом же каталоге `/home/user`.

Если же знак минус отсутствует в команде `su user`, то вход в систему будет произведён от имени пользователя `user`, но с домашним каталогом и настройками текущего пользователя.

В завершении отметим особенности команд `su` и `su -`.

После выполнения команды `su` и ввода пароля администратора, Вы получаете права администратора и можете выполнять любые действия. Однако следует помнить, что домашним каталогом остаётся каталог текущего пользователя системы и сохраняются *все настройки системы текущего пользователя*.

После выполнения команды `su -` и ввода пароля администратора, ситуация становится другой. Вы входите в систему, как администратор. Домашним каталогом становится каталог `/root`. Настройки системы – *настройки пользователя root*. Именно у пользователя `root` в настройках пути ко всем системным программам. И поэтому рекомендуется перед выполнением команд обслуживания системы получить не просто «права администратора» (`su`), а именно «стать администратором» (`su -`).

Обновление системы

Обновление системы осуществляется с помощью двух команд, которые должны быть выполнены от имени администратора (пользователь `root` (`su -`)).

- `apt-get update` – обновление баз данных доступных программных пакетов;
- `apt-get dist-upgrade` – команда проверяет наличие обновлений для всех установленных пакетов и предлагает загрузить и установить их.

Источники программного обеспечения

Программное обеспечение может быть установлено из репозитория АльтЛинукс, непосредственно из `rpm` пакета, а также с помощью особого механизма установки популярных приложений, в том числе и проприетарных.

Установка программного обеспечения из репозитория (Сизиф)

Свободные приложения могут быть установлены из репозитория АльтЛинукс – Сизиф. В терминале можно воспользоваться командой `apt-get` (от администратора)

- `apt-get update` – обновление баз данных доступных программных пакетов;
- `apt-get install package` – загрузка и установка локального пакета или пакета из репозитория;
- `apt-get -f install` – поиск неработающих программных пакетов и попытка отладить те из них, которые отмечены сообщением «unmet dependency» (неудовлетворенные зависимости);
- `apt-get autoclean` – удаляет не полностью загруженные или ещё не установленные пакеты;
- `apt-get autoremove` – удаляет не используемые в системе библиотеки;

- `apt-get remove package` – удаление пакета, оставляя конфигурационные файлы;
- `apt-get purge package` – удаление пакета вместе с конфигурационными файлами;
- `apt-get find package` – поиск пакета (по названию и краткому описанию);
- `apt-get show package` – подробная информация о пакете.

Установку, удаление и обновление приложений можно осуществлять с помощью приложения с графическим интерфейсом Synaptic (см. рис. 8.1).

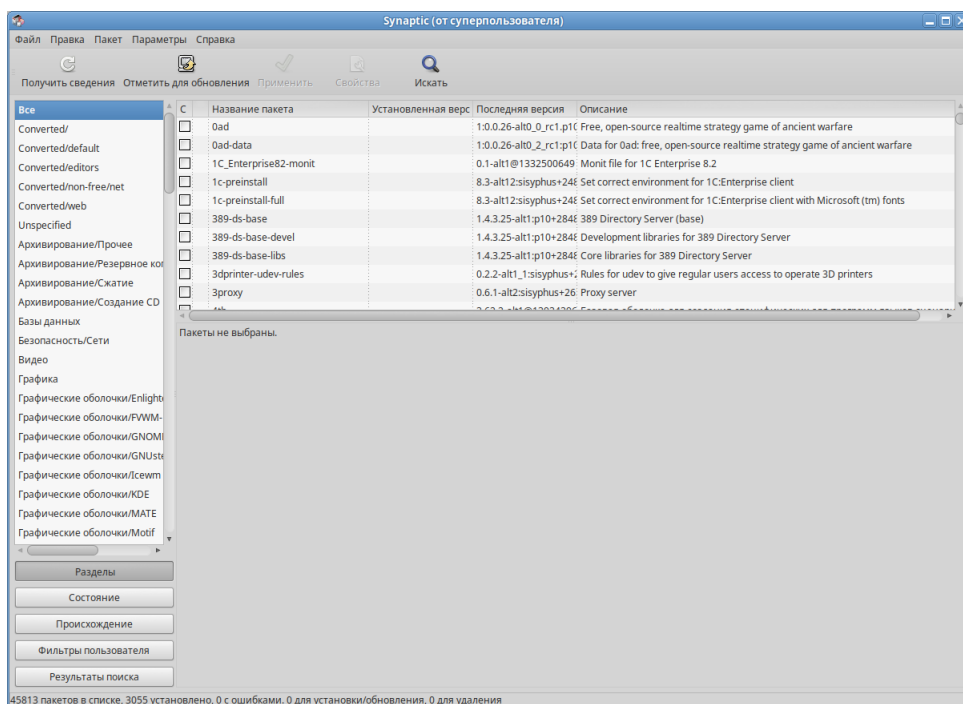


Рис. 8.1: Менеджер пакетов Synaptic

Отдельные локальные `rpm` пакеты могут быть также установлены с помощью утилиты `rpm` (см. стр. 36).

Установка бесплатного и проприетарного ПО

Современный пользователь компьютера привык к ряду популярных бесплатных кроссплатформенных приложений (Geogebra, Yandex Disk, браузеры Google Chrome, Yandex и ряд других). В дистрибутивах АльтЛинукс десятой платформы автоматизирован процесс установки этих программ. Существует графическое приложение `appinstall` (см. рис. 8.2¹).

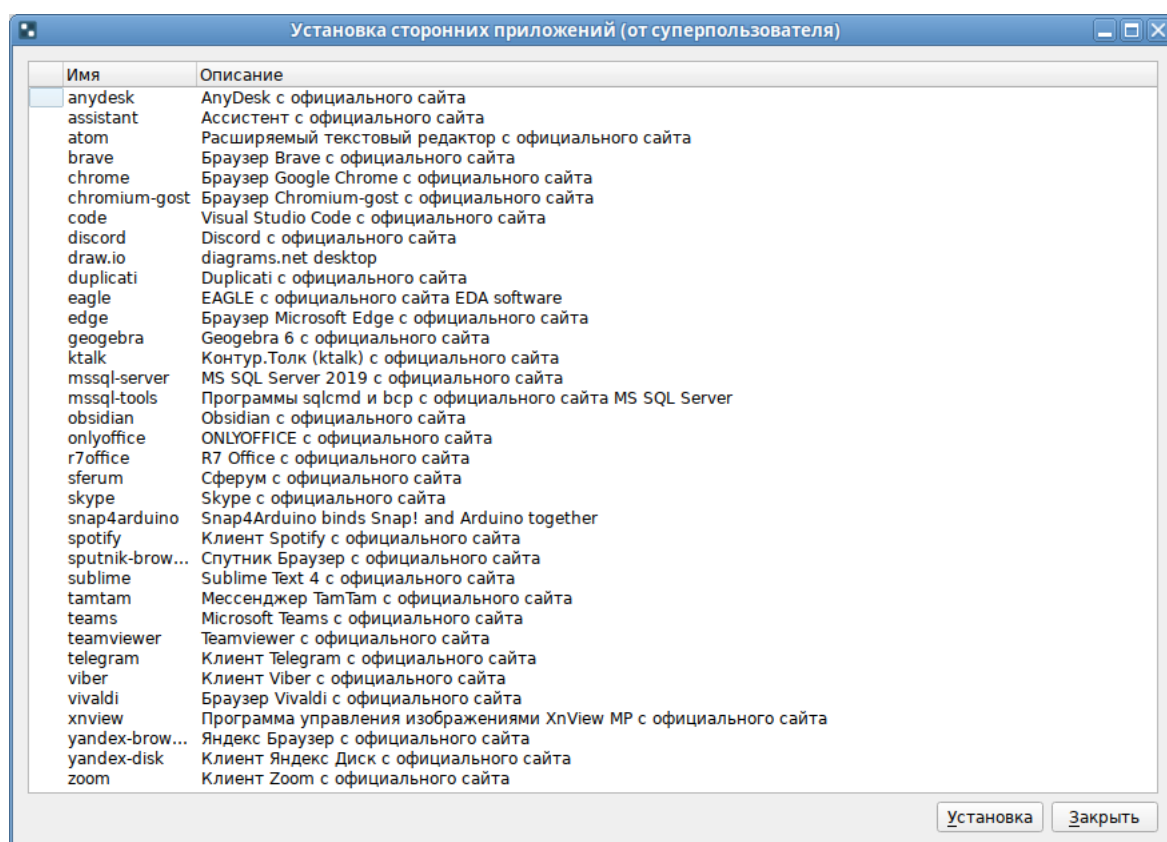


Рис. 8.2: Утилита `appinstall`

С помощью этой программы можно установить любое приложение из списка. Установка включает в себя скачива-

¹Если приложение не установлено, то его можно установить стандартным образом `apt-get install appinstall`.

ние последней версии приложения с официального сайта и непосредственно установка.

Приложение `appinstall` является графическим интерфейсом над консольной утилитой `ерм`.

Утилита `ерм`¹

Для установки, удаления и обновления программ с сайта разработчика необходимо выполнять команду `ерм play` от администратора.

`ерм play`

Available applications:

- `anydesk` - AnyDesk from the official site
- `assistant` - Assistant (Ассистент) from the official site
- `atom` - The hackable text editor from the official site
- `brave` - Brave browser from the official site
- `chrome` - The popular and trusted web browser by Google (Stable Channel) from the official site
- `chromium-gost` - Chromium with GOST support from the official site
- `code` - Visual Studio Code from the official site
- `discord` - Discord from the official site
- `draw.io` - diagrams.net desktop
- `duplicati` - Duplicati from the official site
- `eagle` - EAGLE (EDA software) from the official site
- `edge` - Microsoft Edge browser(dev) from the official site
- `geogebra` - Geogebra 6 from the official site
- `ktalk` - Контур.Толк (ktalk) from the official site
- `mssql-server` - MS SQL Server 2019 from the official site
- `mssql-tools` - MS SQL Server sqlcmd and bcp from the official site
- `obsidian` - Obsidian from the official site
- `onlyoffice` - ONLYOFFICE for Linux from the official site
- `r7office` - R7 Office for Linux from the official site
- `sferum` - Sferum for Linux from the official site
- `skype` - Skype for Linux from the official site
- `snap4arduino` - Snap4Arduino binds Snap! and Arduino together
- `spotify` - Spotify client for Linux from the official site
- `sputnik-browser` - Sputnik browser from the official site
- `sublime` - Sublime Text 4 from the official site
- `tamtam` - TamTam messenger from the official site
- `teams` - Microsoft Teams for Linux from the official site
- `teamviewer` - Teamviewer from the official site
- `telegram` - Telegram client from the official site
- `viber` - Viber for Linux from the official site
- `vivaldi` - Vivaldi browser from the official site

¹Если приложение не установлено, то его можно установить стандартным образом `apt-get install ерм`

xnview -XnView MP:Image management from the official site
yandex-browser - Yandex browser from the official site
yandex-disk - Yandex Disk from the official site
zoom - Zoom client from the official site

Синтаксис утилиты `epm`

`epm play [options] [<app>]`

Options:

- `<app>` - установка приложения `<app>`;
- `--remove <app>` - удаление приложения `<app>`;
- `-- update [<app>|all]` — обновление приложения `<app>` (или всех установленных приложений) если доступны новые версии;
- `--list` — вывод списка всех установленных приложений с помощью `epm`;
- `--list-all` — вывод списка всех доступных приложений;
- `--list-scripts` — вывод списка всех доступных установочных скриптов;
- `--short (with --list)` — вывод списка всех доступных только имён приложений;
- `--installed <app>` - проверка установлено ли приложение с именем `app`.

Набор необходимого ПО для науки и образования, особенности установки и настройки

В репозиторий Альт Линукс десятой платформы включены скрипты для установки наборов образовательного ПО:

- `task-edu-gradeschool` – образовательное программное обеспечение (начальная школа);
- `task-edu-highschool` – образовательное программное обеспечение (средняя школа);
- `task-edu-kde5` – среда KDE5 для Альт Образование;
- `task-edu-lite` – базовый набор образовательного ПО, облегчённый для RPi4;
- `task-edu-preschool` – образовательное программное обеспечение (дошкольное образование);
- `task-edu-school` – образовательное программное обеспечение для школ;
- `task-edu-secondary-vocational` – образовательное программное обеспечение (среднее профессиональное образование);
- `task-edu-server-apps` – образовательное программное обеспечение (серверные приложения);
- `task-edu-teacher` – образовательное программное обеспечение (для учителей);
- `task-edu-tools` – вспомогательные программы для Альт Образование;
- `task-edu-university` – образовательное программное обеспечение (высшее образование);
- `task-edu-video-conferencing` – образовательное программное обеспечение (сервер видеоконференций).

Пример установки программного обеспечения для университета – пакеты `task-edu`, `task-edu-university`

```
su -  
apt-get install task-edu task-edu-university
```

Установка этих скриптов приводит к установке большого количества образовательных и научных приложений.

Создание установочных флешек

Для создания загрузочной (установочной) флешки можно воспользоваться графическим приложением `altmediawriter` (установка из репозитория `apt-get install altmediawriter`) или консольной утилитой `dd`.

Следует учитывать, что при создании загрузочной флешки данные, которые находились на USB устройстве, не сохраняются.

Для создания загрузочной флешки необходим файл образа (файл с расширением `iso`) дистрибутива ALT Linux, который можно скачать на сайте <https://getalt.org/ru/>. Для первого знакомства с ALT Linux рекомендуется использовать ALT Workstation, ALT Образование или Simply Linux.

Утилита `altmediawriter`

Рассмотрим как создать загрузочную флешку с помощью `altmediawriter`.

Утилита `altimagewriter` позволяет скачать файл образа одного из дистрибутивов ALT (Альт Рабочая Станция, АльтОбразоваание, Симпли Линукс и др.), а затем записать его на флешку. Следует учитывать, что современный дистрибутив занимает достаточно много места. Для скачивания файла образа должен быть стабильный канал Интернет. Если образ выкачан заранее и есть файл с расширением `iso`, следует выбрать режим Другой образ. После этого необходимо выбрать файл с образом и USB накопитель, на который будет записан образ (см. рис. 4). Утилита `altimagewriter` является универсальной, она создает загрузочную флешку с любым дистрибутивом Linux.

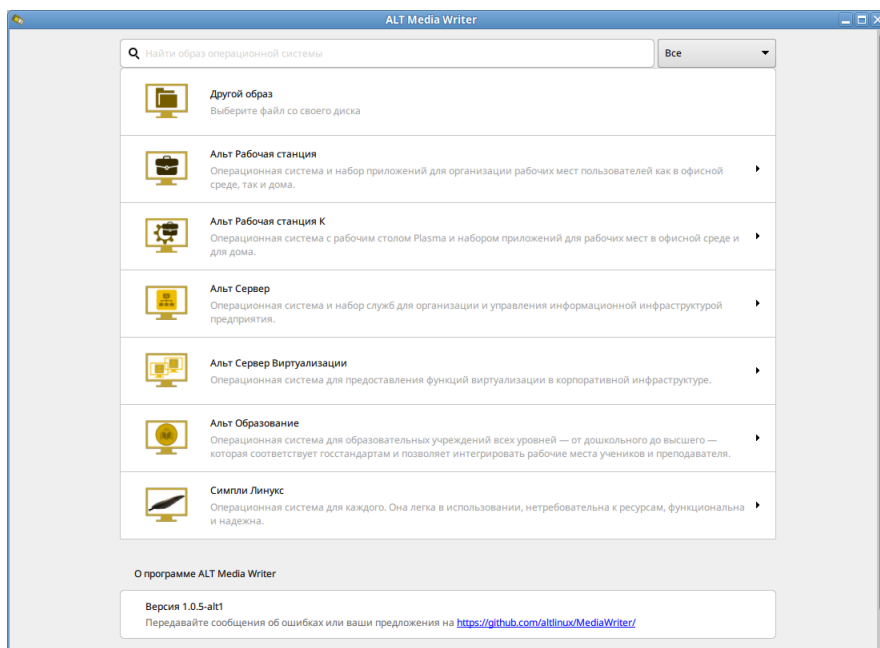


Рис. 8.3: Утилита altmediawriter

Консольная утилита dd

Консольная утилита `dd` предназначена для создания загрузочной флешки путем прямой записи `iso` файла на USB устройства.

Команду `dd` нужно выполнять от имени пользователя `root` (`su -`).

```
dd if=name.iso of=/dev/sdX status=progress
```

Здесь

`name.iso` – имя `iso` файла дистрибутива вашей операционной системы.

`/dev/sdX` – файл устройства флешки.

Имя файла флешки можно узнать с помощью утилит `df` и `fdisk`.

Рассмотрим результат вывода утилиты `df`.

```
(base) aer@aer Загрузки $ df
Файловая система  Размер  Использовано  Дост  Использовано%  Смонтировано в
udevfs             3,8G       96K        3,8G           1% /dev
runfs              3,8G      1,7M        3,8G           1% /run
/dev/sda6          114G       58G        50G          54% /
```

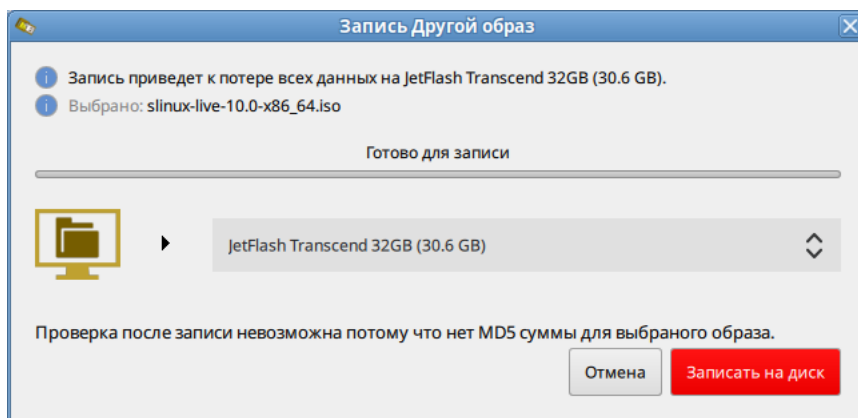


Рис. 8.4: Запись образа

tmpfs	3,8G	16K	3,8G	1%	/dev/shm
tmpfs	3,8G	11M	3,8G	1%	/tmp
/dev/loop0	172M	172M	0	100%	/var/lib/snapd/snap/codium/262
/dev/loop1	47M	47M	0	100%	/var/lib/snapd/snap/snapd/16292
/dev/loop2	56M	56M	0	100%	/var/lib/snapd/snap/core18/2538
/dev/sda4	510M	422M	89M	83%	/mnt/sdb4
/dev/sda3	107G	41G	67G	38%	/mnt/sdb3
/dev/sda1	96M	35M	62M	37%	/mnt/sdb1
/dev/sdb6	118G	8,0G	110G	7%	/mnt/sda6
/dev/sdb5	801G	690G	71G	91%	/DATA
tmpfs	776M	96K	776M	1%	/run/user/500
/dev/sdc1	5,7G	5,7G	0	100%	/run/media/aer/ALT Workstation 10.0 x86_64

Очевидно, что в данном случае файл устройства флешки – `/dev/sdc`.

Консольная утилита `fdisk` предназначена для разметки дисков вашего компьютера, поэтому не следует использовать её без ключа `-l` (просмотр разделов дисков). Утилита выполняется от имени администратора. Рассмотрим результат её выполнения.

```
su -
fdisk -l
Диск /dev/sdb: 931,51 GiB, 1000204886016 байт, 1953525168 секторов
Disk model: TOSHIBA MQ04ABF1
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 4096 байт
Размер I/O (минимальный/оптимальный): 4096 байт / 4096 байт
Тип метки диска: dos
Идентификатор диска: 0xa2bb2c02
Устр-во   Загрузочный   начало      Конец      Секторы    Размер    Идентификатор  Тип
/dev/sdb4                                4094 1953523627 1953519534 931,5G      5 Расширенный
/dev/sdb5          246147072 1953521663 1707374592 814,1G     83 Linux
```

```

/dev/sdb6          6144 246145023 246138880 117,4G      7 HPFS/NTFS/exFAT
Диск /dev/sda: 223,57 GiB, 240057409536 байт, 468862128 секторов
Disk model: TS240GMTS420S
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: gpt
Идентификатор диска: 32A24718-B0D7-4DB0-8D34-142FF84726E7
Устр-во      начало      Конец    Секторы Размер Тип
/dev/sda1      2048        206847    204800   100М EFI
/dev/sda2      206848      239615    32768    16М Зарезервированный раздел Microsoft
/dev/sda3      239616      224233672 223994057 106,8G Microsoft basic data
/dev/sda4      224235520   225279999 1044480   510М Среда для восстановления Microsoft
/dev/sda5      225280000   225492991 212992    104М EFI
/dev/sda6      225492992   468856831 243363840 116Г Файловая система Linux
Диск /dev/sdc: 28,48 GiB, 30579621888 байт, 59725824 секторов
Disk model: Transcend 32GB
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0хbc616086
Устр-во      Загрузочный  начало      Конец    Секторы Размер Идентификатор Тип
/dev/sdc1    *              64 11870207 11870144   5,7G      0 Пустой
/dev/sdc2              4308 16787    12480    6,1М      ef EFI (FAT-12/16/32)

```

В результатах вывода утилиты `fdisk -l` также можно найти информацию, что файл устройства флешки – `/dev/sdc`.

Для создания загрузочной флешки в операционной системе **Windows** разработчики **Alt Linux** советуют использовать `AltImageWriter` ([s://github.com/altlinux/ALTMediaWriter](https://github.com/altlinux/ALTMediaWriter)) или `Win32 Disk Image` (<https://sourceforge.net/projects/win32diskimager>). Альтернативой является широко известная в мире Windows утилита `Rufus` (<https://rufus.ie/ru/>).

Глава 9

Установка Linux на компьютер пользователя

Установка ОС Linux будет рассмотрена на примере дистрибутива Linux Mint, так как он прост в использовании и подходит для начинающих пользователей.

Для того чтобы установить дистрибутив Linux (Linux Mint или любой другой) необходимо скачать файл **iso** (образ дистрибутива), записать образ на флешку, загрузиться в Live режиме, запустить в Live режиме инсталлятор ОС и следовать инструкциям на экране. Теперь рассмотрим процесс установки более подробно.

Получение образа ОС Linux

Ссылки для скачивания современных дистрибутивов:

- Debian – <https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/> или <https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/> (версия с несвободными микропрограммами);
- LinuxMint – https://linuxmint.com/download_all.php;
- Ubuntu – <https://ubuntu.ru/get>;
- Ubuntu Mate – <https://ubuntu-mate.org/ru/download/>;

- ОС семейства ALT – <https://getalt.org/ru/>;
- Астра Линукс – <https://dl.astralinux.ru/astra/stable/>;
- Runtu – <https://runtu.org/download/>;
- Red OS – <https://redos.red-soft.ru/product/downloads/>;
- GreenLinux – <https://greenlinux.ru/download/>.

После скачивания актуальной версии дистрибутива в формате iso, необходимо создать загрузочную флешку.

Запись образа на USB

Для записи образа ОС на USB под управлением Windows необходима сторонняя программа Rufus. Скачать её можно с сайта <https://rufus.ie/ru>. При записи образа на USB, устройство форматируется. Все важные данные необходимо сохранить на другом носителе. После запуска программы необходимо выбрать (рис. 9.1):

1. Носитель (в графе Устройство).
2. ISO файл (щелкнуть по кнопке «Выбрать»).
3. Схему разделов: GPT или MBR (в современных устройствах чаще GPT).
4. Нажать «Старт»

И далее подтвердить форматирование устройства.

Установка Linux рядом с Windows

Чтобы установить Linux рядом с Windows необходимо предварительно выделить на диске место, которое будет использоваться ОС Linux. Для этого нужно через меню **Пуск**

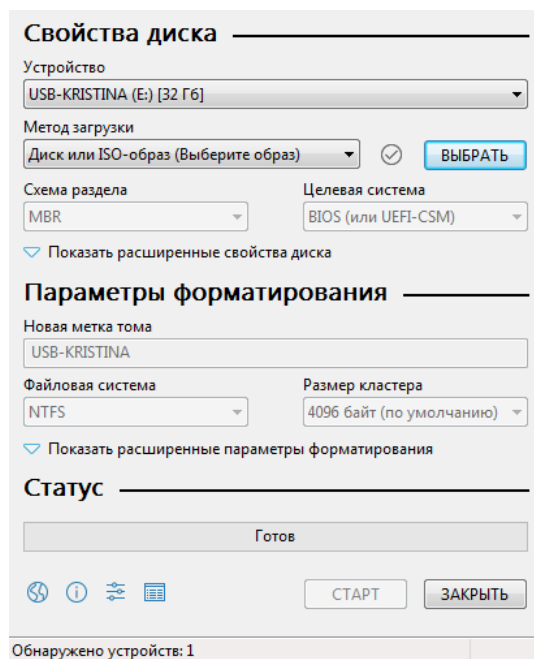


Рис. 9.1: Утилита rufus

найти утилиту **Управление компьютером**. В списке слева выбрать пункт **Управление дисками** (рис. 9.2), после чего появится таблица со списком разделов. От раздела, которым есть свободное место нужно отрезать 30-50 Gb для будущей ОС. Для этого правой кнопкой мышки щелкнуть по разделу и во всплывающем меню выбрать **Сжать том** (рис. 9.3).

В поле «Размер сжимаемого пространства» нужно указать, сколько места будет отрезано от раздела. Нажать кнопку «сжать».

Итог: в таблице разделов появится пустая область указанного размера.

Модификацию размеров диска можно осуществить и в процессе установки Linux.

Загрузка LiveCD

Чтобы загрузить ОС в Live режиме, нужно зайти в BIOS/UEFI и изменить порядок загрузки. На первое место поставить загрузку с USB.

Вход в BIOS на разных компьютерах осуществляется

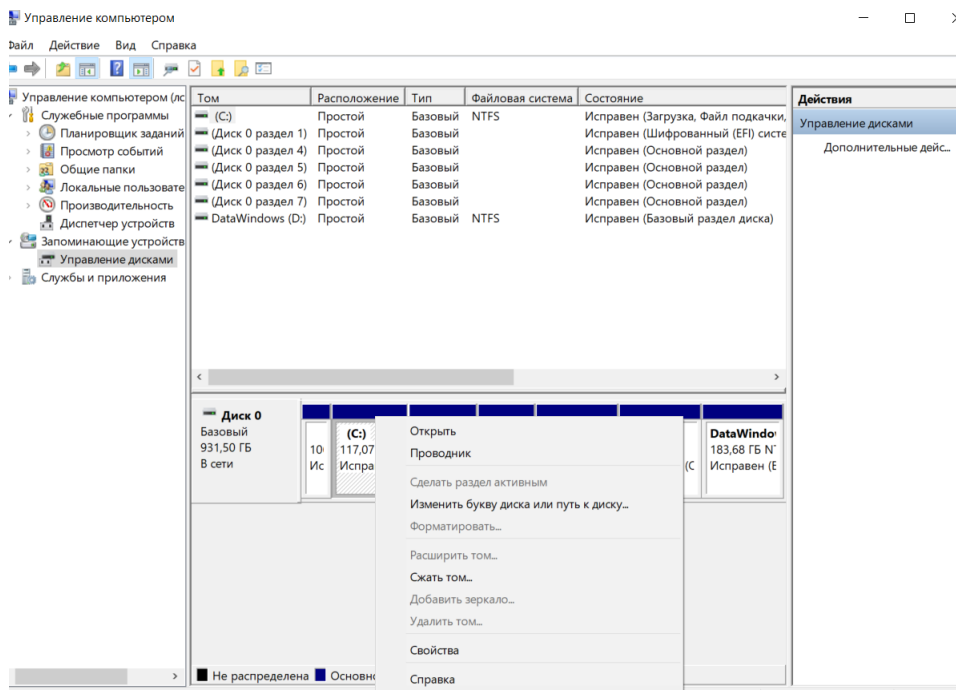


Рис. 9.2: Управление дисками

по разному. Необходимо при включении компьютера удерживать определённую клавишу, но у разных производителей она отличается. Иногда это: F1, F2, F10, F11, Esc, Del. После изменения порядка загрузки требуется сохранить изменения и перезагрузить компьютер.

Загрузочная флешка должна быть вставлена в USB порт до перезагрузки. После перезапуска начнётся загрузка Linux в Live режиме¹, появится загрузочное меню. Выбрать пункт «Start Linux...», после чего появится рабочий стол (рис. 9.4).

Установка Linux

Для начала установки системы на компьютер нужно дважды щелкнуть по иконке **Install Linux Mint**, которая расположена на рабочем столе.

После этого запускается инсталлятор ². В открывшем-

¹Если Вами выбран дистрибутив без live режима, то сразу начнётся процесс установки, описаны, начиная со страницы 58.

²Здесь и далее представлен процесс установки Linux Mint, но логика и

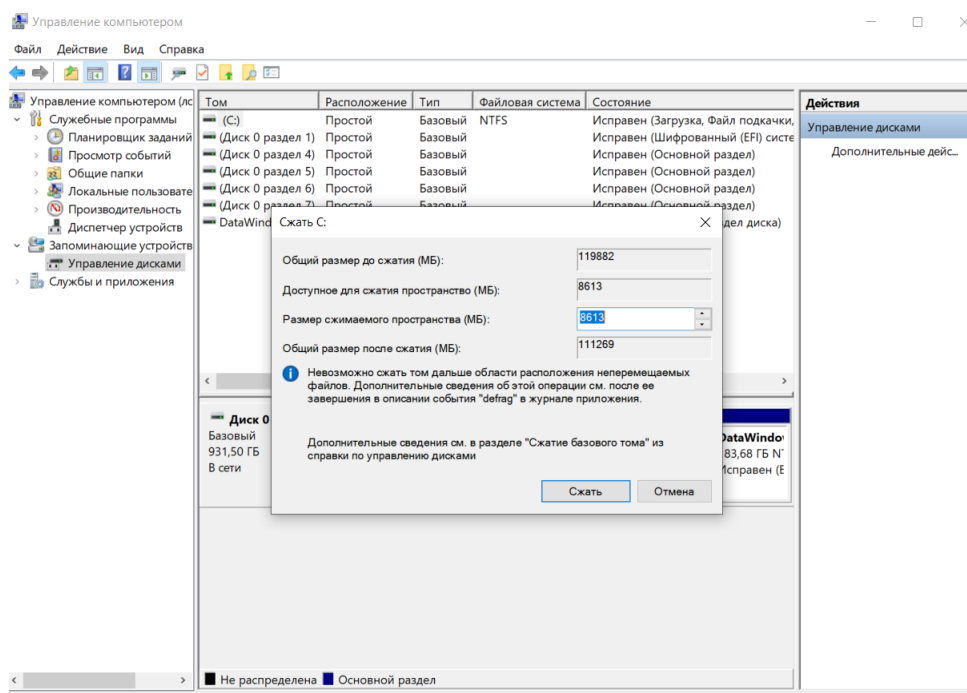


Рис. 9.3: Сжатие тома

ся окне необходимо последовательно выбрать язык будущей системы, установить дополнительные компоненты (драйвера видео-карты, WiFi, поддержка Flash, MP3 и др.), раскладку клавиатуры (рис. 9.5-9.7).

После чего в окне появится список типов установки. Выбрать «Другой вариант» для того, чтобы можно было самостоятельно разметить диск.

Особенности разметки диска при установке Linux

Создаваемые на диске разделы могут отличаться размером, файловой системой, назначением. Linux может занимать несколько разделов, каждый из которых служит для своих целей.

1. / («корень») – основной (корневой) раздел, в котором размещаются все каталоги системы;
2. /home – домашний раздел, в котором размещаются пользовательские данные;

основные принципы установки в других дистрибутивах похожи.

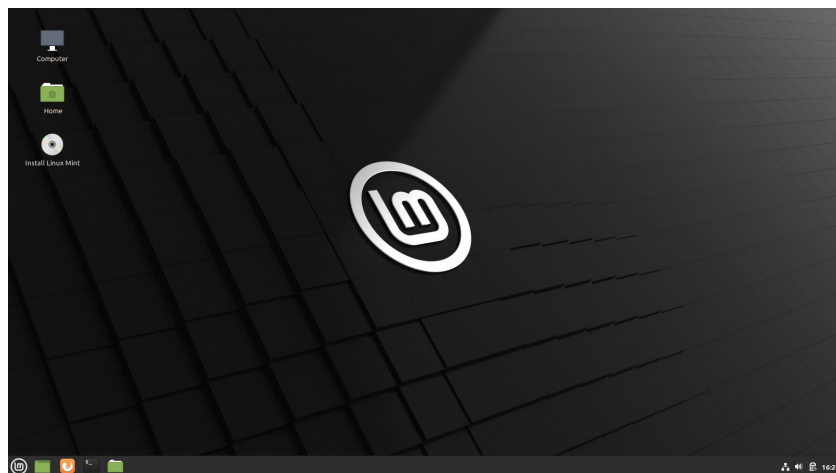


Рис. 9.4: Рабочий стол Linux Mint



Рис. 9.5: Выбор языка установки

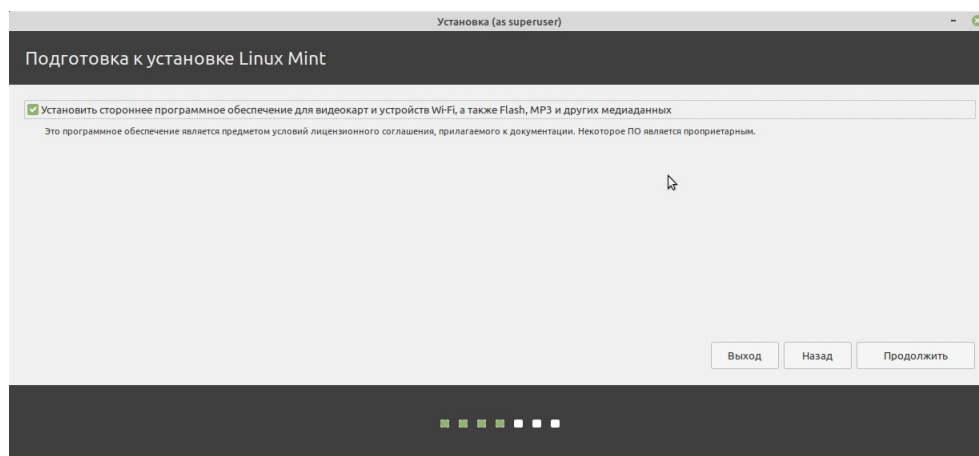


Рис. 9.6: Дополнительные компоненты при установке

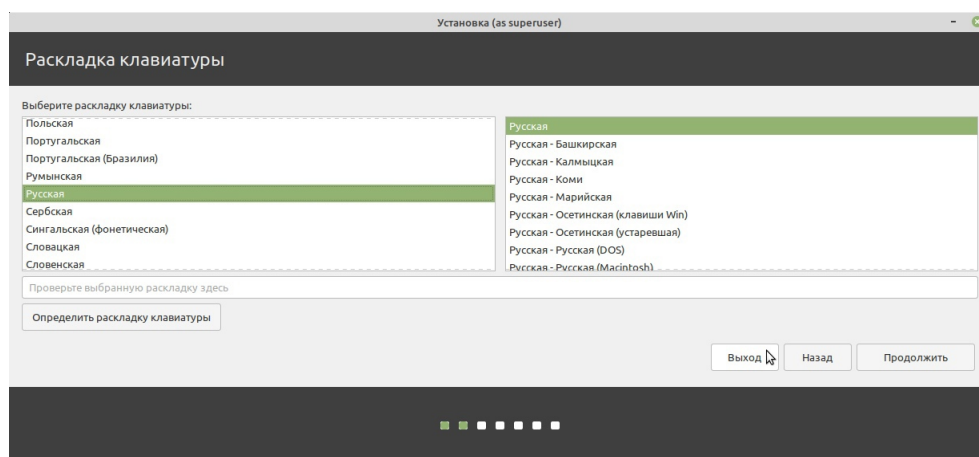


Рис. 9.7: Выбор раскладки клавиатуры

3. swap - раздел подкачки, используется системой, если не хватает оперативной памяти.

Жесткий диск обычно обозначается как `/dev/sda`. В случае, когда дисков несколько, названия будут `/dev/sdb`, `/dev/sdc` и т. д. В списке уже будет несколько разделов (разделы других операционных систем) (см. рис. 9.8).

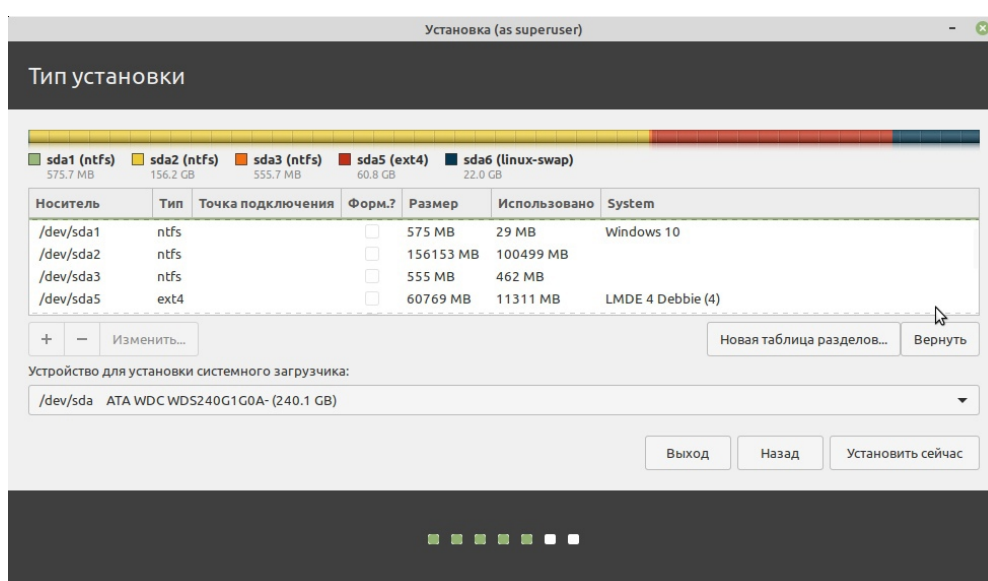


Рис. 9.8: Разделы диска

Если Вы предварительно удалили ненужные разделы или уменьшали какой либо раздел, то в карте диска будет строка «свободное место». В этом случае Вам можно будет создавать новые разделы, используя кнопку **+**.

Если же свободного места нет, то придётся освободить его на каком либо разделе прямо сейчас, используя кнопку **Изменить**. После чего появится на карте диска появится свободное место.

Рассмотрим более подробно процесс создания необходимых разделов. Начнём с раздела подкачки (swap-раздела).

В таблице разделов выделить поле «свободное место» и нажать кнопку «+». В настройках:

1. Указать размер раздела в мегабайтах. Размер раздела подкачки зависит от параметров компьютера. В среднем 4-16 Gb. Рекомендуется отводить для раздела подкачки 1-2 объёма оперативной памяти¹.
2. Местоположение нового раздела. Начало этого пространства.
3. Использовать как: раздел подкачки.

Теперь нужно создать корневой раздел «/» и, возможно, раздел «/home». Алгоритм одинаковый для обоих разделов.². Для этого в таблице разделов снова выделить поле «свободное место» и нажать «+». В настройках: (см. рис. 9.9):

1. Указать размер раздела в мегабайтах (минимум 20 Gb для «/» и оставшееся для «/home», если он планируется как отдельный раздел).
2. Местоположение нового раздела: Начало этого пространства.
3. Использовать как: Журналируемая файловая система Ext4.
4. Точка монтирования: / (/home) .

На современных компьютерах должен существовать загрузочный раздел EFI (файловая система FAT32, размер 100-300 Мб). Поэтому при разметке диска для современной версии Linux, нужно либо создать его, либо проверить его наличие в другой операционной системе. (рис. 9.10).

¹Если у Вас 8 Гб ОЗУ и более, то можно и не использовать раздел подкачки.

²Если раздел /home не создавать, то каталог /home будет создан в корневом разделе

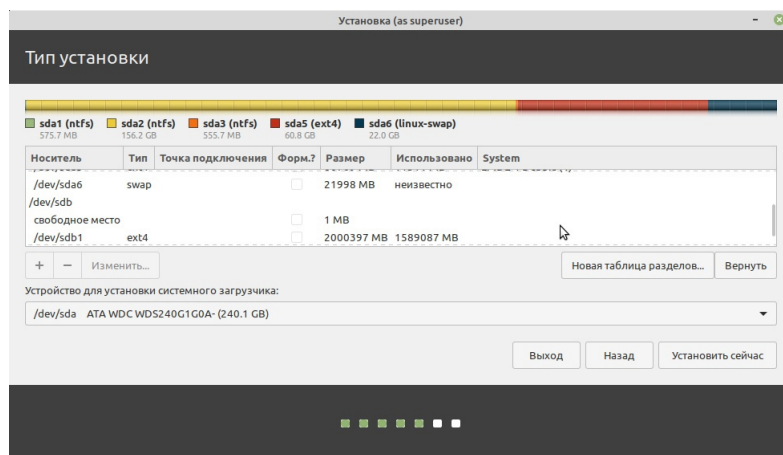


Рис. 9.9: Создание корневого раздела

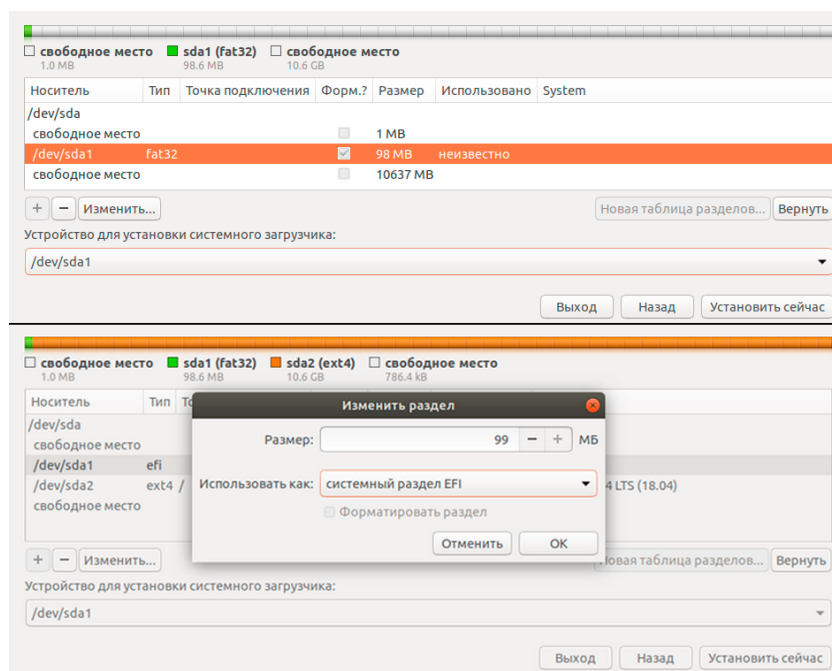


Рис. 9.10: Раздел EFI

После этого можно завершать разметку диска.

Далее необходимо выбрать часовой пояс (рис. 9.11) и создать пользователя (рис.9.12).

При создании пользователя указать:

1. Ваше имя.
2. Имя Вашего компьютера.
3. Имя пользователя.
4. Пароль и повтор пароля.

Далее выбрать «Требовать пароль для входа в систему» или «Входить в систему автоматически», если при входе ввод пароля не нужен.

Есть возможность выбрать опцию «Шифровать мою домашнюю папку», но начинающему пользователю этого делать не нужно.

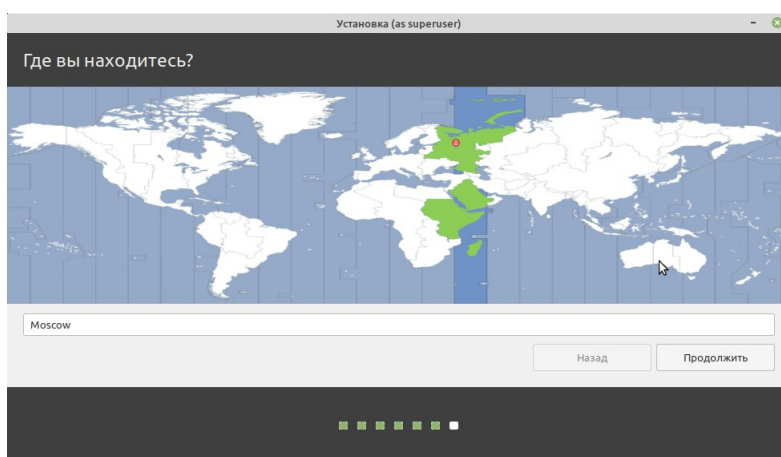


Рис. 9.11: Выбор часового пояса

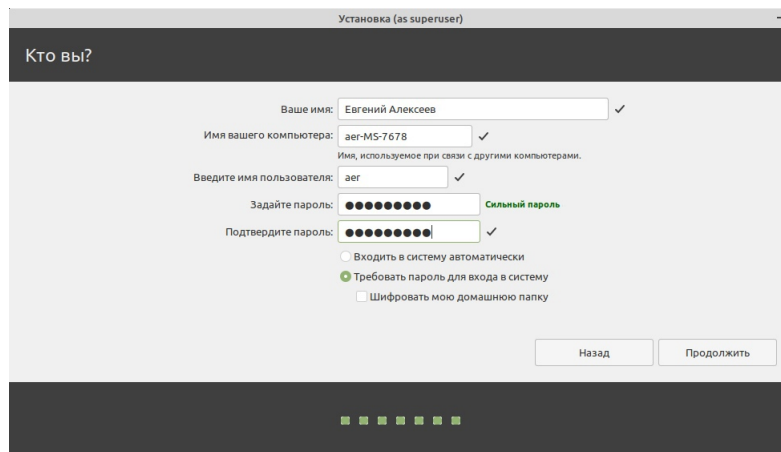


Рис. 9.12: Создание пользователя

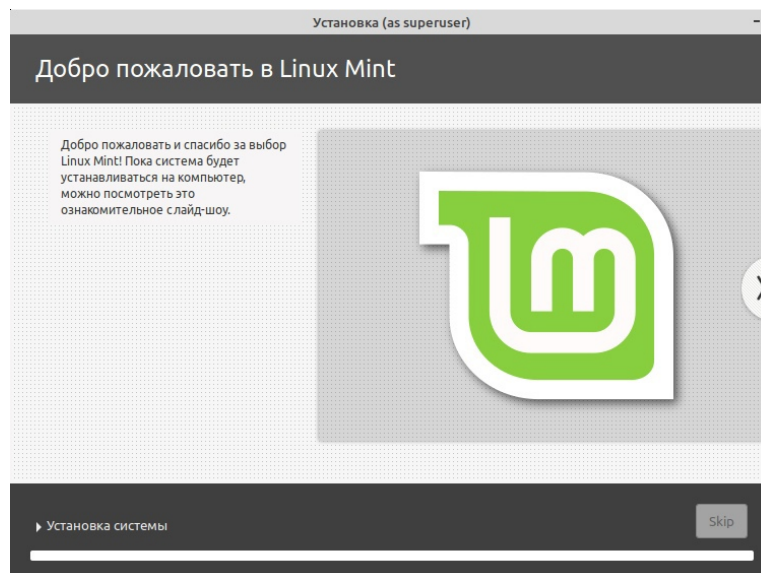


Рис. 9.13: Установка системы

После указания всех настроек нажать кнопку «Продолжить». Начинается процесс установки (рис. 9.13)).

После его завершения можно перезагрузить компьютер. При перезагрузке в меню запуска будет автоматически выбран **Linux**. Для входа в **Windows** нужно самостоятельно выбрать пункт «**Windows Boot Manager**» (название может быть другим).

Видео установки Linux Ubuntu можно увидеть тут https://vk.com/video/@aer64?z=video7785282_456239349, Здесь <https://disk.yandex.ru/i/6fhZCt2zQYM0zw> находится видео по установке дистрибутивов семейства **Альт**.

Настройка начального загрузчика

Одной из первых проблем, с которой сталкивается начинающий пользователь, является настройка начального загрузчика **Grub**.

После установки **Linux** на компьютер, с установленной ОС **Windows**, по умолчанию загружается **Linux**, и есть четыре варианта загрузки (количество вариантов может отличаться):

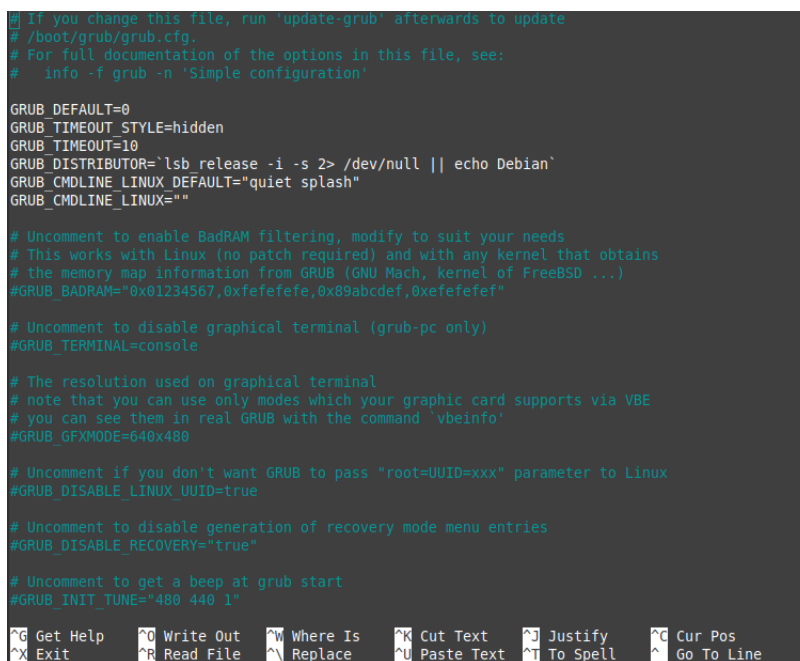
1. Загрузка ОС **Linux**.
2. Загрузка ОС **Linux** в безопасном режиме.
3. Тест оперативной памяти.
4. Загрузка **Windows**.

По умолчанию окно загрузчика видно 10 секунд, в течении которых нужно выбрать операционную систему для загрузки. Эти параметры можно изменить, они настраиваются в файле `/etc/default/grub`.

Файл **grub** относится к файлам, которые управляют работой операционной системы, поэтому редактировать его может только администратор.

Необходимо вызвать терминал командой Пуск – Системные – Терминал. В открывшемся окне вызвать следующие команды:

`sudo nano /etc/default/grub` (вызван текстовый редактор для редактирования файла **grub**).



```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
```

Рис. 9.14: Редактирование файла grub

После ввода пароля откроется окно (см. рис. 9.14), в котором можно изменить следующие параметры:

- **GRUB_DEFAULT** - изменение порядка загрузки ОС (указывается номер строки в загрузчике). Важно заметить, что нумерация строк идёт с 0, поэтому, если Windows в списке загружаемых систем находится на 4 строке, то в параметре указывается 3.
- **GRUB_TIMEOUT** - изменение времени. По умолчанию поль-

зователь видит загрузчик 10 секунд. Это можно изменить, указав здесь другое число (в секундах).

После внесения изменений нажать **Ctrl+X** для выхода из редактора. После чего подтвердить сохранения, нажав клавиши **Y** и **Enter**. Затем обязательно вводим команду

```
sudo update-grub
```

для обновления загрузчика. После этого можно перезагружать компьютер.

Глава 10

Лабораторный практикум «Основы работы в Linux»

Лабораторная работа №1

Установите ОС семейства Linux (предпочтительно отечественные ALT Linux, Green Linux, либо Debian, Ubuntu, Linux Mint) на свой ноутбук (стационарный компьютер).

Лабораторная работа №2

Установите пакеты полной русификации man (deb пакеты русификации находятся здесь https://disk.yandex.ru/d/oyeIzh586_qwSA и <https://disk.yandex.ru/d/LV0VS5UTN9AmCw>). Получите справочную информацию о командах и перенаправьте ее в файл lab2.txt.

1. whatis, touch
2. apropos, cat
3. cp, file
4. mv, adduser
5. which, find

6. `tac, id`
7. `rm, useradd`
8. `less, chmod`
9. `mkdir, chown`
10. `dpkg, chgrp`
11. `ls, su`
12. `rmdir, cal`
13. `apt, locate`
14. `sudo, apt install`
15. `pwd, echo`

Лабораторная работа №3

1. Создать каталог `Var`(номер варианта).
2. В каталоге создать дерево каталогов как показано на рисунке (рис. 10.1 - 10.15).
3. В каталоге `A2` создать подкаталоги `B4` и `B5` и удалить каталог `B2`.
4. В каталоге `Personal` создать файл **`name.txt`**, содержащий информацию о фамилии, имени и отчестве студента. Здесь же создать файл **`Date.txt`**, содержащий информацию о дате рождения студента. В этом же каталоге создать файл **`school.txt`**, содержащий информацию о школе, которую закончил студент.
5. В каталоге `University` создать файл **`name.txt`**, содержащий информацию о названии Вуза и специальность, на которой студент обучается.

6. В каталоге **Hobby** создать файл **hobby.txt** с информацией об увлечениях студента.
7. Скопировать файл **hobby.txt** в каталог **A2**.
8. Объедините файлы **hobby.txt**, **name.txt**, **school.txt** в файле **all.txt** каталога **Info**.
9. Удалите файлы **hobby.txt**, **name.txt**, **school.txt**.
10. Выведите содержимое файла **all.txt** на экран.
11. Скопировать файл **all.txt** в директорию **A1**.

Лабораторная работа №4

Используя задание №2, выполните:

1. Для **Info** установите права доступа «только чтение».
2. Перейдите в каталог **Var** (номер варианта) и выполните:
 - 2.1. Создайте каталог **test** и перейдите в него.
 - 2.2. Создайте файл **file**, определите его права доступа и запретите любые действия с файлом. Попробуйте записать строку «тест» в **file**. Затем попробуйте прочитать файл.
 - 2.3. Измените права доступа для **file** так, чтобы только текущий пользователь мог осуществлять в него запись. Запишите строку «test» в файл. Попробуйте прочитать содержимое.
 - 2.4. Разрешите группе текущего пользователя читать содержимое файла. Попробуйте прочитать содержимое.
 - 2.5. Разрешите текущему пользователю читать содержимое файла

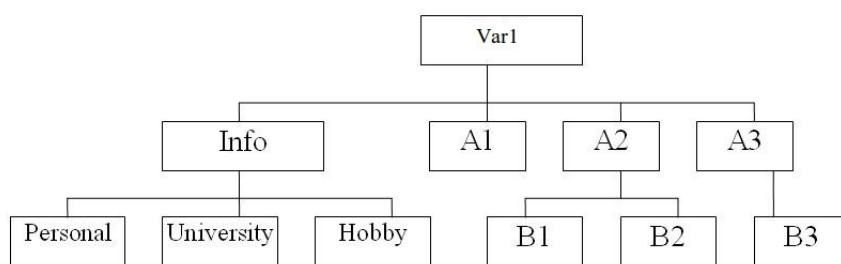


Рис. 10.1: Вариант 1

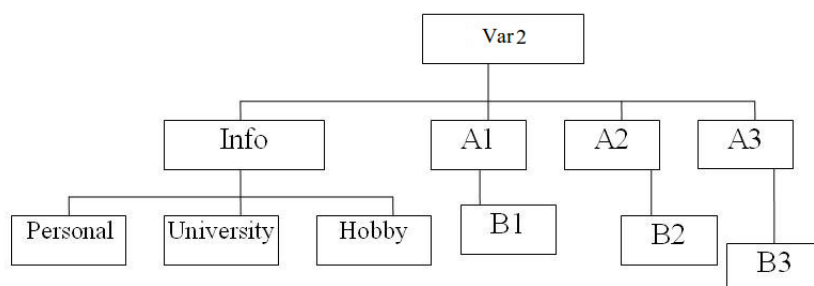


Рис. 10.2: Вариант 2

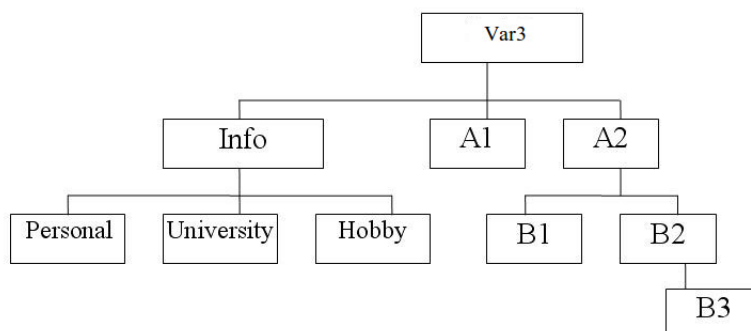


Рис. 10.3: Вариант 3

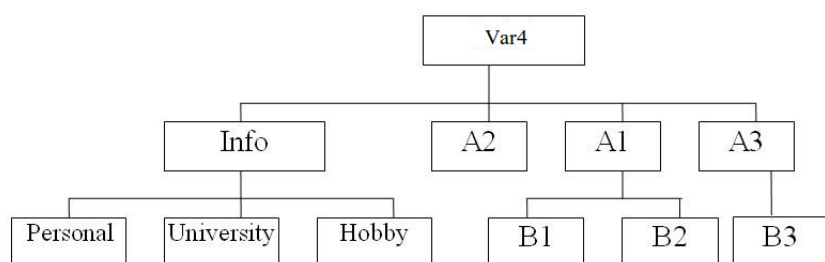


Рис. 10.4: Вариант 4

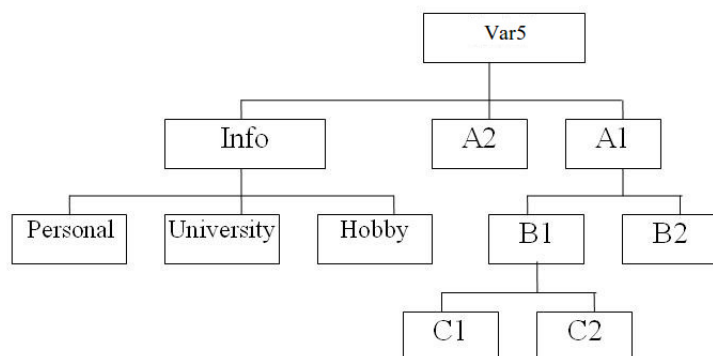


Рис. 10.5: Вариант 5

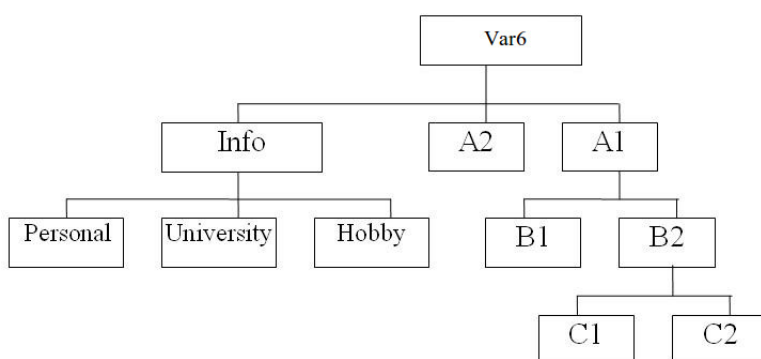


Рис. 10.6: Вариант 6

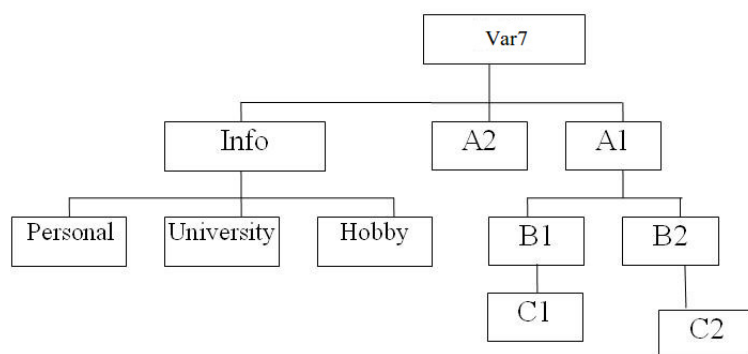


Рис. 10.7: Вариант 7

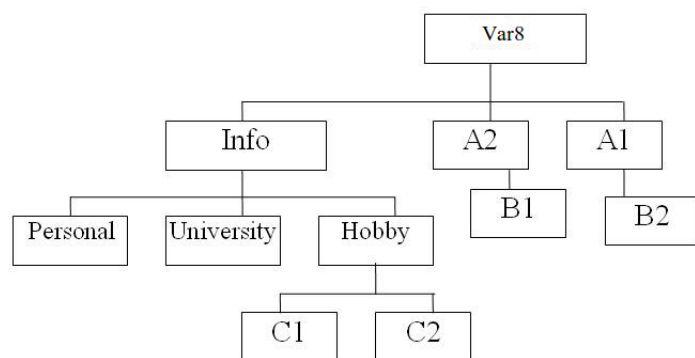


Рис. 10.8: Вариант 8

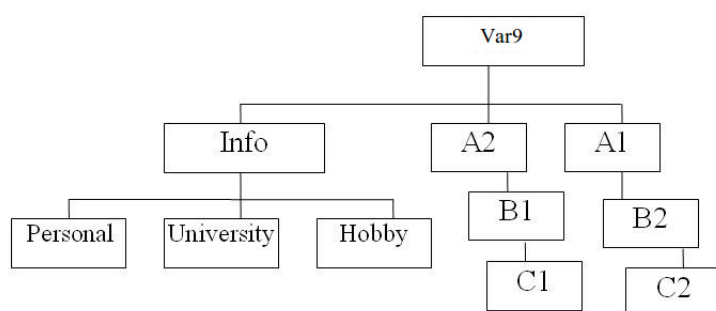


Рис. 10.9: Вариант 9

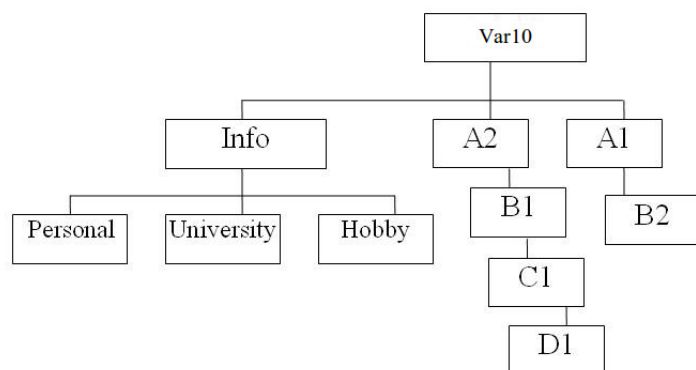


Рис. 10.10: Вариант 10

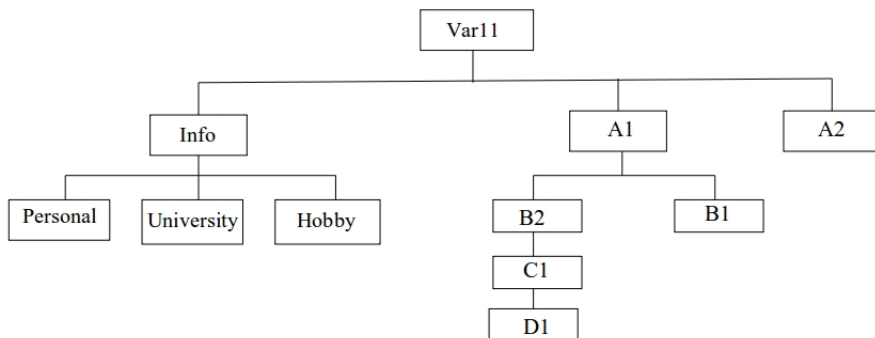


Рис. 10.11: Вариант 11

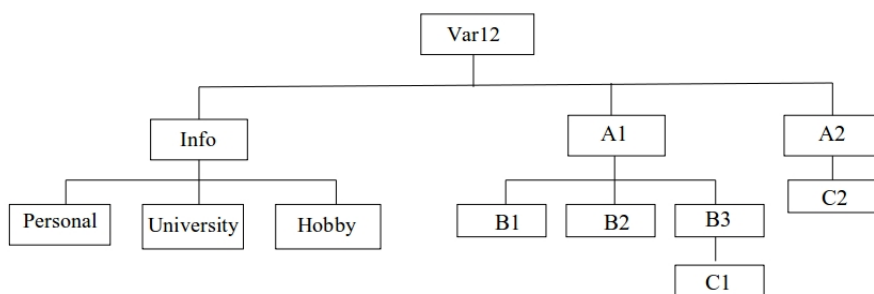


Рис. 10.12: Вариант 12

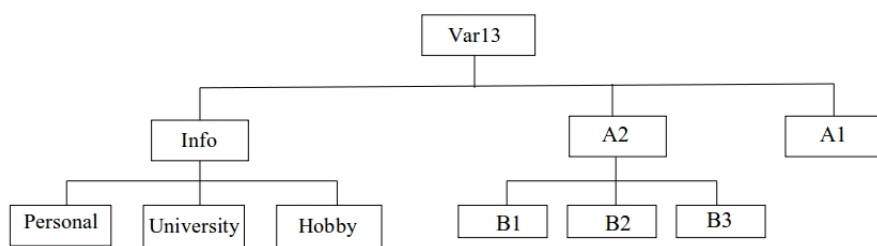


Рис. 10.13: Вариант 13

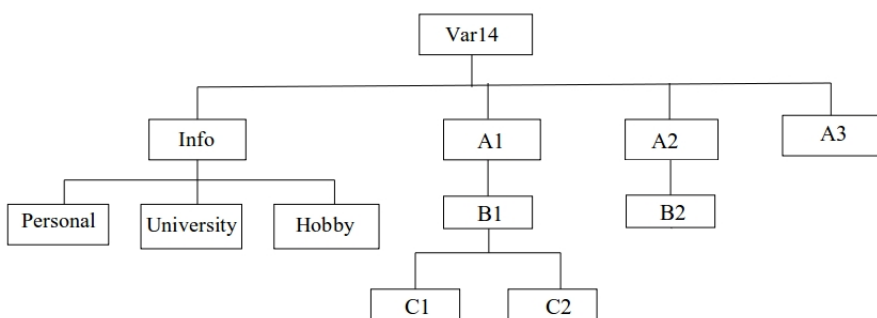


Рис. 10.14: Вариант 14

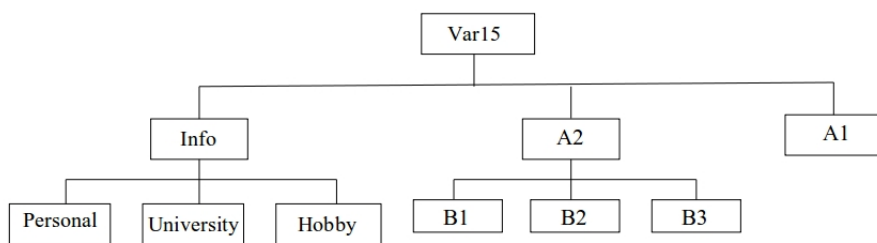


Рис. 10.15: Вариант 15

- 2.6. Попробуйте прочитать содержимое – должна появиться ранее добавленная туда строка «*test*».
- 2.7. Создайте каталог **dir**. Создайте в этом каталоге файл **newfile**, запишите в него текст «*file2*». Просмотрите содержимое каталога.
- 2.8. Измените права доступа на каталог **dir** так, чтобы пользователи не имели права на исполнение (**x**). Попробуйте просмотреть содержимое файла **newfile** в каталоге **dir**. Попробуйте удалить этот файл.
3. Создайте в **Var** каталог **Primer**, который будет содержать текстовые файлы **primer1.txt** и **primer2.txt**.
4. Выполните далее задание своего варианта:
 - 4.1. Установите права **441** каталогу **Primer**.
 - 4.2. Установите файлу **primer1.txt** права **454**, удалите его.
 - 4.3. Установите каталогу **Primer** и его содержимому права **344**.
 - 4.4. Установите всем файлам права **234**.
 - 4.5. Установите права **455** файлу **primer2.txt**.
 - 4.6. Установите права **543** каталогу **Primer**.
 - 4.7. Установите каталогу **Primer** права **552**.
 - 4.8. Установите каталогу **Primer** права **447**.
 - 4.9. Установите каталогу **Primer** права **444**.
 - 4.10. Установите обоим файлам права **333**.
 - 4.11. Установите файлу **primer1.txt** права **411**.
 - 4.12. Установите созданным файлам права **300**.
 - 4.13. Установите каталогу **Primer** и его содержимому права **411**.

4.14. Установите каталогу `Primer` права **432**.

4.15. Установите каталогу `Primer` права **321**.

Лабораторная работа №5

Найдите файл (файлы):

- с именем на Ваш выбор;
- размер которых меньше 300 байт;
- модифицированные больше четырех дней назад;
- пустые;
- определенного типа (`f` – обычный файл).

Лабораторная работа №6

1. Изучить назначение и ключи команды `ln`.
2. Создайте файл с описанием команд `dpkg`, `apt`, `apt-get`, `ln`.
3. Создайте жесткую ссылку на него.
4. Просмотрите содержимое файла, используя ссылку.
5. Создайте символическую ссылку на исходный файл.
6. С помощью команды `ls` просмотрите `inod` файла и ссылок.

Лабораторная работа №7

1. Пользуясь командами терминала, установите пакет `ttf-mscorefonts-installer`. Каково назначение этого пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.

2. Пользуясь командами терминала, установите пакет `dia`. Каково назначение этого пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
3. Пользуясь командами терминала, установите пакет `wxmaxima`. Каково назначение этого пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
4. Пользуясь командами терминала, установите пакет `g++`. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
5. Пользуясь командами терминала установите пакет `geany`. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
6. Пользуясь командами терминала, установите пакет `gfortran`. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
7. Скачайте пакет `scilab` и установите его, пользуясь командами терминала. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
8. Пользуясь командами терминала, установите пакет `fpc`. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
9. Пользуясь командами терминала, установите пакет `gimp`. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
10. Пользуясь командами терминала установите пакет `Transmission`. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.

11. Пользуясь командами терминала, установите пакет **octave**. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
12. Пользуясь командами терминала, установите пакет **Mathomatic**. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
13. Пользуясь командами терминала, установите пакет **lazarus**. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
14. Пользуясь командами терминала, установите пакет **Cantor**. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.
15. Пользуясь командами терминала, установите пакет **vlc**. Каково назначение пакета? Создайте текстовый файл **Lab7** с ответом на вопрос.

Лабораторная работа №8

1. Создайте пользователя **labuser** с настройками по умолчанию. Задайте **labuser** пароль. В каких группах состоит **labuser**? Войдите под новым пользователем в систему. Проверьте, создан ли домашний каталог пользователя, наполнен ли он файлами и кому он принадлежит?
2. Удалите пользователя **labuser**.
3. Создайте пользователя со своим именем и паролем (с правами **sudo**). Зайдите под новым пользователем в систему. Просмотрите записи для нового пользователя в файлах **/etc/passwd** и **/etc/shadow**.
4. Определите владельца и группу файла **/etc/passwd**.
5. Выведите на экран **UID** и **GID**.

6. Смените пароль у пользователя с административными правами.

Сетевые возможности ОС Linux

Глава 11

Общие сведения о сетевых протоколах

Интернет — это сеть, связывающая устройства по всему миру посредством многочисленных, связанных друг с другом, более мелких сетей. С другой стороны, интернет — архитектура, выраженная набором протоколов.

Протокол — это некий заранее оговоренный набор правил и стандартов, которому все договорились следовать, чтобы без проблем связываться друг с другом.

Глобальные и локальные сети

Сеть в интернете может быть глобальной и локальной (WAN и LAN соответственно). Все устройства в пределах квартиры или офиса подключаются к маршрутизатору (роутеру, сетевому шлюзу — это одно и то же), тем самым объединяясь в локальную сеть. В рамках этой локальной сети устройства могут обмениваться данными друг с другом, однако иногда требуется выйти в интернет, а это уже глобальная сеть.

Для простоты понимания сравним сети с дорогами. Дороги одного города — это локальная сеть. Она связывает нас с домами, магазинами, местами работы и прочими объектами города. Но чтобы попасть в другой город, нужно проехать по федеральной трассе, а это в нашем случае и есть глобальная сеть.

Глобальные и локальные IP-адреса

Давайте вспомним, как работает почта. Чтобы послать письмо, нужно указать точный адрес, по которому оно будет доставлено. Так же необходимо указать адрес отправителя, чтобы адресат знал, куда направить ответ. У каждого дома есть свой полный адрес. Например, Россия, ДНР, г. Донецк, пр. Комсомольский, д. 52, кв. 32. Точно так же у каждого устройства в сети есть свой адрес, разделенный на 4 части точками. Например, 192.168.1.2

Система адресов — часть одного из самых важных протоколов интернета, который называется (**internet protocol** – **IP**). Адрес компьютера в интернете — **IP** адрес. Когда вы просматриваете веб-сайт, ваш компьютер обращается к другому компьютеру (серверу) с запросом на информацию (в нём он указывает свой обратный адрес, чтобы было понятно, кому отдать ответ на запрос).

IP — это набор чисел, выстроенных иерархически. Как и любые данные в цифровом мире, каждое из этих чисел представлено в двоичном виде. Стандартный **IP** имеет длину **32** бита по **8** бит на каждую часть адреса. Первые числа означают страну и региональную сеть, затем подсети и, наконец, адрес конкретного устройства. Эта версия **IP**-адресации называется **IPv4**. В ней возможно выделить свыше 4 миллиардов уникальных адресов, однако со временем этого оказалось недостаточно. Сейчас осуществляется многолетний переход к новому формату **IP** адресов: **IPv6**. В нём каждый адрес кодируется 128 битами и количество уникальных адресов, конечно, многократно увеличилось. Итак, **IP**-адрес необходим для связи с другими устройствами в сети. Чтобы узнать локальный **IP** своего устройства, в терминале введите команду

```
$ sudo ifconfig | grep "inet addr"
```

или

```
$ sudo ifconfig
```

Узнать свой глобальный IP немного сложнее. Нужно зайти на такой сайт, который увидит, с какого IP-адреса мы его открыли, и выведет нам требуемые числа. Подобных сайтов несколько. Вот команды, обращающиеся к разным сайтам для получения внешнего IP

```
$ wget -qO- eth0.me
$ wget -qO- ipinfo.io/ip
$ wget -qO- ipecho.net/plain
$ wget -qO- icanhazip.com
$ wget -qO- ipecho.net
$ wget -qO- ident.me
$ wget -qO- myip.gelma.net
```

Перед выходом в интернет роутер подменяет локальный IP устройства на глобальный (выданный интернет-провайдером).

Кроме IP, у компьютера есть MAC-адрес (физический), который задается производителем сетевой карты. Он используется для связи компьютеров в локальной сети.

DHCP-сервер и подсети

Чтобы выйти в интернет с компьютера, мы подключаем один конец провода (витой пары, кабеля — это одно и то же) к компьютеру, а второй — в свободный порт роутера. После этого устройство автоматически получает IP и подключается к глобальной сети.

Что касается **wi-fi**, IP-адрес появляется после ввода пароля (если он есть) или же сразу после подключения к самой сети **wi-fi**.

Устройство получает локальный IP автоматически благодаря DHCP-серверу, он есть в каждом маршрутизаторе. DHCP (англ. **D**ynamic **H**ost **C**onfiguration **P**rotocol - протокол динамической настройки узла) - прикладной протокол, позволяющий сетевым устройствам автоматически получать IP-адрес.

Таким образом IP-адреса, которые получены автоматически называются динамическими, потому что они имеют свойство меняться. Например, при каждом новом подключении или перезагрузке маршрутизатора. Можно задать статический IP, ровно как и отключить DHCP вообще (тогда адреса задаются вручную). Если задать несколько DHCP-серверов на одном маршрутизаторе, то мы получим сеть, разделённую на подсети. Обычно это нужно в компаниях с большим количеством устройств.

DNS-сервер

Как правило, пользователи не сталкиваются с IP адресами. Для этого есть система доменных имён DNS, которая сопоставляет имена компьютеров с их IP адресами. Ваш компьютер при помощи системы **dns**-серверов находит IP адрес, связанный с именем сайта. И уже по адресу связывается с компьютером, к которому вы хотите обратиться. DNS сервера объединены в иерархическую систему и разделены на зоны. Каждая зона отвечает за свой корневой домен (**.org .com .net** и т.д.)

Сетевой интерфейс – это набор аппаратных и программных инструментов для работы с сетью.

Глава 12

Общий доступ к каталогам с помощью Samba

Основные сведения о Samba

Samba <https://www.samba.org> — пакет программ, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу **SMB/CIFS**. Является свободным программным обеспечением. **Samba** выпущена под лицензией **GPL**, включена практически во все дистрибутивы **Linux**¹.

Samba состоит из клиентской и серверной части:

1. Клиентская часть позволяет получить доступ к сетевым папкам и ресурсам.
2. Серверная открывает общий доступ к папке **Linux** для других машин, в том числе и **Windows**.

Для установки запускаем команду от имени администратора (с правами **root**).

```
sudo apt-get install -y samba samba-common
```

¹Данное пособие ориентировано на дистрибутивы семейства **Debian**, **Ubuntu** и их производные. Команды протестированы в **Linux Ubuntu 20.04 (22.04)**, **Linux Mint 20(21)**.

После установки будет автоматически запущен демон `smbd`, обеспечивающий доступ к файлам и принтерам. Для проверки его работоспособности необходимо выполнить команду:

```
systemctl status smbd
```

Результаты работы представлен на рис. 12.1.

```
smbd.service - Samba SMB Daemon
Loaded: loaded (/lib/systemd/system/smbd.service; enabled; vendor preset:
enabled)
Active: active (running) since Sun 2019-11-03 13:55:35 MSK; 39min ago
Docs: man:smbd(8)
      man:samba(7)
      man:smb.conf(5)
Main PID: 11570 (smbd)
Status: "smbd: ready to serve connections..."
Tasks: 4 (limit: 4915)
CGroup: /system.slice/smbd.service
├─11570 /usr/sbin/smbd --foreground --no-process-group
├─11572 /usr/sbin/smbd --foreground --no-process-group
├─11573 /usr/sbin/smbd --foreground --no-process-group
└─11574 /usr/sbin/smbd --foreground --no-process-group
ноя 03 13:55:34 aer-MS-7678 systemd[1]: Starting Samba SMB Daemon.
ноя 03 13:55:35 aer-MS-7678 systemd[1]: Started Samba SMB Daemon.
```

Рис. 12.1:

По умолчанию этот демон включен, поэтому он будет запускаться вместе с операционной системой.

Общий каталог в Сети под управлением Linux

Универсальный способ настройки общего каталога в Linux (да и вообще в Unix) – редактирование файла

`/etc/samba/smb.conf`. Предварительно имеет смысл сделать копию этого файла.

Рассмотрим некоторые его поля.

- **[global]** – глобальные настройки Samba;
- **workgroup = WORKGROUP** – имя рабочей группы WORKGROUP (одинаково на всех машинах локальной сети);
- **server string = %h server (Samba, Ubuntu)** – имя машины, отображаемое в сетевом окружении;
- **netbios name = Ubuntu Share** – имя компьютера в сети;
- **dns proxy = no** – не использовать прокси;
- **log file = /var/log/samba/log.%m** – адрес файла (журнал), куда будут складываться сообщения об ошибках и другая информация;
- **max log size = 1000** – задает максимальный размер log-файла (журнала) в кб;
- **passdb backend = tdbsam** – параметр определяет механизм для хранения информации о пользователях и группах, что позволяет переключаться между механизмами без переустановки Samba. Значение параметра делится на две части: имя механизма и местонахождение (строка, которая означает, что механизм имеет отношение только к указанной зоне) Они разделяются знаком «:».
 - **smbpasswd** — механизм по умолчанию. Необязательным параметром указывается путь к файлу **smbpasswd**;

- **tdbsam** — Механизм, основанный на TDB¹. Необязательным параметром указывается путь к TDB (одинаково на всех по умолчанию файл **passdb.tdb** ищется в каталоге **private dir**);
- **ldapsam** — механизм, основанный на LDAP²; обязательным параметром указывается ссылка на LDAP;
- **unix password sync = yes** – синхронизация паролей пользователей **samba** с локальными паролями **Unix** при изменении зашифрованного пароля **SMB** в файле **smbpasswd**; при включении этого параметра от пользователя **ROOT** вызывается программа, определенная в параметре **passwd program**, что позволяет установить новый пароль **UNIX** без доступа к старому паролю **UNIX** (т.к. у программы смены пароля **SMB** нет никакого доступа к старому паролю **cleartext**, есть только новый пароль); по умолчанию: **unix password sync = no**;
- **passwd program = /usr/bin/passwd %u** – имя программы, которую можно использовать для смены паролей **UNIX**; любые вхождения **%u** будут заменены именем пользователя; перед вызовом этой **%u** будут заменены именем пользователя; перед вызовом этой программы имя пользователя проверяется на валидность;
- **pam password change = yes** – параметр позволяет использовать флаг контроля изменяемости паролей **pam**

¹В этом случае **Samba** использует для хранения данных очень простую и не требовательную к ресурсам базу данных, называемую **Trivial Database (TDB)**; TDB организована по принципу «ключ — значение», подробнее тут <https://habr.com/ru/post/91188>.

²LDAP, или **Lightweight Directory Access Protocol**, является открытым протоколом, используемым для хранения и получения данных из каталога с иерархической структурой; обычно используемый для хранения информации об организации, ее активах и пользователях, LDAP является гибким решением для определения любого типа сущностей и их свойств, подробнее тут <https://habr.com/ru/post/538662/>.

для **samba**; если включено, тогда для смены паролей будет использован `ram`¹, вместо программы указанной в параметре **passwd program**;

- **map to guest = bad user** – указывает, когда пользователю будет предоставляться гостевой доступ. Доступно три значения: **never** – никогда, **bad user** – когда такого пользователя не существует, **bad password** – когда пароль введен неверно;
- **usershare allow guests = yes** – этот параметр позволяет неопознанным пользователям получить доступ к общим ресурсам пользователей. Это эквивалентно параметру **guest ok = yes** в настройках общего ресурса.

Общий каталог, доступный в локальной сети

Для начала рассмотрим некоторые полезные параметры для настройки каталога:

1. **path** – путь к общему каталогу;
2. **browsable** – будет ли каталог отображаться на сервере;
3. **writable** – будет ли каталог доступен для записи;
4. **read only** – каталог доступен только для чтения;
5. **guest ok** – будет ли разрешен гостевой доступ;
6. **public** – доступ на чтение всем авторизованным пользователям, включая и гостей, если они определены)
7. **only guest** – если установлено **yes**, то каталог доступен только гостям;

¹модуль, который проверяет подлинность пользователей локального домена

8. **hosts allow** – ip адреса, с которых можно получить доступ к этому серверу (в этом параметре в качестве разделителя может использоваться запятая, пробел, или табуляция как разделитель для имен компьютеров, которым разрешен доступ к сервису. Если параметр определен в разделе **[global]**, то будет действовать для всех общих ресурсов, независимо от индивидуальных настроек каждого общего ресурса, компьютеры можно указывать по имени или ip адресу);
9. **valid users** – по умолчанию могут авторизоваться все пользователи, если в этом параметре передать все список пользователей, то авторизоваться смогут только они (список пользователей и @групп для входа в общий каталог, например, **valid users = vi alena nataly @manager**)
10. **write list** – список пользователей и @групп с правом записи (например, **write list = vi alena nataly**):
11. **create mask** – маска прав для создаваемых файлов.

Подробности можно почитать тут <https://wiki.lblss.ru/smb.conf>, <https://smb-conf.ru>.

Алгоритм создания общего каталога, доступного всем анонимно, без авторизации samba

Шаг 1. Открываем доступ к каталогу.

Для теста его можно создать (**mkdir -p /home/aer/smb/shared**) или использовать уже существующий.

Шаг 2. Установим общему каталогу правильные права доступа.

Следующие команды разрешают доступ к папке для всех и делают владельцем nobody:

```
$ cd /home/aer
$ sudo chmod -R 0755 smb
$ sudo chown -R nobody:nogroup smb
```

Шаг 3. Опишем каталог в файле настройки samba (/etc/samba/smb.conf):

```
[shared]
path = /home/aer/smb/shared
browsable = yes
writable = yes
guest ok = yes
read only = no
```

Шаг 4. Теперь ваш конфигурационный файл должен выглядеть вот так:

```
[global]
workgroup = WORKGROUP
server string = %h server (Samba, Ubuntu)
netbios name = DeskTop Share Library
dns proxy = no
log file = /var/log/samba/log.%m
max log size = 1000
passdb backend = tdbsam
unix password sync = yes
passwd program = /usr/bin/passwd %u
pam password change = yes
map to guest = bad user
usershare allow guests = yes
#=====
[shared]
path = /home/aer/smb/shared
browsable = yes
writable = yes
guest ok = yes
read only = no
```

Шаг 5. После всех внесенных изменений перезапускаем сервер Samba:

```
$ sudo systemctl restart smbd
```

Настройка Samba для анонимного доступа завершена. Теперь можно проверить доступность общего каталога, находясь на любом устройстве в текущей локальной сети (независимо от операционной системы).

Защищенный общий доступ к каталогу Linux

Чтобы открыть доступ к каталогу Samba, к которому будут иметь доступ только пользователи из определенной группы, необходимо описать отдельный каталог в файле настройки Samba в Linux. Пусть это будет каталог `/home/aer/smb/secured` (указывать следует именно `/home/<имя пользователя>`, а не `~`).

Шаг 1. Создать каталог (если используется существующий каталог, пропускаем этот шаг)

```
$ sudo mkdir -p /smb/secured
```

Шаг 2. Создание группы и пользователей

Создание группы:

```
$ sudo addgroup securedgroup
```

Создаём первого пользователя

```
$ sudo adduser user1
```

Создаём второго пользователя

```
$ sudo adduser user2
```

Помещаем пользователя `user1` в группу `securedgroup`

```
$ sudo usermod -a -G securedgroup user1
```

Помещаем пользователя `user2` в группу `securedgroup`

```
$ sudo usermod -a -G securedgroup user2
```

Помещаем системного пользователя `aer` в группу `securedgroup`

```
$ sudo usermod -a -G securedgroup aer
```

Шаг 3. Добавление пользователей в базу Samba

Чтобы пользователь мог работать с общими каталогами в Linux, он должен быть в базе данных Samba.

Создадим для пользователя `aer` пароль Samba:

```
$ sudo smbpasswd -a aer
```

Создадим для пользователя `user1` пароль Samba:

```
$ sudo smbpasswd -a user1
```

Создадим для пользователя `user2` пароль Samba:

```
$ sudo smbpasswd -a user2
```

Шаг 4. Настройка прав

Делаем пользователей владельцами общей папки

```
$ sudo chown -R aer:securedgroup /home/aer/smb/secured  
$ sudo chown -R user1:securedgroup /home/aer/smb/secured
```

Устанавливаем права доступа к папке

```
$ sudo chmod -R 0777 /home/aer/smb/secured
```


Шаг 5. Настройка конфигурационного файла

Последний шаг, добавление настроек в конфигурационный файл Samba:

```
$ sudo nano /etc/samba/smb.conf
```

```
[secured]
path = home/aer/smb/secured
valid users = @securedgroup
guest ok = no
writable = yes
browsable = yes
```

Перезапуск сервер Samba. Теперь доступ к общей папке в Linux могут получить только пользователи группы `securedgroup`.

Настройка общего сетевого каталога, доступного одному пользователю

Настройка происходит также, как и для группы, но при необходимости открыть доступ только конкретному пользователю, меняем **`valid users = @securedgroup`** на **`valid users = user1`**.

Общий каталог Linux, доступ из Windows

Алгоритм подключения к общему каталогу из Windows

1. Узнать имя компьютера в локальной сети либо IP-адрес. Имя компьютера в локальной сети вводится при установке Линукс. Его можно посмотреть в файле `/etc/hosts`, там же его можно и изменить. IP-адрес можно увидеть с помощью команды **`ifconfig`**. Если сетевых соединений несколько, выбираем то с помощью которого мы подключены к локальной сети.

2. В проводнике Windows ввести имя компьютера в локальной сети и IP-адрес

\\192.168.1.67

\\aer-Lenovo-G500

Чтобы подключаться к общему диску **Linux** из **Windows** с помощью своей локальной учётной записи, нужно задать для своей учётной записи **Samba** пароль:

```
sudo smbpasswd -a $USER
```

После этого можно из **Windows** входить в общую папку **Linux** со своим логином и паролем.

Глава 13

Лабораторная работа №9. Локальная сеть. Общий доступ к каталогу. Работа с программой Samba

Задание 0

В конфигурационном файле настройте секцию **[global]**:

1. Имя компьютера в сети.
2. Имя рабочей группы.
3. Имя машины.
4. DNS проху.
5. Адрес журнала и его размер.
6. Механизм хранения информации о пользователях и группах.
7. Синхронизация паролей пользователей samba с локальными паролями.
8. Условия предоставления гостевого доступа (когда пользователя не существует).

9. Позволять ли неопознанным пользователям получить доступ к общим ресурсам пользователей.

Задание 1

На одном из компьютеров в локальной сети (**У1**) создайте каталог **common** с файлом **hello.txt**. В файле напишите «Привет». Укажите следующие настройки **common**:

1. Каталог виден на сервере.
2. Доступ открыт всем.
3. Запись разрешена.

Находясь на другом компьютере этой же сети (**У2**) попытайтесь заменить запись «Привет» на «Hello» в файле **hello.txt** каталога **common**.

Задание 2

На одном из компьютеров в локальной сети (**У1**) в каталоге **common** создайте подкаталог **share** с файлом **fmkn.txt**. В файле напишите номер вашей группы. Укажите следующие настройки **share**:

1. Каталог виден на сервере.
2. Доступ открыт только для гостей.
3. Запись запрещена.

На другом компьютере этой же сети (**У2**) попытайтесь зайти в каталог **share** и в файле **fmkn.txt** вместо номера группы написать свой возраст.

Задание 3

На одном из компьютеров в локальной сети (**Y1**) в каталоге **common** создайте подкаталог **safe** с файлом **name.txt**. Впишите в файл своё имя и укажите следующие настройки **safe**:

1. Каталог виден на сервере.
2. Доступ на чтение всем, включая гостей.
3. Запись разрешена **user1** и **user2** (см. теорию).

На другом компьютере этой же сети (**Y2**) попытайтесь анонимно зайти в каталог **safe** и заменить вписанное в файле **name.txt** имя на другое. Затем сделайте то же самое, но зайдя под **user1**.

Задание 4

На одном из компьютеров в локальной сети (**Y1**) в каталоге **common** создайте подкаталог **team** с файлом **school.txt**. Впишите в файл номер школы, которую вы окончили. Укажите следующие настройки **team**:

1. Каталог виден на сервере.
2. Доступ открыт только участникам группы **class** (включите в неё себя и **user1**).
3. Запись разрешена только **user1**.

На другом компьютере этой же сети (**Y2**) попытайтесь зайти в каталог **team** под именем **user2** и заменить вписанное в файле **school.txt** имя на другое. Затем сделайте то же самое, но зайдя под своим пользователем и **user1**.

Задание 5

На одном из компьютеров в локальной сети (**Y1**) в каталоге **common** создайте подкаталог **foruser1** с файлом **secret.txt**. Впишите в любой текст. Укажите следующие настройки **foruser1**:

1. Каталог виден на сервере.
2. Доступ открыт только вам и **user1**.
3. Запись разрешена **user1**.

На другом компьютере этой же сети (У2) попытайтесь зайти в каталог **team** под именем **user2** и заменить вписанное в файле **school.txt** имя на другое. Затем сделайте то же самое, но зайдя под своим пользователем и **user1**.

Глава 14

Клиент и сервер SSH

Про SSH в Linux

Довольно часто может понадобиться получить доступ к удаленному компьютеру или серверу через интернет. В случае с персональным компьютером, это может понадобиться для срочного решения какой-либо проблемы, а в случае с сервером это вообще очень распространенная практика. В **Linux** наиболее часто для решения таких задач используется протокол **ssh**.

Служба **ssh** позволяет получить доступ к терминалу удаленного компьютера и выполнить там все необходимые вам команды. При своей простоте она достаточно безопасна, чтобы использоваться для решения серьезных коммерческих задач, так и задач обычных пользователей.

SSH (Secure Shell) – это протокол безопасного доступа одного компьютера к другому по сети. Вы можете создавать защищенные соединения между компьютерами, открывать командную строку на удаленном компьютере, запускать графические программы, передавать файлы и организовывать частные сети.

За поддержку протокола **SSH** в **Linux** отвечает набор программного обеспечения **OpenSSH**. Это открытая реализация этого протокола, которая предоставляет все необходимые возможности. В состав пакета **OpenSSH** входят утилиты для

установки соединения, передачи файлов, а также сам **ssh** сервер.

Установка OpenSSH в Linux

Установить **ssh** на **Linux** очень просто, так как программа считается стандартной и есть в официальных репозиториях. Откройте терминал с помощью сочетания клавиш **Ctrl+Alt+T** и выполните команду:

```
$ sudo apt install openssh-server
```

Можно попробовать подключиться к локальному **ssh** серверу:

```
$ ssh localhost
```

Чтобы получить **ssh** доступ к любому компьютеру из сети, достаточно выполнить команду

```
$ ssh name@host
```

где **host** – IP-адрес компьютера, к которому мы хотим подключиться (сервер), а **name** – имя пользователя, от имени которого мы входим (этот пользователь должен существовать на сервере).

Как пользоваться **SSH**

В **Linux** для работы с компьютером по **ssh** используется:

- сервер **ssh** на машине, к которой нужно подключиться;
- клиент **ssh**, на машине, из которой подключаются.

Рассмотрим как пользоваться **ssh**, а также ее возможности. Синтаксис команды выглядит следующим образом:

```
$ ssh [опции] имя@хост [команда]
```

Опции команды **SSH**

- `f` – перевести `ssh` в фоновый режим;
- `g` – разрешить удаленным машинам обращаться к локальным портам;
- `l` – имя пользователя в системе;
- `n` – перенаправить стандартный вывод в `/dev/null`;
- `p` – порт `ssh` на удаленной машине;
- `q` – не показывать сообщения об ошибках;
- `v` – режим отладки;
- `x` – отключить перенаправление X11;
- `X` – включить перенаправление X11;
- `C` – включить сжатие;

Это далеко не все опции утилиты. Многие настройки работы `ssh` можно изменять через конфигурационный файл **`/.ssh/config`**.

Настройка SSH в Linux

Все настройки сервера SSH хранятся в конфигурационном файле `sshd_config`, который находится в папке `/etc/ssh`. Перед внесением изменений лучше сделать резервную копию этого файла:

```
$ sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.factory-defaults
```

Дальше можно перейти к настройке (используем редактор `nano`):

```
sudo nano /etc/ssh/sshd_config
```

В конфигурационном файле следует изменить следующие значения:

1. Сменить порт, на котором работает ssh, возможно злоумышленник не знал включен ли у вас этот сервис. В строке Port изменить значение по умолчанию на любое другое число (раскомментируйте строку, если это необходимо). Например, Port 2222.
2. Отключить возможность входа от имени суперпользователя, так как это может быть небезопасно. Раскомментируйте строку PermitRootLogin и установите значение no.
3. Разрешить аутентификацию по ключу, а не по паролю. Раскомментируйте строку PubkeyAuthentication и установите значение yes.
4. По умолчанию сервер ssh может работать по двум версиям протокола, для совместимости. Чтобы использовать только протокол версии два раскомментируйте (или добавьте) строчку: Protocol 2

После завершения настройки, необходимо перезапустить службу ssh:

```
$ sudo systemctl restart ssh
```

Если изменён порт, то при подключении в клиенте тоже нужно указать новый порт, так как по умолчанию будет использоваться 22, например:

```
$ ssh -p 2222 localhost
```

К тому же, если на компьютере установлен и настроен брандмауэр, то в нем тоже нужно разрешить доступ к новому порту ssh, для этого выполните:

```
$ sudo ufw allow 2222
```

Даже если служба **ssh** находится на порту по умолчанию, его тоже нужно открыть в брандмауэре, если вы собираетесь подключаться к компьютеру через интернет:

```
$ sudo ufw allow 22
```

Доступ только определенного пользователя к SSH

Можно разрешить доступ к **ssh** только для определенного пользователя или группы. Для этого необходимо добавить строки:

```
AllowUsers User1, User2, User3  
AllowGroups Group1, Group2, Group3
```

Здесь **User1** и **Group1** – пользователь и группа к которым нужно разрешить доступ.

SSH соединение по паролю

SSH нужна следующая информация для подключения:

- IP или имя хоста;
- номер порта;
- имя пользователя;
- пароль пользователя.

SSH может предположить имя пользователя и номер порта. Если порт не указан, то используется порт по умолчанию. Если не указан пользователь, то используется то же имя, что и в системе, из которой происходит подключение.

Алгоритм соединения ssh по паролю

Пусть адрес сервера для подключения 192.168.1.36. (адрес компьютера можно узнать командой **sudo ifconfig**).

ivan – имя пользователя на сервере (компьютер, к которому мы хотим подключиться). alex – имя пользователя на клиенте (компьютер, на котором мы работаем).

Шаг 1. Водим команду в формате *name@host*.

```
$ ssh ivan@192.168.1.36
```

Эту команду можно расшифровать так: «мы хотим войти на компьютер с адресом 192.168.1.36 от имени пользователя ivan».

В результате увидим нечто подобное:

```
The authenticity of host '192.168.1.36 (192.168.1.36)' can't be established.  
ECDSA key fingerprint is SHA256:sIxZeSuiivoEQ00RXAQHxylxuEA8SC5r/YPhL8wfp8s.  
Are you sure you want to continue connecting (yes/no)?
```

При первом подключении хост является незнакомым. Спрашивают, хотим ли мы продолжить. Вводим **yes**:

```
Warning: Permanently added '192.168.1.36' (ECDSA) to the list of known hosts.
```

Хост добавлен в список знакомых хостов. Теперь нас просят ввести пароль. Это пароль пользователя `ivan` на сервере.

```
ivan@192.168.1.36's password:
```

Если пароль введен верно, то в приглашении до знака @ вместо Alex будет написано ivan (рис. 14.1). На картинке приглашение выделено красной рамкой.

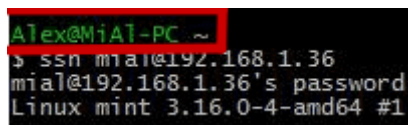


Рис. 14.1:

Примечание:

Если порт отличается от стандартного, то его нужно указать после ключа **-p**. Например, порт **10456** указывается так:

```
$ ssh ivan@192.168.1.36 -p 10456
```

Шаг 2. Проверим текущий каталог и данные устройства и ОС.

Вводим команду

```
$ pwd && uname -a
```

В результате видим текущий каталог (в нашем случае это должен быть каталог `/home/ivan`) и информацию о сервере.

Чтобы закончить сессию (отключиться) введите **exit** или нажмите **Ctrl+D**.

Выполнение команд на сервере без создания сессии. Чтобы выполнить несколько команд на сервере, не обязательно открывать сессию (то есть выполнять **ssh user@host**, после чего появляется информация о последнем соединении). Можно просто ввести команду.

Пример:

Можно ввести команду

```
$ ssh ivan@192.168.1.36 ls
```

После чего вы увидите на экране список подкаталогов домашнего каталога пользователя `ivan`. Эту информацию можно вывести в файл.

```
$ ssh ivan@192.168.1.36 'ls' > dirlist.txt
```

Обратите внимание на одиночные кавычки. Это сделано для того чтобы раскрытие пути было выполнено на сервере, а не на локальной машине. А вот если мы хотим вывод перенаправить в файл на удалённой машине, мы можем поместить оператор редиректа и имя файла внутри одиночных кавычек:

```
$ ssh remote-sys 'ls * > dirlist.txt'
```

Настройка соединений SSH без пароля

Использование **ssh** пароля для входа на сервер не только неудобно, но и небезопасно, потому что этот пароль в любой момент может быть подобран. Самый надежный и часто используемый способ аутентификации – с помощью пары ключей RSA. Секретный ключ хранится на компьютере, а публичный используется на сервере для удостоверения пользователя.

Алгоритм соединения ssh с помощью RSA

Шаг 1. Создадем ключ RSA

```
$ ssh-keygen -t rsa
```

Предлагается указать место для хранения нашего ключа. По умолчанию это будет каталог **.ssh** в вашей домашнем каталоге. Чтобы согласиться с настройками по умолчанию, нажимаем **«Enter»**.

```
Enter file in which to save the key (/home/kristina/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

Нас просят ввести идентификационную фразу, которая нужна для разблокировки закрытого ключа. Это не обязательно, и чтобы оставить ее пустой, нажимаем **«Enter»**. Вы увидите сообщение о том, что ключ был создан.

Теперь открытый (публичный) и закрытый SSH-ключи лежат в каталоге **.ssh** в файлах **id_rsa.pub** и **id_rsa** соответственно.

Шаг 2. Загружаем файл id_rsa.pub на сервер.

На удалённой машине нам нужно создать каталог **.ssh** (если его нет).

!В данном примере команды выполняются от пользователя Alex, но можно по паролю зайти на сервер под именем ivan и выполнять стандартные команды терминала без указания user@host

```
$ ssh ivan@192.168.1.36 mkdir .ssh
```

Скопировать содержимое файла id_rsa.pub на удалённую машину.

```
$ cat .ssh/id_rsa.pub | ssh ivan@192.168.1.36  
'cat >> .ssh/authorized_keys'
```

Шаг 3. Вводим команду в формате name@host

Вводим

```
$ ssh ivan@192.168.1.36
```

и, если все сделано верно, соединение должно быть установлено без запроса пароля.

Решение возможных проблем

Теперь мы можем подключаться по ключу, но если вдруг что-то не получается, то нужно еще выставить права:

```
$ chmod -u=rwX,go= ~/.ssh  
$ chmod -u=rw,go=r ~/.ssh/authorized_keys
```

Если не помогло, смотрим конфигурационный файл (в примере используем редактор nano):

```
$ sudo nano /etc/ssh/sshd_config
```

Нужно проверить, чтобы эти атрибуты имели корректные значения:

```
RSAAuthentication yes  
PubkeyAuthentication yes  
PasswordAuthentication no
```

Дальше нажимаем «**Ctrl+O**»(сохранить) и «**Ctrl+X**»(закрыть файл).

Перезапускаем сервер SSH на удаленном хосте:

```
$ sudo /etc/init.d/ssh reload
```

Теперь мы можем выполнить авторизацию по ssh-ключу со своим удаленным хостом с помощью команды:

```
$ ssh -i /path-to-private-key user@remote-host-ip-address
```

Взять пароль из локального файла

Хранить пароли в обычных текстовых файлах небезопасно, но если очень хочется, то можно:

```
$ ssh user@host < local_file.txt
```

Изменить приветствие SSH

При входе по ssh может выводиться приветствие, изменить его очень легко. За это отвечает файл `/etc/issue`. В нём можно хранить нужный текст:

```
$ nano /etc/issue  
Welcome!
```

Смотрим неудачные попытки входа SSH

Чтобы посмотреть были ли попытки неудачного доступа по ssh к вашему серверу и с каких IP адресов, надо знать, что все запросы записываются в файл `/var/log/secure`. Их можно отфильтровать командой:

```
$ cat /var/log/secure | grep "Failed password for"
```

Передача стандартного вывода с локальной машины на удалённую по SSH

Вариант выполнения команды:

```
$ cat .ssh/id_rsa.pub | ssh user@host  
'cat >> .ssh/authorized_keys'
```

Команда `cat` построчно считывает и отображает содержимое файла **.ssh/id_rsa.pub**, расположенного на локальной машине.

`|` (труба) передаёт то, что должно было бы появиться в стандартном выводе, другой команде.

Вместо команды, которая должна была бы обрабатывать передаваемые ей строки, происходит соединение к удалённой системе (`ssh user@host`).

На удалённую систему приходят строки, для которых предусмотрена команда

```
cat >> .ssh/authorized_keys
```

Команда содержимое стандартного вывода построчно записывает в файл **.ssh/authorized_keys**, находящийся на удалённой машине.

Выполнение локального скрипта

Выполнение интерпретатора `bash` на удаленном сервере и передача ему локального скрипта с помощью перенаправления ввода `bash`:

```
$ ssh name@host 'bash -s' < script.sh
```

Бекап на удаленный сервер и восстановление

Можно сохранять бекэп диска сразу на удаленном сервере с помощью **ssh**. Перенаправление вывода **dd** с помощью оператора перенаправления `|`, затем сохранение его на той стороне в файл:

```
$ sudo dd if=/dev/sda | ssh user@host 'dd of=sda.img'
```

Восстановление состояние диска из сделанной копии:

```
$ ssh user@host 'dd if=sda.img' | dd of=/dev/sda
```

Здесь и выше **/dev/sda** – имя файла вашего жесткого диска. Открытие графической программы, расположенной на удалённом компьютере. Для следующего фокуса нужно два компьютера с системой **Linux**. Причём оба **Linux**'а должны быть с графическим пользовательским интерфейсом.

Туннелирование с SSH

При соединении с сервером через SSH происходит создание зашифрованного туннеля, который образуется между локальной и удалённой системами. Обычно, этот туннель используется для того, чтобы набранные на локальной машине команды безопасно были переданы удалённой машине, а результат также безопасно прислан обратно.

Пожалуй самая интересная возможность – это возможность транслировать трафик систем **X Window**. Проще говоря, запускать графические приложения на сервере, а результат видеть на локальной машине. Есть возможность использовать **ssh** для запуска полноценных **X11** приложений. Чтобы все заработало, необходимо разрешить эту возможность на стороне сервера, добавьте такую строчку:

X11Forwarding yes

Нужно перезагрузить ssh сервер, чтобы сохранить изменения:

```
$ service sshd restart
```

После настройки конфигурационного файла можно выполнить:

```
$ ssh -X ivan@192.168.1.36
$ gedit
```

Таким образом, **SSH** запускается с ключом **-X**. А затем просто запускается программа. Конечно, подразумевается, что **gedit** установлен на сервере. На некоторых системах для этого «фокуса» нужно использовать опцию **-Y** вместо опции **-X**.

Копирование с/на удалённый компьютер (scp и sftp)

Команда scp

OpenSSH также включает программы, которые используются для копирования файлов по сети. Первая программа, **scp** («безопасное копирование»).

Синтаксис команды:

```
$ scp path/to/local_file name@host:path/to/dir
```

Кроме утилиты **scp**, передача файлов **ssh** может быть выполнена более хитрым способом:

Прочитаем файл и с помощью **cat**, передадим, а там сохраним поток в файл.

Команда:

```
$ cat local_file | ssh name@host "cat > remote_file"
```

Или так:

```
$ ssh name@host "cat > remote_file" < local_file
```

Можно сжимать файлы перед передачей с помощью **tar**, а потом их сразу же на лету распаковывать:

```
$tar czf - /home/user/file | ssh name@host tar -xvz
```

Такое копирование файлов **ssh** позволяет отправлять сразу целые каталоги.

Пример: давайте скопируем документ с названием **document.txt**.

Для копирования файла с удаленной машины на локальную:

Синтаксис команды:

```
scp name@host:file
```

Удалим файл на локальной машине, если он есть

```
rm dirlist.txt
```

Создадим файл на удалённой машине

```
ssh ivan@192.168.1.36 'ls * > dirlist.txt'
```

Проверим его наличие

```
ssh ivan@192.168.1.36 'ls -l'
```

Скопируем его на локальную машину

```
scp ivan@192.168.1.36:dirlist.txt
```

Проверим его содержимое

```
cat dirlist.txt
```

Для копирования файла с локальной машины на удалённую:

Синтаксис команды:

```
scp file name@host:file:.
```

Создаём новый файл

```
touch nfile.txt
```

Отправляем файл

```
scp nfile.txt ivan@192.168.1.36: .
```

Проверяем наличие файла на удалённой машине

```
ssh ivan@192.168.1.36 'ls -l'
```

В команде отправки:

nfile.txt – имя файла,

ivan@192.168.1.36 – имя пользователя и удалённый хост,

. (точка) означает, что файл нужно скопировать в текущий рабочий каталог на удалённом сервере, при этом имя файла останется прежним, т.е. **nfile.txt**

Памятка:

Для копирования файла с В на А когда залогинены в В:

```
scp /path/to/file username@a:/path/to/destination
```

Копирование файла с В на А когда залогинены в А:

```
scp username@b:/path/to/file /path/to/destination
```

Команда sftp

Вторая программа для копирования файлов через SSH – это **sftp**. Как следует из её имени, она является безопасным заменителем **ftp** программ. **sftp** работает как и оригинальная **ftp** программа. Тем не менее, вместо отправки чистым текстом она использует зашифрованный туннель SSH. Важным преимуществом **sftp** перед **ftp** является то, что для неё не требуется запущенный FTP сервер на удалённом хосте. Для неё требуется только SSH сервер. Это означает, что любая удалённая машина, которая подключена через SSH клиент может также быть использована как FTP-подобный сервер. Вот пример сессии:

```
$ sftp ivan@192.168.1.36
Connected to 192.168.1.36.
sftp> ls
dirlist.txt newfile.txt
nfile.txt temp
Видео Документы
Загрузки Изображения
Музыка Общедоступные
Рабочий стол Шаблоны
sftp> ll
dirlist.txt  nfile.txt
sftp> ls temp
temp/TakeMeHome
sftp> cd temp/
sftp> get TakeMeHome
Fetching /home/mial/temp/TakeMeHome to TakeMeHome
sftp> bye
```

SFTP протокол поддерживается многими графическими файловыми менеджерами, которые можно найти в дистрибутивах Linux. Используя как Nautilus (GNOME), так и Konqueror (KDE), мы можем вводить URI (ссылки) начинающиеся на `sftp://` в строку перехода и работать с файлами, расположенными на удалённой системе с запущенным SSH сервером.

Как сравнить файлы на удалённом компьютере (выполнить команду diff) через SSH

`diff` — это утилита командной строки, которая сравнивает содержимое двух файлов (или каталогов) и построчно показывает, чем они различаются. Очень просто пользоваться `diff`, когда оба файла размещены на локальном хосте. А как быть если один или оба входных файла размещены на удалённом хосте?

Сравнение локального файла и файла на сервере:

```
$ ssh name@host "cat remote_file.txt" | diff - local_file.txt
```

Вышеприведённая команда запросит у вас **SSH** пароль для удалённого хоста. После корректного ввода пароля, команда прочитает содержимое удалённого файла через **SSH**, а затем передаст его по трубе в утилиту **diff** для сравнения с локальным файлом.

Другой метод для сравнения содержимого файла по **SSH** – это явное использование перенаправления ввода. Чтобы использовать этот метод, вы должны подключиться к серверу по **ssh** (открыть сессию). Подразумевая что это готово, вы можете сравнить локальный и удаленный файл.

```
$ diff local_file.txt <(ssh name@host  
'cat remote_file.txt')
```

Сравнение двух удалённых файлов:

Здесь так же нужно сначала открыть сессию, а затем выполнить команду:

```
$ diff <(ssh user@remote_host 'cat remote_file.txt')  
<(ssh user2@remote_host2 'cat remote_file2.txt')
```

Запомните: вышеприведённое перенаправление ввода не поддерживается оболочкой **sh**. Используйте оболочку **bash** или что-нибудь совместимое.

Как редактировать файл на другом компьютере через SSH

Когда нужно отредактировать файл, размещённый на удалённом хосте, было бы здорово, если бы можно было отредактировать его так, будто это локальный файл. Конечно, можно поднять через **ssh** туннель, но это излишне.

Один из плагинов **vim** – **netrw**, задействует чтение и запись файлов по Сети. **netrw** поддерживает различные протоколы уровня приложений, включая **ftp**, **http**, **scp**, **rsync** и **rscp**.

Если вы хотите отредактировать удалённый файл через **ssh** с использованием **vim**, вы можете сделать так.

```
$ sudo apt-get install vim
$ vim scp://user@remote_host//home/user/path/file
```

Обратите внимание на двойной слеш **"/"** для каталога **root** на удалённом хосте. Вышеприведённая команда **vim** откроет файл, размещённый на удалённом хосте для редактирования. За сценой, **vim** использует **scp** для получения запрошенного файла с удалённого хоста, размещает файл в **/tmp** и, наконец, открывает его для редактирования.

Когда вы попытаетесь сохранить изменения в файле, изменения, в первую очередь, будут применены к локальной копии в **/tmp**, которая затем выгрузится через **scp** на удалённый хост.

Завершение сессии SSH

Когда соединение с удалённым сервером разорвано невозможно ввести никакую команду и сочетания клавиш **Ctrl+C**, **Ctrl+Z**, **Ctrl+D** не работают. И не будут работать поскольку клиент пытается отправить эти команды на сервер. Но есть решение – **Escape**-последовательности. Чтобы активировать их поддержку добавьте строку:

```
EscapeChar ~
```

в файл **/etc/ssh/ssh_config**. Теперь, чтобы разорвать **ssh** соединение достаточно нажать «**Enter**» и набрать **~**. Другие управляющие символы можно узнать нажав **~?**.

Каталог **~/ .ssh**

Помимо того, что утилита **ssh-keygen** генерирует публичные и закрытые ключи, она также создает скрытый каталог **.ssh** с необходимыми правами доступа. Вы можете создать каталог **.ssh** вручную, но в этом случае вам придется выполнить команду **chmod 700** для установки необходимых прав доступа к ее содержимому. В противном случае клиент **ssh** откажется от использования ключей.

Отладка соединений по протоколу SSH

Используйте команду **ssh -v** для получения отладочной информации о попытке соединения по протоколу **ssh**.

Сервер sshd

Исполняемый файл сервера **ssh** носит имя **sshd** и содержится в пакете программного обеспечения **openssh-server**.

```
sudo dpkg -l openssh-server | tail -1
ii  openssh-server  1:5.9p1-5ubuntu1  secure shell (SSH) server,...
```

Ключи sshd

Файлы публичных ключей, используемые **sshd**, располагаются в каталоге **/etc/ssh** и могут читаться всеми пользователями. Файлы закрытых ключей могут читаться исключительно пользователем **root**.

```
sudo ls -l /etc/ssh/ssh_host_*
-rw----- 1 root root 668 июн 7 2011 /etc/ssh/ssh_host_dsa_key
-rw-r--r-- 1 root root 598 июн 7 2011 /etc/ssh/ssh_host_dsa_key.pub
-rw----- 1 root root 1679 июн 7 2011 /etc/ssh/ssh_host_rsa_key
```

Демон ssh-agent

При генерации ключей с помощью утилиты **ssh-keygen** у вас имеется возможность ввести ключевую фразу для ограничения доступа к этим ключам. Чтобы не вводить эту фразу каждый раз при использовании закрытых ключей, вы можете

добавить закрытый ключ в базу данных демона **ssh-agent** с помощью утилиты **ssh-add**.

В большинстве дистрибутивов Linux демон **ssh-agent** запускается автоматически после входа пользователя в систему. Проверить, выполняется ли демон **ssh-agent** можно командой:

```
sudo ps -ef | grep ssh-agent
Alex      2405  2365  0 08:13 ?
00:00:00 /usr/bin/ssh-agent...
```

Выводим список открытых ключей, которые соответствуют закрытым ключам, добавленным в базу данных демона **ssh-agent**.

```
$ ssh-add -L
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAvgI+Vx5UrIsusZPl8da8URHGsxG7yivv...
wMGqa48Kelwom8TGb4Sgcwpp/V0/ldA5m+BGCw== Alex@deb503
```

Более подробная информация есть на страницах следующих руководств:

- `man ssh-keygen`
- `man ssh-agent`
- `man ssh-add`

Глава 15

Лабораторная работа №10. Протокол SSH

Для выполнения лабораторной работы понадобится два компьютера под управлением **Linux** (с графическим пользовательским интерфейсом). Один из них мы будем называть сервером (удаленной машиной), а другой – клиентом (локальной машиной).

Задание 0

Установите **SSH** на оба компьютера.

Задание 1

На сервере:

- Создайте копию конфигурационного файла.
- В конфигурационном файле отключите возможность входа от имени суперпользователя и установите использование протокола версии 2 (после изменения конфигурационного файла не забудьте перезагрузить **ssh**).
- Создайте пользователя **user1** и разрешите доступ по **ssh** только ему.
- Узнайте **IP** сервера.

Задание 2

На клиенте:

Установите SSH соединение по паролю

- Сначала от имени пользователя клиента (того пользователя, от имени которого вы работаете на локальной машине).
- Затем от имени пользователя `user1`, созданного на сервере.
- После удачного соединения проверьте текущий каталог и информацию об устройстве.

Задание 3

На клиенте:

Создайте ключ **RSA** с идентификационной фразой и соединитесь с сервером по этому ключу (без пароля)

!!Не забудьте завершить сессию **ssh** (**Ctrl+D** или **exit**).
RSA создается от имени пользователя локальной машины.

Задание 4

На клиенте:

Нижеприведенный текст поместите в файл **script.sh** на локальной машине и выполните удаленно на сервере, не открывая сессию.

```
#!/bin/bash
echo "это список каталогов твоей домашней директории" > server.txt
ls -a | cat » server.txt
```

Задание 5

На сервере:

Установите приложение **geany** (если оно не установлено).

На клиенте:

1. Запустите приложение **geany** с сервера (результат вы должны видеть на экране клиента).
2. Создайте файл с любым текстом и сохраните в домашнем каталоге **user1**.
3. Проверьте, сохранен ли файл.

Задание 6

На клиенте:

1. В домашнем каталоге **user1** удаленно создайте файл **hello.txt** с текстом "Привет".
2. Закройте сессию. Скопируйте файл **hello.txt** на локальную машину.
3. Проверьте содержимое файла **hello.txt**.
4. Исправьте "Привет" на "Hello".
5. В домашнем каталоге **user1** создайте подкаталог **Changed_Hello**.
6. Отправьте измененный файл **hello.txt** на сервер в этот каталог.
7. Удаленно сравните два файла на сервере с помощью **diff**.

Задание 7

На клиенте:

Создайте каталог **Arh** на локальной машине и поместите туда 3 пустых файла: **file1.txt**, **file2.txt**, **file3.txt**.

Одной командой заархивируйте каталог, отправьте на сервер и разархивируйте.

Задание 8

На сервере:

- Проверьте права доступа у файлов публичных и закрытых ключей в **/etc/ssh**. Файлы открытых ключей должны быть доступны для чтения всем, а закрытых – только суперпользователю.
- Проверьте, исполняется ли демон **ssh-agent**.
- Просмотрите список открытых ключей, которые соответствуют закрытым ключам, добавленным в базу данных демона **ssh-agent**.

Глава 16

Настройка и использование FTP-сервера в Linux

Что такое FTP

FTP расшифровывается как File Transfer Protocol. Название подразумевает, что этот протокол используется для передачи файлов или директорий с одного хоста на другой по сети - как локальной, так и через интернет. Здесь будет использоваться легкий FTP-сервер **vsFTPd**, предназначенный для работы на высоконагруженных серверах.

Установка FTP-сервера в Linux

В процессе установки создается пользователь **ftp** с домашним каталогом **/srv/ftp**. Сама установка FTP-сервера выполняется одной командой:

```
$ sudo apt-get install vsftpd
```

Нормальный и автономный режимы работы FTP Автономный режим

По умолчанию **vsftpd** запускается в автономном режиме, в котором запускаемая на сервере служба использует собственный стартовый скрипт, называемый демоном. В случае

`vsftpd` это `/etc/init.d/vsftpd`. Данный автономный демон в момент старта службы FTP берет управление на себя. Демон `vsftpd` предоставляет администратору несколько команд для управления FTP-сервером `vsftpd`:

- **start** или **stop** – используется для запуска или остановки ftp-сервера.
- **status** – выводит подробную информацию о текущем состоянии вашего FTP-сервера.
- **restart** – это альтернатива последовательности из остановки и запуска сервера. Если сервер уже остановлен, команда `restart` запустит его.
- **reload** – эта команда позволяет перезагрузить и применить все новые настройки. Ее отличие от `restart` заключается в том, что применение новых настроек производится без остановки сервера.

Для запуска, перезагрузки и применения новых настроек используется утилита `service`:

```
$ sudo service vsftpd start
```

Такой синтаксис используется и при выполнении других команд из списка выше.

Нормальный режим

Шаг 1. Установить xinetd

В нормальном режиме за работу службы отвечает суперсервер `xinetd`, поэтому перед запуском `vsftpd` нужно установить его:

```
$ sudo apt-get install xinetd
```


В случае, если он у вас уже установлен, эта команда не нужна.

Шаг 2. Создать файл vsftpd

В каталоге `/etc/xinetd.d/` создадим файл **vsftpd** со следующим содержимым:

```
service ftp
{
    disable                = no
    socket\_type            = stream
    wait                   = no
    user                    = root
    server                  = /usr/sbin/vsftpd
    per\_source              = 5
    instances               = 200
    no\_access               = 10.1.1.10
    banner\_fail             = /etc/vsftpd.busy
    log\_on\_success          += PID HOST DURATION
    log\_on\_failure          += HOST
}
```

Здесь можно изменять любые опции, чтобы настроить супер-сервер в соответствии со своими требованиями.

Опции, на которые стоит обратить внимание:

- **server** – необходимо ввести в командной строке "\$ which vsftpd чтобы узнать правильный путь.
- `no_access` – все хосты с IP-адресами, указанными в этой директиве, будут заблокированы.
- `banner_fail` - здесь можно указать путь к текстовому файлу, содержимое которого будет показано для любых заблокированных IP-адресов.

Шаг 3. Отредактировать конфигурационный файл FTP-сервера

Сначала скопируем оригинальный файл настроек, чтобы в случае проблем вернуть все в исходное состояние:

```
$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf.orig
```

Отредактируем файл **/etc/vsftpd.conf** в соответствии с нашими задачами. Для этого его нужно открыть в каком-нибудь редакторе. Например, в **nano**.

```
$ sudo nano /etc/vsftpd.conf
```

Раскомментируйте строки с нужными директивами и установите значение, которое вам необходимо. Если строки с какой-то директивой нет, впишите её вручную.

- **listen=YES/NO** – открывать порты или не открывать, полностью перепоручив их суперсерверу **xinetd**;
- **anonymous_enable=YES/NO** – разрешить/запретить анонимное соединение;
- **no_anon_password=YES/NO** – не запрашивать(YES)/запрашивать(NO) пароль при анонимном соединении.
- **local_enable=YES/NO** – разрешить/запретить авторизацию тем пользователям, которые прописаны в **/etc/passwd**;
- **userlist_file=/etc/vsftpd.userlist** – расположение файла с именами тех пользователей, которым запрещено использовать соединение по **ftp**;
- **userlist_enable=YES** – использовать/не использовать этот файл;

- **userlist_deny=YES/NO** – при этом вы открываете доступ только пользователям, перечисленным в **/etc/vsftpd.userlist**.

Для начала установим следующие настройки: не открывать порты, разрешить подключение анонимно, не спрашивать при этом пароль, разрешить подключение авторизованным пользователям. Это соответствует следующим полям в конфигурационном файле:

```
listen=NO  
local_enable=YES
```

Внимание! После любых изменений в конфигурационном файле, необходимо перезапустить **vsftpd** следующей командой:

```
sudo service vsftpd restart
```

Шаг 4. Убедиться, что демон vsftpd отключен.

```
$ sudo service vsftpd stop
```

Шаг 5. Запустить FTP-сервер в нормальном режиме.

```
$ sudo service xinetd restart
```

Шаг 6. Протестировать и открыть порт 21.

Чтобы убедиться в нормальной работе FTP-сервера, необходимо протестировать и открыть порт 21 с помощью команды **netstat**:

```
$ netstat -ant | grep 21  
tcp    0    0 0.0.0.0:21    0.0.0.0:*    LISTEN
```

Настройка пользовательского доступа

Анонимный доступ по FTP

Независимо от того, запустили вы FTP-сервер в автономном режиме или в нормальном, вы можете создать первое локальное **ftp**-подключение. Для начала установим тестовое подключение, которое будет анонимным.

Шаг 1. Дать разрешение на создание анонимного подключения.

Проверьте, чтобы в файле `/etc/vsftpd.conf` была раскомментирована строка **`anonymous_enable`** и имела значение «**YES**». Можно сразу же выключить запрос пароля, установив **YES** в строке **`no_anon_password`**.

```
anonymous_enable=YES  
no_anon_password=YES
```

Шаг 2. Создать анонимное подключение

В терминале введите команду

```
$ ftp localhost
```

После соединения с **localhost** будет запрошено имя пользователя, от имени которого осуществляется вход. В нашем случае вход анонимный, поэтому вводим «**anonymous**».

Результат примерно следующий:

```
Connected to localhost.  
220 (vsFTPd 2.3.5)  
Name (localhost:root): anonymous  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>
```

Успешное подключение подтверждает, что FTP-сервер в локальной системе настроен правильно.

Доступ локальных пользователей по FTP

Чтобы разрешить авторизацию всем пользователям, указанным в файле `/etc/passwd`, необходимо изменить значение директивы `local_enable` в файле `/etc/vsftpd.conf`.

```
local_enable=YES
```

После этого любой пользователь, указанный в файле `/etc/passwd`, сможет авторизоваться, используя свой пароль.

Список доступа пользователей

Шаг 1. Создать список доступа

Список доступа – файл, который будет использоваться сервером `vsftpd`. Обычно он находится в директории `/etc/` и имеет произвольное имя. Например, `vsftpd.userlist`. В него запишем список пользователей, которым будет разрешён или запрещён доступ. Каждое имя пользователя в этом файле должно располагаться на отдельной строке.

Шаг 2. Определить список доступа в конфигурационном файле vsftpd

Далее, необходимо определить этот новый список пользователей в конфигурационном файле `/etc/vsftpd.conf` и активировать директиву `userlist_enable`:

```
userlist_file=/etc/vsftpd.userlist  
userlist_enable=YES
```

Таким образом всем пользователям, перечисленным в файле `/etc/vsftpd.userlist`, будет отказано в доступе к FTP-серверу.

Шаг 2.1 (дополнительно) разрешить соединение пользователям из списка

Для этого установите в строке `userlist_deny` значение "NO"

`userlist_deny=NO`

При этом вы открываете доступ только тем пользователям, которые перечислены в `/etc/vsftpd.userlist`.

Другие опции конфигурации

Сервер `vsFTPD` имеет множество опций настроек, которые позволяют тонко настроить его в соответствии со своими нуждами. Ниже приведен список наиболее важных опций:

Максимальное количество пользователей, одновременно использующих FTP-сервер. 0 означает неограниченное количество пользователей.

`max_clients=0`

Максимальное количество пользователей с одного IP-адреса (2).

`max_per_ip=2`

Отклонять запросы на скачивание

`download_enable=NO`

Разрешить запись авторизованным пользователям:

`write_enable = YES`

Установить значение `umask` для новых файлов, создаваемых по FTP:

`local_umask = 022`

Включить сообщение о необходимости выбрать каталог после регистрации:

`dirmessage_enable = YES`

Использовать порт 20 для передачи данных вместо случайного, это нужно для нормальной работы фаервола:

```
connect_from_port_20 = YES
```

Использовать PAM сервис vsftpd:

```
pam_service_name=vsftpd
```

При входе пользователей на FTP-сервер, они могут работать только в корневом каталоге FTP. Если пользователи будут ограничены только своим домашним каталогом, то необходимо раскомментировать эти строки:

```
chroot_local_user = YES  
allow_writeable_chroot = YES
```

Первая строка указывает, что нужно разместить пользователя в изолированном домашнем каталоге, а вторая сообщает, что ему можно разрешить запись в этот каталог.

После того как настройка завершена, необходимо сохранить изменения в конфигурационном файле и перезапустить vsftpd:

```
$ sudo systemctl restart vsftpd
```

Настройка защищенного соединения

Установка FTP завершена, но в наше время небезопасно использовать открытую передачу данных через сеть. Поэтому рассмотрим, как настроить защищенное соединение для vsftpd. Необходимо создать каталог и сгенерировать сертификат, который будем использовать:

```
sudo mkdir /etc/ssl/private  
sudo openssl req -x509 -nodes -days 720 -newkey rsa:2048 -keyout  
/etc/ssl/private/vsftpd.key -out /etc/ssl/private/vsftpd.pem
```

Затем нужно разрешить доступ к портам защищенного соединения FTP с фаерволе UFW:

```
sudo ufw allow 990/tcp
sudo ufw allow 40000:50000/tcp
sudo ufw status
```

И осталось внести несколько правок в настройку самого FTP сервера. Включаем **ssl_enable** и отключаем поддержку шифрования **sslv2** и **sslv3**, оставляем только **tlsv1**:

```
ssl_enable=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
```

Далее раскомментируйте такие строки:

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.key
```

Нужно запретить вход анонимных пользователей по SSL:

```
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
```

Необходимо установить самый сложный шифр SSL:

```
ssl_ciphers=HIGH
```

И настроить диапазон портов для передачи данных:

```
pasv_min_port=40000
pasv_max_port=50000
```

Осталось перезагрузить сервис:


```
$ sudo systemctl restart vsftpd
```

Теперь тестируем что у нас получилось. Как видите, мы уже не можем авторизоваться с помощью утилиты **ftp**. FTP сервер выдает, что мы обязаны использовать шифрование. Поэтому для тестирования применим **FileZilla**.

Теперь все данные будут передаваться по зашифрованному соединению. Установка ftp-сервера Linux завершена.

Передача файлов по FTP в Filezilla

Установка Filezilla

```
$ sudo apt install filezilla
```

Вот так выглядит главное окно утилиты (рис. 16.1). Для подключения к серверу необходимо ввести данные аутентификации в поля, которые находятся в верхней части окна (см. рис. 16.1):

Затем нажмите кнопку "**Быстрое соединение**". Если программа удачно подключится к серверу, то вы увидите список файлов на правой панели. На левой панели находится ваша локальная файловая система. Чтобы передать файлы, перетасщите их в нужную часть окна. В контекстном меню вы можете выполнять различные действия с файлами, удалять, перемещать, копировать, переименовывать и даже редактировать. К тому же программа поддерживает подключение к серверам FTP, которые используют шифрование.

Команды FTP

Синтаксис команд следующий:

```
ftp [-v] [-d] [-i] [-n] [-U] [-p] [-g] [host]
```

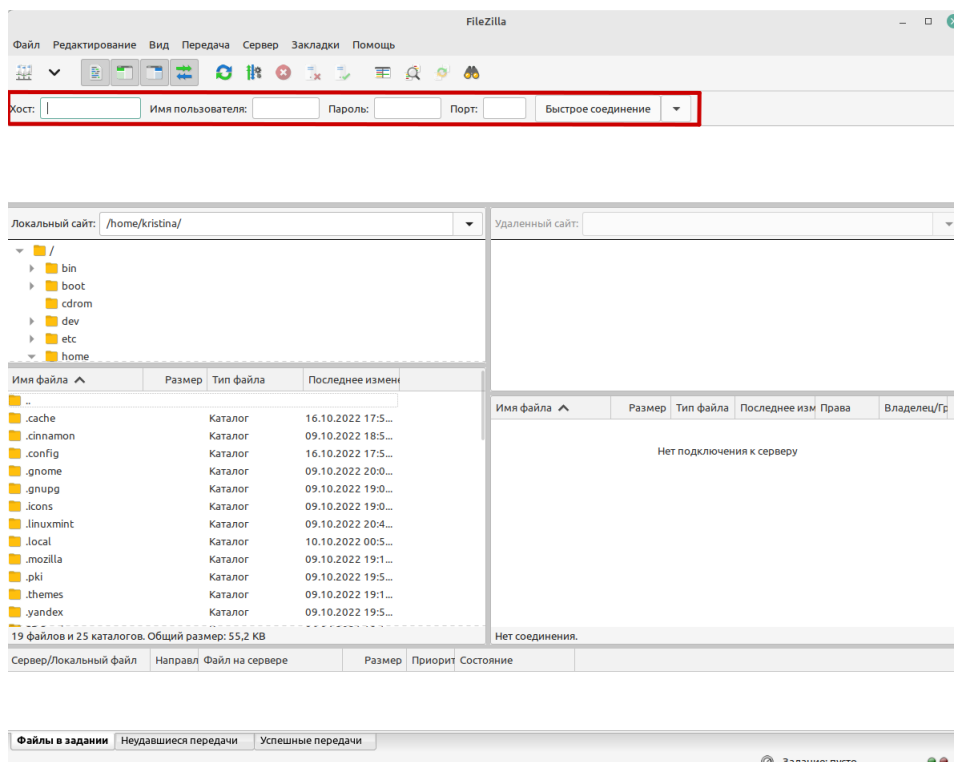


Рис. 16.1: Внешний вид графической утилиты Filezilla

С помощью **ftp** можно создавать связи с несколькими машинами. Связь, которая используется в данный момент, называется текущей.

Виды записи ftp

Программа **ftp** позволяет использовать два вида записи:

- ASCII для текстовых файлов (по умолчанию).
- Двоичный для, например, программ, картинок, архивов.

Вызов ftp

Чтобы вызвать **ftp** из оболочки LINUX, введите команду

```
$ ftp
```

На вашем экране появится приглашение. Оно выглядит следующим образом:

ftp>

Вы можете указать имя машины, с которой хотите связаться, хотя это необязательно. Следующий пример показывает как записывается имя машины **ftp.botik.ru**:

```
$ ftp ftp.botik.ru
```

Вы можете также вызвать ftp без имени машины, например:

```
$ ftp
```

Опции ftp

Дополнительно при вызове **ftp**, вы можете задать некоторые опции. Они помещаются после названия команды, но до имени машины (если оно вообще указывается). Каждая опция состоит из дефиса (-) и одной буквы, например: **-v**.

Значения опций:

- **-v** заставляет **ftp** работать в многословном режиме; этот режим устанавливается по умолчанию, если **ftp** выполняется интерактивно; если **ftp** выполняется в командном режиме, многословный режим выключен.
- **-d** заставляет **ftp** работать в режиме отладки; в этом режиме сообщения **ftp**, посланные **ftp** в удаленную машину, отображаются на экране вашего дисплея; если вы не используете эту опцию, информация не отображается; вы можете также вызвать этот режим в **ftp**, используя команду **debug**.
- **-n** предотвращает использование **ftp** авторегистрации, во время связи с удаленной машиной; когда установлен режим авторегистрации, **ftp** идентифицирует вас автоматически на удаленной машине и регистрирует вас на этой машине.

- **-g** является причиной лишения имен файлов **UNIX** их расширений, таких как универсальное (*); если Вы не используете эту опцию, **ftp** расширяет имена файлов с помощью универсального расширения в списке файлов; вместо этой опции можно использовать команду **glob**.

Ниже даны примеры использования ftp опций:

```
$ ftp -v -d ftp.botik.ru
```

Вызывает **ftp** в многословном и отладочном режиме и говорит **ftp** открыть связь с удаленной машиной по имени **ftp.botik.ru**.

```
$ ftp -vd
```

Вызывает **ftp** в многословном и отладочном режиме, но не открывает связи с удаленной машиной.

```
$ ftp -ng ftp.botik.ru
```

Вызывает **ftp** с отменой авторегистрации и универсального расширения, и заставляет открыть связь с удаленной машиной **ftp.botik.ru**.

```
$ ftp -n -d
```

Вышеприведенная команда вызывает **ftp** с отменой авторегистрации и универсального расширения, не открывая связи с какой-либо машиной.

Использование файла .netrc для автоматической регистрации

Вы можете создать файл по имени **.netrc** в вашем домашнем каталоге в качестве дополнительного удобства. Этот файл содержит строки-элементы регистрационных данных для каждой машины, которая вам нужна для автоматической

связи. Когда вы вызываете **ftp**, указывая машину, т.е. когда вы одновременно с вызовом открываете связь с машиной, **ftp** читает файл **.netrc**. Если есть строка-элемент для этой машины, то **ftp** автоматически соединяет вашу машину с этой удаленной машиной. Если вы открыли связь в многословном режиме, то вы увидите как это происходит. Формат файла состоит из отдельных полей, представленных ключевыми полями:

```
machine имя login имя password пароль
```

где **machine**, **login**, **password** это ключевые слова, за которыми следуют символьные данные, необходимые для регистрации:

- **machine** – имя узла;
- **login** – пользовательское имя для регистрации;
- **password** – пароль пользователя на данном узле.

Пароль пишется в незашифрованной форме текста. Если вы включили пароль в файл **.netrc**, то должны запретить чтение/запись этого файла для вашей группы и всех остальных пользователей. В противном случае **ftp** не позволит вам использовать этот файл.

Если вы не введете пароль в файл, **ftp** спросит вас о нем. Вот пример записи в файле **.netrc**:

```
machine admin login guido password open
```

где: **admin** – имя узла, **guido** – пользователь, который регистрируется в машине **admin**, **open** – пароль пользователя **guido**.

Команды ftp

Когда на экране появляется приглашение **ftp**, вы можете вводить одну из команд, описанных ниже в этом разделе. После выполнения команды снова появится приглашение **ftp**. В зависимости от того, какой режим установлен: многословный или отладочный могут появиться дополнительные сообщения. После каждой команды нужно нажимать **Enter**. Выполнение команды не начнется пока вы не нажмете **Enter**. Если вы сделали ошибку при наборе команды, вы можете использовать клавишу **BACKSPACE** для редактирования текста. Вам нет необходимости вводить имя команды полностью, можно вводить определенное количество символов, достаточное для идентификации. В большинстве случаев это один или два символа от названия команды **ftp**. Однако, лучше не лениться и набирать команды полностью. Дело в том, что клиент **ftp** на сервере **joker** может быть изменен системным администратором. Не все клиенты понимают краткие команды. Например, раньше их можно было использовать. Но в один прекрасный день, программу **ftp** заменили. Поставили более продвинутую версию.

! Эта команда приостанавливает **ftp** и вызывает командный язык (**shell**) на локальной машине. Любой(ые) символ(ы) после восклицательного знака воспринимаются и выполняются как **shell** команды. Затем можно вернуться в **ftp**, выйдя из **shell**. Все **ftp** опции и связанные удаленные машины возвращаются в то же самое состояние, которое было до появления этой команды. Если команда **shell** напечатана в той же строке, где и **!**, то выполняется только эта команда. После выполнения команды программа **ftp** возвращается в командный режим.

Дальше рассмотрим команды интерактивного режима, которые вы будете использовать для подключения к серверу и управления файлами:

- **account** – отправить серверу свой пароль для авторизации;
- **append** – добавить локальный файл в конец удаленного

файла на сервере;

- **ascii** – передавать файлы в кодировке ASCII;
- **binary** – передавать файлы в бинарном формате;
- **bye** – завершить сессию;
- **cd** – перейти в папку;
- **cdup** – перейти в папку уровнем выше;
- **chmod** – изменить права на файл;
- **delete** – удалить файл с сервера;
- **dir** – посмотреть список файлов в папке;
- **get** – скачать файл из сервера на локальную машину;
- **hash** – отображать статус передачи файла;
- **lcd** – сменить рабочую директорию на локальной машине;
- **ls** – список файлов в папке;
- **mdelete** – удалить несколько файлов;
- **mget** – скачать несколько файлов;
- **mkdir** – создать каталог;
- **mput** – передать несколько файлов на сервер;
- **open** – подключится к удаленному FTP серверу;
- **put** – передать файл на сервер;
- **pwd** – вывести текущую папку;

- **recv** – то же самое что и **get**;
- **rename** – переименовать файл на сервере;
- **rmdir** – удалить папку;
- **size** – узнать размер файла;
- **user** – авторизоваться на сервере.

Рассмотрим более подробно некоторые из этих команд

Команда **append** заставляет **ftp** добавлять содержимое локального файла в конец файла удаленной машины, с которой вы соединены в данный момент. Когда вы вызываете эту команду, вы можете указать какой файл к какому добавить, например:

```
ftp> append имя_локального_файла имя_файла_удаленной_машины
```

Вы можете также использовать только имя команды, а затем получить запрос об имени файлов, например:

```
ftp> append
(local-file) имя_локального_файла
(remote-file) имя_файла_удаленной_машины
```

Команда **ascii** даёт указание ftp преобразовывать файлы в ASCII код.

Команда **bell** указывает, что на вашем терминале появляется сигнал после завершения передачи каждого файла. Чтобы прекратить подачу сигнала, нужно снова набрать эту команду **ftp**.

Команда **binary** указывает, что необходимо передавать файл в двоичном коде.

Команда **quit** осуществляет выход из **ftp**. Эта команда закрывает все открытые связи.

Команда **cd** переходит в другой каталог на удаленной машине. Вы можете записать новое имя, когда вызываете команду, как показано в примере:


```
ftp> cd /usr/bin
```

Вы можете использовать только имя команды **ftp**, тогда машина запросит имя нового каталога, например:

```
ftp> cd  
(remote-directory) /usr/bin
```

По команде **close** закрывается текущая связь. Однако, не происходит выхода из **ftp**.

Команда **debug** включает и выключает отладочный режим. Если режим включается, то об этом появляется сообщение на экране, при выключении сообщений нет.

Команда **verbose** включает и выключает многословный режим. Если режим включается, то об этом появляется сообщение на вашем дисплее, при выключении сообщений нет.

Команда **delete** удаляется файл в удаленной машине. Вы можете указать имя файла, который нужно удалить, при вызове **ftp** команды:

```
ftp> delete имя файла для удаления
```

Если вам больше нравится, то вы можете не указывать имя при вызове команды **ftp**. Тогда машина вас запросит об имени, например:

```
ftp> delete  
(remote-file) имя файла для удаления
```

Команда **dir** выводит содержимое каталога на удаленной машине. Вы можете задать имя каталога, при вызове команды **ftp**. Например:

```
ftp> dir /usr/bin
```

Если вы не указали имя каталога, то будет выведен текущий каталог. Вы можете вывести результаты выполнения команды в файл. Делается это следующим образом:

```
ftp> dir /usr/bin printfile
```

Вы должны указать имя каталога до имени выходного файла (здесь `printfile`). Если вы хотите вывести текущий каталог в файл, необходимо ввести команду:

```
ftp> dir . printfile
```

Команда **get** копирует файл из удаленной машины, к которой вы в данный момент подсоединены. Этот файл копируется в ваш каталог на сервере. Используйте команду **mget** для копирования нескольких файлов одновременно. Когда вы вызываете эту команду, вы можете указать имя файла на удаленной машине и имя в вашем каталоге, в который вы скопируете файл. Например:

```
ftp> get имя_файла_удаленной_машины имя_файла_вашей_машины
```

Если вы просто укажете имя файла удаленной машины, который нужно скопировать, то файл на вашей машине будет иметь то же самое имя. Пример:

```
ftp> get имя_удаленной_машины
```

Вы можете записать только команду **ftp get**. Тогда программа **ftp** запросит вас об имени файла, например:

```
ftp> get
(remote-file) имя файла удаленной машины
(local-file) имя файла вашей машины
```

Команда **glob** заставляет **ftp** запрещать расширение имен файлов UNIX, такое универсальное, как '*'. Эта команда служит и для запрещения и для разрешения расширения таким образом что, если задать ее повторно, то расширение вновь будет разрешено. После разрешения расширения **ftp** ко всем именам файлов добавит расширение при выводе списка файлов.

При использовании команды **hash ftp** будет выдавать на экране символ # после каждого блока данных, который посылается удаленной машиной. Размер блоков данных может быть различным в зависимости от версии программного обеспечения. На сегодняшний день на машине **joker** он 1024 байтам. После ввода этой команды будет напечатан текущий размер блока данных. Эта команда включает и выключает выдачу символа # на экран ее при повторном наборе. Таким образом, Вы сможете оценить скорость передачи файла.

Команда **help** выдает на экран информацию о работе **ftp**. Если после **help** задать имя команды, то появится информация об этой команде. Если набрать просто **help**, то появится список команд **ftp**.

Команда **lsd** изменяет рабочий каталог, используемый **ftp**, на вашей машине. Вы можете задать имя каталога, который Вам нужен как рабочий, например:

```
ftp> lcd /home/student/имя_вашего_каталог
```

Если Вы не задали имя каталога, то будет использоваться текущий каталог.

Команда **ls** выводит содержание каталога удаленной машины. Вы можете задать имя каталога, который вы хотите распечатать. Например:

```
ftp> ls /usr/bin
```

Если вы не задали имя, будет выведен текущий каталог. Можно задать, чтобы результаты выполнения команды были

помещены в файл до появления их на дисплее. Это делается, если указать имя файла на вашей машине, куда следует поместить листинг каталога, например:

```
ftp> ls /usr/bin printfile
```

Например, если вы хотите распечатать текущий каталог в файл с именем **printfile**:

```
ftp> ls . printfile
```

где "." ставится, чтобы подчеркнуть, что каталог текущий.

Команда **mdelete** удаляет файлы на удаленной машине. Вы можете задать имена файлов для удаления при вызове команды. Например:

```
ftp> mdelete имя_1 имя_2 ...
```

В противном случае вы можете просто использовать имя команды. Программа ftp запросит Вас об имени(ах):

```
ftp> mdelete  
remote-files) имя_1 имя_2 ...
```

Команда **mkdir** создает каталог на удаленной машине. Вы можете задать имя каталога при вызове команды, например:

```
ftp> mkdir /u/mydir
```

Если вы не задали имя, ftp запросит вас о нем, например:

```
ftp> mkdir  
(directory-name) имя_каталога
```

Разумеется, для выполнения этой команды вы должны иметь право на запись на удаленной машине.

Команда **mls** получает аббревиатурный список группы файлов текущего каталога на удаленной машине и помещает результат в файл на вашей машине. Вы можете задать список файлов удаленной машины и файл вашей машины, куда поместить результат при вызове команды, например:

```
ftp> mls имя_1файла_удаленной_машины ... printfile
```

Вы можете не указывать имена при вызове команды, и тогда программа запросит вас о них:

```
ftp> mls
(remote-files) имя_1файла_удаленной_машины ... printfile
local-file printfile? y
```

Команда **mput** копирует один или более файлов из вашего каталога на удаленную машину. На удаленной машине файлы будут иметь те же имена. Разумеется, для выполнения этой команды вы должны иметь право на запись на удаленной машине. Вы можете задать список файлов при вызове команды, например:

```
ftp> mput 1файл_вашего_каталога 2файл_вашей_машины ...
```

Если вы не указали имена, программа **ftp** вас об этом спросит:

```
ftp> mput
(local-files) имя_1файла_вашего_каталога имя_2файла ...
```

Используйте команду **nmap** для установки и отмены механизма преобразования имен файлов. Эта команда полезна при связи с UNIX-несовместимыми машинами в которых используется другой способ именования файлов. Например, серверов на базе Apple Macintosh. Переименовывание файлов локальной машины производится с помощью **get** и **mget** команд, а удаленной машины с помощью **put** и **mput** команд.

Используйте команду **ntrans** для включения и выключения механизма трансляции символов имени файла. Эта команда полезна при связи с UNIX-несовместимой удаленной машиной, где применяется другой способ именования файлов. Например, серверов на базе Apple Macintosh. Трансляция имен файлов локальной машины производится с помощью **get** и **mget** команд, а удаленной машины с помощью **put** и **mput** команд.

Команда **open** устанавливает связь с удаленной машиной, которая предполагается для передачи файлов. При вызове команды вы можете указать имя машины, например:

```
ftp> open ftp.botik.ru
```

Если имя не указано, программа запросит его:

```
ftp> open  
(to) имя машины
```

Если Вы указали имя машины при вызове команды, вы можете также указать номер порта на удаленной машине. Если указан порт, то **ftp** откроет связь через этот порт, если он предпочтительнее того, что задается по умолчанию. Изменения порта делаются в том случае, если вас об этом попросят или это задается системным администратором. Если порт не задан, то программа его не запрашивает.

Команда **prompt** предотвращает ваш запрос к **ftp** о разрешении на переход между файлами в многофайловых командах, таких как **mget**. Эта команда подключается и отключается при повторном наборе.

Команда **put** перемещает файл из вашей машины в удаленную машину. Для передачи нескольких файлов одновременно используйте команду **mput**. Вы можете задать имя файла вашей машины и имя файла удаленной машины при вызове команды **ftp**, например:

```
ftp> put имя_вашего_файла имя_файла_удаленной_машины
```

или

```
ftp> put имя_вашего_файла
```

Разумеется, для выполнения этой команды вы должны иметь право на запись на удаленной машине. Если вы не указали имя файлов(а), то программа вас о них запросит, например:

```
ftp> put  
(local-file) имя_вашего_файла  
(remote-file) имя_файла_удаленной_машины
```

Если Вы не укажете имя файла удаленной машины, то команда **put** создаст файл на удаленной машине под тем же именем, что и на вашей машине.

Команда **pwd** выводит имя текущего рабочего каталога на удаленной машине.

Команда **bye** аналогична команде **quit**, о которой говорилось выше.

Команда **quote** заставляет **ftp** посылать параметры, которые вы вводите на удаленную машину для выполнения.

Команда **recv** аналогична команде **get**, описанной выше.

Команда **remotehelp** запрашивает помощь **ftp** на удаленной машине, с которой вы связаны в данный момент. Эта информация сообщает о том какие команды поддерживает удаленная машина.

Команда **rename** переименовывает файл на удаленной машине. При вызове команды вы можете использовать имена файлов, например:

```
ftp> rename старое_имя_файла новое_имя_файла
```

Если вы использовали только имя команды, то **ftp** запрашивает имена файлов:

```
ftp> rename  
(from-name) старое_имя_файла  
(to-name) новое_имя_файла
```

Разумеется, для выполнения этой команды вы должны иметь право на запись на удаленной машине.

Команда **rmdir** удаляет каталог на удаленной машине. Вы можете задать имя каталога, который следует удалить, при вызове команды, например:

```
ftp> rmdir /u/mydir
```

или вы можете не задавать имя при вызове команды, и машина запросит вас о нем:

```
ftp> rmdir  
(directory-name) /u/mydir
```

Эта команда не всегда поддерживается удаленной машиной. Разумеется, для ее выполнения вы должны иметь право на запись на удаленной машине.

Команда **send** аналогична команде **put**, описанной выше.

Команда **sendport** заставляет **ftp** запрещать возможность задания порта локальной машины для данных удаленной машины. Эта команда может подключаться и отключаться ее повторным набором. При вызове **ftp** по умолчанию задается определенный порт. Эту команду следует использовать по совету вашего системного администратора. Как правило, она нужна только при работе с каким-либо "неправильным" сервером **ftp**, который несовместим с вашей программой-клиентом.

Команда **status** заставляет **ftp** изображать свой текущий статус на вашем терминале. **status** включает режимы, которые выбраны командами **bell**, **hash**, **glob**, **port**, **type**.

Команда **type** устанавливает в каком виде передается файл. Допустимы коды ASCII и двоичный. Эта команда аналогична командам **ascii** и **binary**. Если вы не указали тип при вызове команды, то устанавливается ASCII.

Команда **user** позволяет вам идентифицировать самого себя на удаленной машине при установлении связи. Эта авторегистрация разрешена с опцией **-n** при вызове **ftp**. В этом случае данная команда не нужна. Если автоматическая регистрация запрещена, то используйте эту команду для собственной регистрации и идентификации на удаленной машине. Удаленной машине нужно сообщить три вида информации о том, кто вы: регистрационное имя, пароль и имя ресурса. Имя пользователя требуется для всех машин, пароль и имя ресурса требуются только для некоторых систем. Вы можете ввести всю эту информацию при вызове команды **user**, например:

```
ftp> user mike cat myaccount
```

Вы можете не указывать всю эту информацию при вызове команды. Тогда программа вас о ней запросит, например:

```
ftp> user
(username) mike(имя пользователя)
password:
Account: myaccount(имя ресурса)
```

Заметьте, что ваш пароль не отображается при его наборе для защиты вашей информации. Если вы не набрали пароль или имя ресурса, то запроса о них не будет.

Команда **verbose** заставляет **ftp** запрещать многословный режим. Эта команда включается и выключается при повторном наборе. В многословном режиме **ftp** протокольные сообщения, посланные удаленной машиной появляются на вашем терминале. Кроме того, в этом режиме отображается статистика после передачи каждого файла. Если этот режим запрещен, то данная информация не изображается.

Примеры работы ftp

В двух примерах ниже используются две машины: локальная машина с именем **HERE** и удаленная машина с именем **THERE**.

Пример 1:

```
$ ftp THERE
Connected to THERE
220 THERE FTP server (Version 4.160 #1) ready
Name (THERE:stevea):
Password (THERE:stevea):
331 Password required for stevea.
230 User stevea logged in.
ftp> verbose
Verbose mode off.
ftp> cd /etc
ftp> ls
passwd
volcopy
whodo
ftp> get passwd
ftp>put wall
ftp> ls
passwd
volcopy
wall
whodo
ftp> bye
$
```

Этот пример показывает использование **ftp** для отправки и приема файлов. Команда **ftp** вызывается с именем главной машины и происходит автоматическая регистрация пользователя на другой машине, т.к. не используется опция **-n**.

Многословный режим запрещен использованием команды **verbose** (по умолчанию многословный режим включен. Явное указание **verbose** приравнивается к его повтору и включает многословный режим). Пользователь затем изменяет рабочий каталог на удаленной машине на **/etc** каталог. Т.к.

нет опции **-d** и есть команда **verbose**, многословный режим запрещен и никаких сообщений, кроме приглашения **ftp**, не появляется.

Пользователь, используя команду **ls**, получает аббревиатурный листинг каталога **/etc** машины **THERE**. Команда **ftp** выдает три файла в каталоге **/etc**. Команда **get passwd** затем копирует файл **passwd** с машины **THERE** в машину **HERE**. Файл с именем **passwd** создается в машине **HERE**, если было указано имя машины.

Команда **put** используется, чтобы скопировать файл по имени **wall** из текущего рабочего каталога локальной машины (**HERE**) в каталог **/etc** удаленной машины (**THERE**). Этот файл копируется с тем же именем, т. к. новое имя не было указано. После окончания передачи, появляется листинг **/etc**, в котором уже четыре файла, включая **wall**.

Затем используется команда **bye**, чтобы вернуться в оболочку операционной системы на локальной машине **HERE**.

Пример 2:

```
$ ftp THERE
Connected to THERE
220 THERE FTP server (Version 4.160 #1) ready
Name (THERE:stevea):
Password (THERE:stevea):
331 Password required for stevea.
ftp> debug
Debugging on (debug = 1)
ftp> cd /etc
---> CDW /etc
200 CDW command okay.
ftp> pwd
---> PWD
251
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
```

```

ftp> get wall mefile
---> PORT 3,20,0,2,4,51
200 PORT command okay.
---> RETR wall
150 Opening data connection for wall
(3.20.0.2.1075)(24384bytes
#####.
226 Transfer complete.
24550 bytes received in 12.00 seconds (2 Kbytes/s)
ftp> close
---> QUIT
221 Goodbye.
ftp> bye
$

```

Этот пример иллюстрирует экранные изображения, которые можно получить, используя несколько опций **ftp**. После вызова **ftp** с именем удаленной главной машины, пользователь вызывает команду, которая позволит вернуть режим отладки. После этого команда **ftp** выдает сообщения на экран, что свидетельствует о разрешении данной опции. Пользователь затем заменяет рабочий каталог на удаленной машине на **/etc**. Т.к. присутствуют отладочный и многословный режимы, на экране будут появляться сообщения о посылке команд к удаленной машине (**- -> CWD/etc**) и ответы, полученные с удаленной машины (**250 CWD command successful** - команда прошла успешно). Заметьте, что **cd** – команда, которая имеет такую же форму, как команда изменения каталога в **UNIX**, посылается в виде **CWD** команды (изменение рабочего каталога на удаленной машине). Эта команда применяется **ftp** вместо **cd**, чтобы она работала независимо от системной команды.

Следом за командой **cd** пользователь ставит команду **pwd**, чтобы подтвердить изменение рабочего каталога. Команды **ftp** будут изображать на вашем экране сообщения, посылаемые между вашей (локальной) и удаленной маши-

ной, а затем появится текущий рабочий каталог на удаленной машине. Затем пользователь набирает **hash** опцию, на что появляется сообщение, что данная опция разрешена. Команда **get wall myfile** приказывает **ftp** восстановить файл **wall** и поместить его в файл **myfile** в текущем рабочем каталоге на вашей машине. Команда **ftp** выдаст сообщения, посылаемые между локальной и удаленной машинами о начале передачи и затем печатает **hash** метку для каждого блока, полученной информации. После завершения передачи на экране появляется статистика, в которой время, потребовавшееся для передачи, и дата передачи файла. После получения файла пользователь закрывает связь командой **close** и выходит из **ftp** командой **bye**.

Глава 17

Лабораторная работа №11. FТР-сервер

ЛК — действие должно выполняться на локальном компьютере **УК** — действие должно выполняться на локальном компьютере, но под именем пользователя удаленной машины (т. е. по **ftp**).

Задание 0

Установите **vsftpd** на компьютеры

Задание 1

Настройте нормальный режим работы на компьютерах

Задание 2

Настройте анонимный (от имени **anonymous**) доступ по **ftp** без запроса пароля.

Попробуйте анонимно подключиться к другому компьютеру по **ftp** и передать на него любой файл. Выведите на экран текущий каталог, а затем его содержимое.

Задание 3

- Настройте доступ по **ftp** только пользователям **user1** и **user2** (пользователи должны существовать на сервере. Вход на сервер будет производиться от их имени и по их паролю).

- Разрешите запись авторизированным пользователям.
- Установите максимальное количество пользователей, одновременно использующих **ftp** сервер – 2.
- Убедитесь, что третий пользователь не может использовать **ftp** (т. е. попробуйте создать соединение более чем из двух машин).

Задание 4

ЛК: создайте три пустых файла: **name.txt** **age.txt** **color.txt**.

ЛК: откройте соединение по **ftp**.

УК: вывести текущий каталог.

УК: создайте в домашнем каталоге подкаталог **DATA**.

УК: сделайте каталог **DATA** текущим.

УК: переместите файлы **name.txt** **age.txt** **color.txt** с локальной машины на удаленную, в каталог **DATA**.

УК: выведите содержимое каталога **DATA** на экран.

ЛК: в файл **name.txt** впишите своё имя, в файл

age.txt — свой возраст, а в файл **color.txt** — свой любимый цвет.

УК: запишите содержимое своих локальных файлов в удаленные файлы. Для записи использовать одну из команд **ftp**.

УК: переименуйте файл **color** в **favourite_color**.

ЛК: удалите файлы **name.txt** **age.txt** **color.txt**.

УК: передайте файлы **name.txt** **age.txt** **favourite_color.txt** на локальную машину.

УК: удалите файлы **name.txt** **age.txt** **favourite_color.txt**.

УК: перейдите в домашний каталог.

УК: удалите каталог **DATA**.

Задание 5

- Установите **filezilla**.
- Авторизируйтесь.
- Передайте с удаленной машины на локальную любой файл.

В качестве отчёта напишите инструкцию по созданию и настройке локального **FTP**-сервера.

Литература

- [1] *Георгий Курячий, Кирилл Маслинский*. Операционная система Linux. Курс лекций. — М.: ALT LINUX, ДМК Пресс, 2010. — 348с.
- [2] *Шоттс У.* Командная строка Linux. Полное руководство — СПб.: Питер, 2020. — 544с.
- [3] *Негус К. Каэн Ф.* Ubuntu и Debian Linux для продвинутых — СПб.: Питер, 2010. — 352 с.
- [4] Настройка samba в Ubuntu – Losst. URL: <https://losst.pro/nastrojka-samba-v-ubuntu-15-10> (дата обращения 29.03.2023).
- [5] Samba | Русскоязычная документация по Ubuntu. URL: <https://help.ubuntu.ru/wiki/samba> (дата обращения 29.03.2023).
- [6] Установка SSH в Ubuntu – Losst. URL: <https://losst.pro/ustanovka-ssh-ubuntu-16-04> (дата обращения 29.03.2023).
- [7] Как пользоваться SSH – Losst. URL: <https://losst.pro/kak-polzovatsya-ssh> (дата обращения 29.03.2023).
- [8] Сервер FTP | Русскоязычная документация по Ubuntu URL : https://help.ubuntu.ru/wiki/%D1%80%D1%83%D0%BA%D0%BE%D0%B2%D0%BE%D0%B4%D1%81%D1%82%D0%B2%D0%BE_%D0%BF%D0%BE_ubuntu_server/%D1%84%D0%B0%

D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%B5_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%B0/ftp_server (дата обращения 29.03.2023).

- [9] Настройка и использование FTP-сервера в Ubuntu Linux. URL: <http://rus-linux.net/MyLDP/server/ftp.html> (дата обращения 29.03.2023).
- [10] Шпаргалка программиста: Настройка и использование FTP-сервера в Ubuntu Linux. URL: <http://4ybakut2004.blogspot.com/2014/08/ftp-ubuntu-linux.html> (дата обращения 29.03.2023).
- [11] Команды Linux: FTP. URL: <http://rus-linux.net/MyLDP/consol/hdrguide/rusman/ftp.htm> (дата обращения 29.03.2023)
- [12] Установка FTP на Ubuntu 20.04 – Losst. URL: <https://losst.pro/ustanovka-ftp-na-ubuntu-16-04> (дата обращения 29.03.2023)

Учебное издание

Алексеев Евгений Ростиславович,

Грушевский Сергей Павлович,

Дога Кристина Вячеславовна,

Обухова Юлия Николаевна

Основы работы в Linux

Учебное пособие

Подписано в печать **xx.xx.2023**, Формат 60x84 1/16
Печать цифровая. Бумага тип. № 1. Уч-печ. л. **0,0**. Усл. печ.
л. **0.0**

Тираж **50** экз. Заказ №

Кубанский государственный университет
350040, г. Краснодар, ул. Ставрапольская, 149

Тираж изготовлен в типографии
с оригинал-макета заказчика.

3500xx, г. Краснодар,