

Кубанский Государственный Университет  
Факультет Математики и Компьютерных наук  
кафедра информационных образовательных технологий  
Е.Р. Алексеев, К.В. Дога, Ю.Н. Номоконова

«Первое знакомство с ОС Linux»

Учебное пособие



Краснодар, 2022

# Глава 1. Знакомство с ОС Linux

Начнём с базового понятия в данной теме – понятия операционная система.

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 деления дистрибутивов по производителям, существует деление по типам использования:

1. LiveCD – дистрибутив, при использовании которого происходит загрузка операционной системы (рабочего стола) с CD (или USB). После этого с операционной системой можно работать, как с ОС, запущенной с жесткого диска (винчестера).

2. Установочные дистрибутивы – дистрибутивы, с которых осуществляется установка операционной системы 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 environment (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 2. Файловая система ОС Linux

Основными понятие в *файловой системе* ОС Linux – понятия файла и каталога.

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 «/» (корневого 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*<sup>1</sup>, который находится в каталоге */home/user/pas*<sup>2</sup>.

---

1 .pas – расширение файлов программ на языке Pascal.

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

Древо каталогов устроено так, что среди подкаталогов конкретно взятого каталога не может находиться он сам. Это создано 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:

- /bin и /sbin (каталог для хранения исполняемых файлов системных программ);
- /etc (каталог для конфигурационных файлов) ;
- /dev (каталог для файлов внешних устройств) ;
- /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 для данных);

## Глава 3. Терминал Linux и основные команды

Множество действий в Linux могут быть осуществлены только в режиме терминала, а иногда выполнить действия 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**. После чего в окне терминала можно вводить команды вида

***команда параметры***

Однако, ряд действий можно осуществить только с правами администратора, для чего вводится команда<sup>3</sup>:

***sudo команда параметры***

После этого система спросит пароль администратора.

В системе доступен текстовый редактор nano, который входит в набор программ GNU и является стандартным консольным редактором для многих дистрибутивов Linux.

Рассмотрим основные команды Linux, которые могут понадобиться рядовому пользователю.

---

3 Право выполнять команду от имени администратора есть у отдельных пользователей

## 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 страницы интерактивного руководства *man*.

### Команда *man*

Команда **man** имеет вид **man name**.

Где **name** - название команды или другого объекта, о котором нужна информация. Например, команда **man pwd** выведет на экран информацию о команде **pwd**

Страницы *man* просматриваются с помощью команды *less* (для получения возможности перемещаться по страницам руководства). 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 имя\_команды | lpr**,

для postscript-принтера – **man -t имя\_команды | 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 man**

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** создает в текущем 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.

```
gedit ./my.txt
```

Вызываем текстовый редактор gedit (требует установки) 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 созданную в нем каталог 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***

**find путь -опции**

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

- `-name` (поиск файлов по имени, используя приведенный шаблон);
- `-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 +n` (поиск файлов, содержание которых изменялось менее чем «-» или более чем «+» дней назад)

**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

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

Пример:

**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 отобразилась 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:

**file1 file2 >file3**

**Строка1**

**cat>file2**

**Строка2**



```
cat>file3
```

Строка3

```
cat file1 file2 >>file3 cat file3
```

В результате на экране появятся три надписи в следующем порядке:

Строка3

Строка1

Строка2

## Права доступа в ОС Linux

В Windows можно определить права доступа на директорию, и они автоматически распространяются на все файлы и поддиректории. В ОС Linux права не наследуются, они у каждого файла свои. Если посмотреть на вывод программы `ls -l`, в первом столбце показаны права доступа файла. Здесь первый символ - это тип файла, остальные девять - права.

```
$ ls -l
-rw-r--r-- 1 evgeniy evgeniy 212 2008-09-12 19:38 1.tex~
-rw-r--r-- 1 evgeniy evgeniy 195 2008-09-12 19:42 2~
-rw-r--r-- 1 evgeniy evgeniy 223 2008-09-12 20:00 2.tex~
drwxr-xr-x 2 evgeniy evgeniy 4096 2008-09-14 13:03 Example
-rw-r--r-- 1 evgeniy evgeniy 1647594 2008-09-12 22:59 Example-1.zip
-rw-r--r-- 1 evgeniy evgeniy 1585850 2008-09-13 19:30 llang.pdf
-rw-r--r-- 1 evgeniy evgeniy 6229739 2008-09-13 19:33 lvovskij-3izd.pdf
-rw-r--r-- 1 evgeniy evgeniy 878 2008-09-12 19:09 newfile1.lyx~
drwxr-xr-x 2 evgeniy evgeniy 4096 2008-09-14 12:40 primer
drwxr-xr-x 2 evgeniy evgeniy 4096 2008-09-14 13:03 proba
drwxr-xr-x 2 evgeniy evgeniy 4096 2008-09-10 13:42 Regulation
-rw----- 1 evgeniy evgeniy 342861 2008-09-12 23:45 Regulation-1.zip
-rw-r--r-- 1 evgeniy evgeniy 23112 2008-09-13 17:15 gedit.odt
```

Типы файлов:

- `b` – специальный блочный файл;
- `d` – каталог;
- `c` – специальный символьный файл;
- `f` – обычный файл;
- `l` – символическая ссылка;
- `p` – именованный канал;
- `s` – сокет.

Права доступа делятся на три группы:

```
r w x | r w x | r w x  
user | group | other
```

- **user** – права хозяина файла;
- **group** – права группы которой принадлежит файл;
- **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 этот 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:

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

## Глава 4. Создание нового пользователя. Изменение прав доступа

Команды управления пользователями:

**sudo adduser *name\_user* (sudo useradd *name\_user*)** - создание нового пользователя.

`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 *name\_user*** – задать пароль пользователю *name\_user* .

**sudo usermod *name\_user*** — изменение параметров учетной записи пользователя *name\_user*.

Параметры:

<b>-d, --home <i>home_dir</i></b>	новый домашний каталог учётной записи
<b>-e, --expiredate <i>date</i></b>	установить дату окончания действия учётной записи в <i>date</i>
<b>-f, --inactive <i>inactivity</i></b>	установить период неактивности пароля после устаревания учётной записи равным <i>inactivity</i>
<b>-h, --help</b>	показать данное сообщение и закончить работу
<b>-l, --login <i>new_name</i></b>	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 <i>pass</i>	задать новый зашифрованный пароль для учётной записи

**sudo deluser *name\_user*** – удаление пользователя *name\_user*.

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 - *name\_user*** (где *name\_user* – это имя пользователя)

## 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 *new\_group 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 приведет 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 *new\_user:new\_group files***

Как и в команде **chgrp**, вы можете использовать как имена пользователей и групп, так и их идентификаторы GID и UID. Команда не проверяет на существование пользователей и группы, 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 zoidberg file.txt
```

Смена владельца на пользователя zoidberg, группа остается без изменений.

```
chown :futurama file.txt
```

Смена группы на futurama, 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 (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 (`гwx`, `г-х` 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 (`г`).

```
chmod a=w finanse.ods
```

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

```
chmod u+rwx,g+rwx,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 -R 744 ~/abc**

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**).

После знакомства с понятием жёстких, символьных ссылок и прав доступа 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**.

Рассмотри на примере

```
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 и права доступа, первый символ определяет тип файла, который может быть.

**d** – каталог,

**-** – обычный файл,

**l** – символьная ссылка.

Следующие 9 символов определяют права доступа, первые три – права доступа владельца, затем права доступа членов группы, последние – права доступа всех остальных.

Число в **третьем столбце** определяет:

- для файлов – количество жёстких ссылок;
- 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 5. Пакеты в ОС Linux

В первой главе 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, которые хранятся в репозитории.

Репозиторий – место, где хранятся и поддерживаются какие-либо данные. Эти данные чаще всего хранятся в виде файлов, доступных для дальнейшего распространения по сети.

Репозиторий дистрибутива – хранилище программ для данного дистрибутива.

Итак, для работы с пакетами и их файлами используются следующие утилиты<sup>4</sup>:

### *Утилита 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.

### *Утилита apt*

- **sudo apt update** – обновление баз данных доступных программных пакетов;
- **sudo apt upgrade** – проверяет наличие обновлений для всех установленных пакетов и предлагает загрузить и установить их;
- **sudo apt install package** – загрузка и установка программного пакета по заданному названию (если он найден в базах данных);
- **sudo apt -f install** - поиск неработающих программных пакетов и попытка отладить те из них, которые отмечены сообщением «unmet dependency» (неудовлетворенные зависимости);
- **sudo apt autoclean** - удаляет не полностью загруженные или ещё не установленные пакеты;
- **sudo apt autoremove** - удаляет не используемые в системе библиотеки;
- **sudo apt remove package** – удаление пакета, оставляя конфигурационные файлы;

---

<sup>4</sup> Описанные ниже утилиты используются в дистрибутивах на базе Debian, а также в российских дистрибутивах «Альт» (<https://www.basealt.ru>)

- **sudo apt purge *package*** – удаление пакета вместе с конфигурационными файлами;
- **sudo apt search *package*** – поиск пакета (по названию и краткому описанию);
- **sudo apt show *package*** – подробная информация о пакете;
- **sudo apt deb *package*** – установка скачанного локального пакета.

## Установка приложений

Стандартный способ установки включает в себя две команды:

### 1. **sudo apt update**

Обновление базы данных пакетов

### 2. **sudo apt install *name***

Непосредственно установка, где *name* — это название приложения.

Так как команды требуют прав администратора, используется **sudo**.

Установка пакетов из стандартных репозиториях считается самой надежной, так как кроме самого пакета, устанавливаются все необходимые зависимости. Нужно это для предотвращения различного рода ошибок и синхронного обновления приложения в системе.

Однако случается, что требуемого ПО нет в стандартных репозиториях. Тогда приложение необходимо скачать в виде deb пакета и установить с помощью одной из команд:

### **sudo dpkg -i *file***

здесь *file* — полный путь к deb пакету, который нужно установить.

Такой командой устанавливается только указанный пакет, то есть с ее помощью нельзя установить все необходимые зависимости. Это значит, что при обновлении (**sudo apt update**) могут возникнуть предупреждения об отсутствии этих самых зависимостей. Проблема исправляется командой **sudo apt install -f**

### **sudo apt install *.file* *.deb***

Этой командой устанавливаются приложение со всеми необходимыми зависимостями.

## Глава 6. Настройка начального загрузчика

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

После установки Linux на ПК, 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 терминал.

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"

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify     ^C Cur Pos
^X Exit          ^R Read File    ^\ Replace      ^U Paste Text  ^T To Spell    ^ Go To Line
```

После ввода пароля откроется окно, в котором нужно 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 пользователь видит загрузчик 10 секунд. Это можно изменить, указав здесь другое число (в секундах).

После внесения изменений нажать **Ctrl+X** для выхода из редактора, **Y**, для сохранения изменений и **Enter** для подтверждения проведенных действий.

Обязательно вводим команду

**sudo update-grub**

для обновления загрузчика.



После этого можно перезагружать компьютер.

## 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
```

```
$ ln -li -R
```

```
ls -li -R
```

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

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

```
3:
```

```
итого 4
```

```
9190023 -rw-rw-r-- 1 aer aer 9 окт 25 10:48 third.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 данной файловой системе.

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.

## Символьные ссылки

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 пример из предыдущего параграфа, изменив 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. Установка Linux Mint

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

Для того чтобы установить дистрибутив Linux Mint 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.

### Создание образа ОС Linux

На сайте <https://linuxmint.com/download.php> скачать актуальную версию дистрибутива, файл в формате iso.

### Запись образа на USB

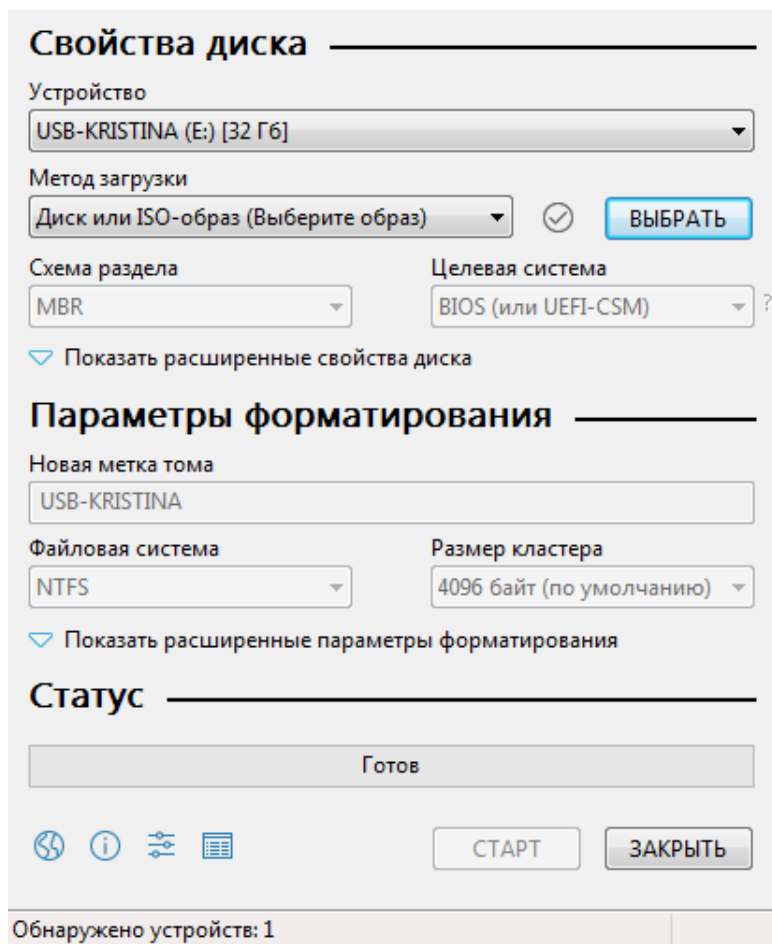
Для записи образа ОС на USB под управлением Windows необходима сторонняя программа Rufus. Скачать её можно с сайта <https://rufus.ie/ru/>

*При записи образа на USB, устройство форматируется. Все важные данные необходимо сохранить на другом носителе.*

После запуска программы необходимо выбрать:

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»

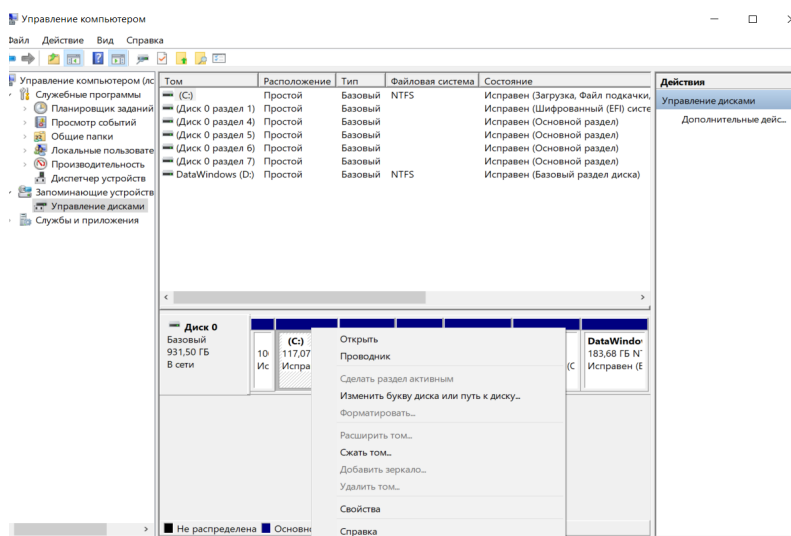
И далее подтвердить форматирование устройства.



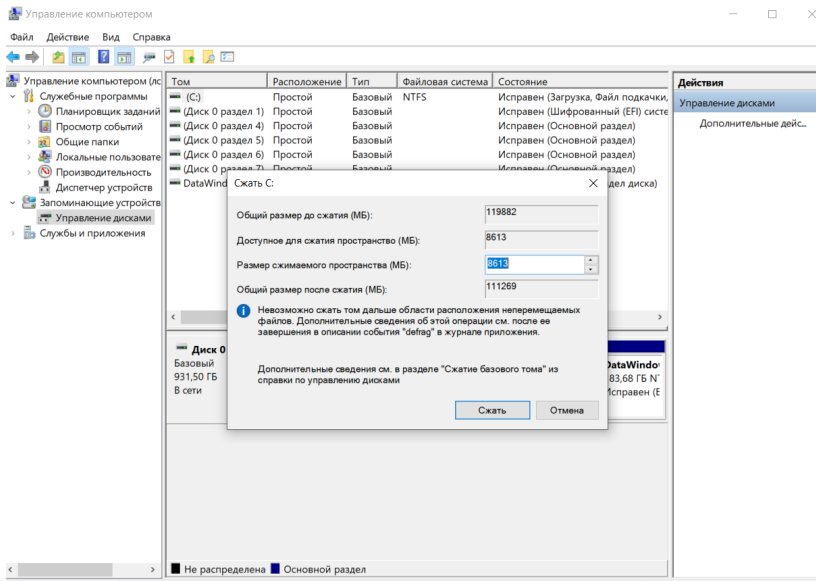
## Установка Linux Mint рядом с Windows (Пример Windows 10)

Чтобы установить Linux рядом с Windows необходимо предварительно выделить на диске место, которое будет использоваться ОС Linux.

Для этого нужно через меню *Пуск* найти утилиту *Управление компьютером*. В списке слева выбрать пункт *Управление дисками*, 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»



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



В итоге 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 по разному. Необходимо при включении компьютера удерживать определённую клавишу, но у разных производителей она отличается. Обычно это: F2, F3, F10, F11, Esc, Del. После изменения порядка загрузки требуется сохранить изменения и перезагрузить компьютер.

*Загрузочная флешка должна быть вставлена в USB порт до перезагрузки.*

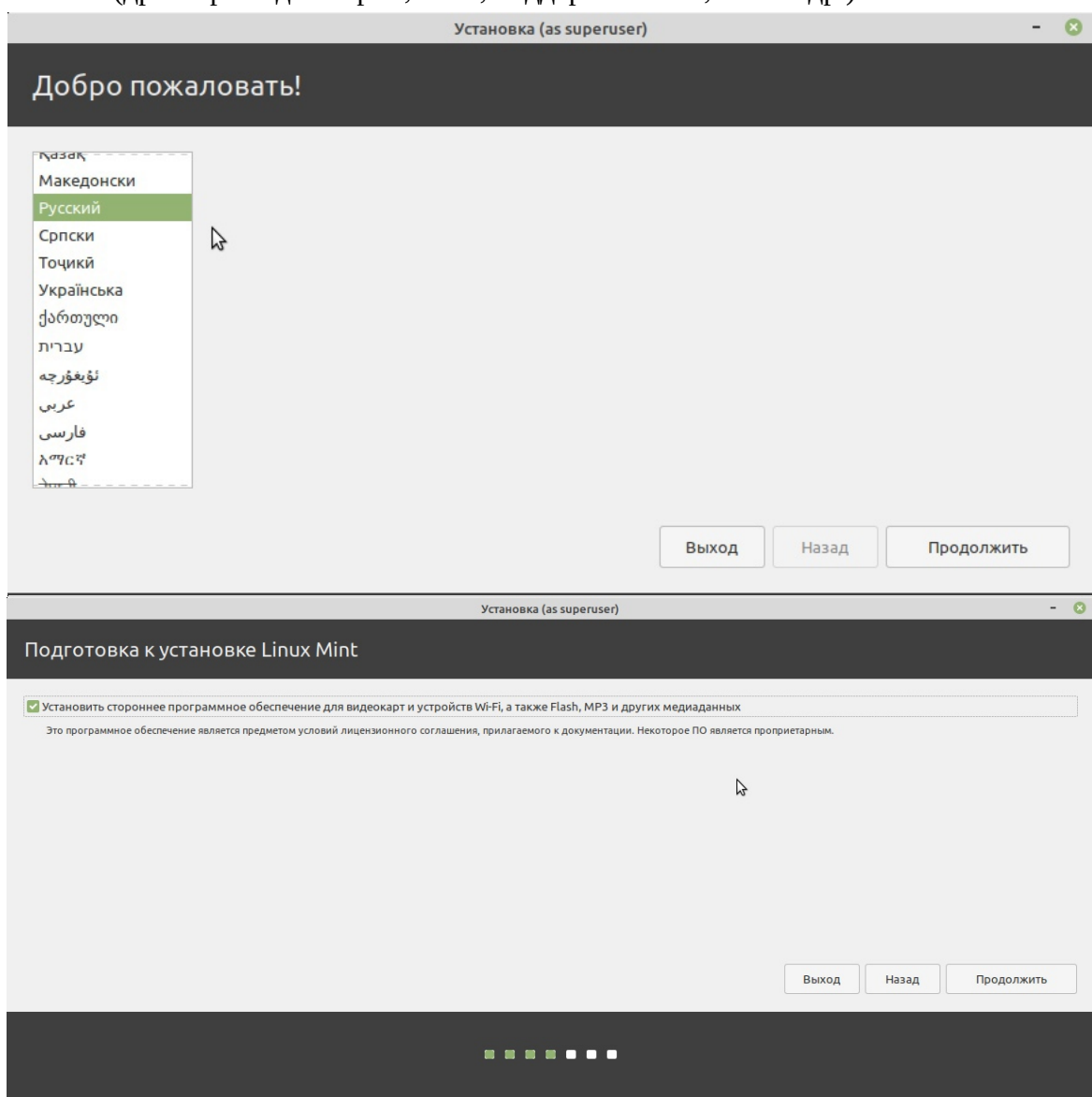
После перезапуска начнётся загрузка Linux Mint в Live режиме, появится загрузочное меню. Выбрать пункт «Start Linux Mint», после чего появится рабочий стол.

## Установка Linux Mint

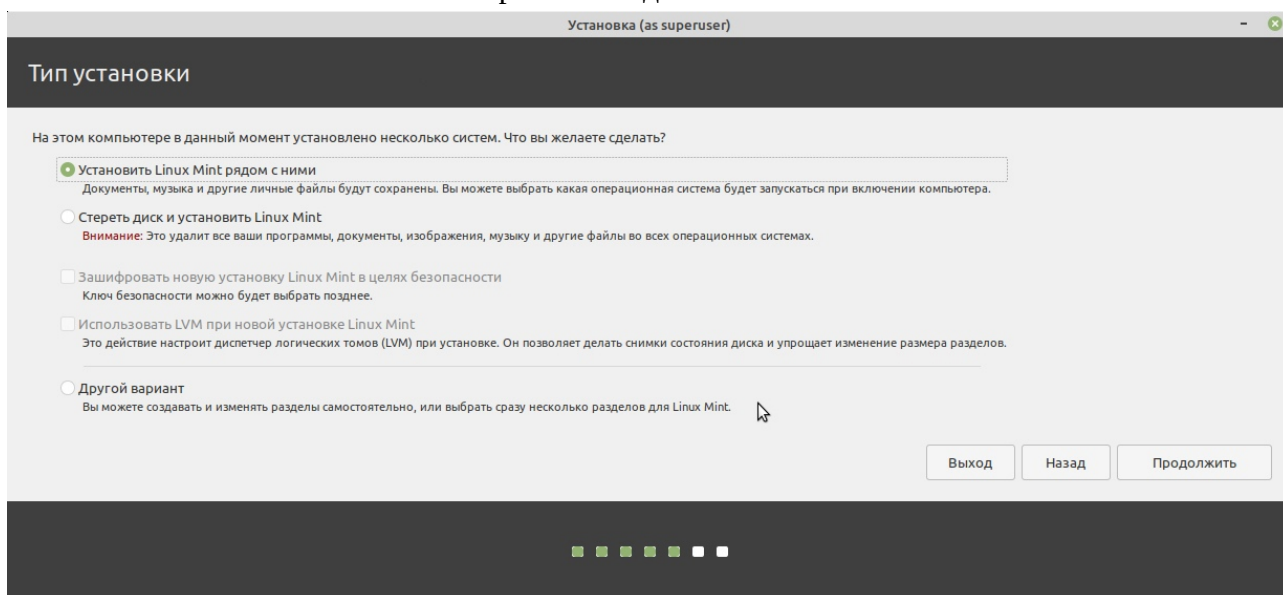


Для начала установки системы на компьютер нужно дважды щелкнуть по иконке Install Linux Mint, которая расположена на рабочем столе.

После этого запускается инсталлятор. В открывшемся окне необходимо последовательно выбрать язык будущей системы, раскладку клавиатуры, установить дополнительные компоненты (драйвера видео-карты, WiFi, поддержка Flash, MP3 и др.).



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



### Отдельно о разметке диска

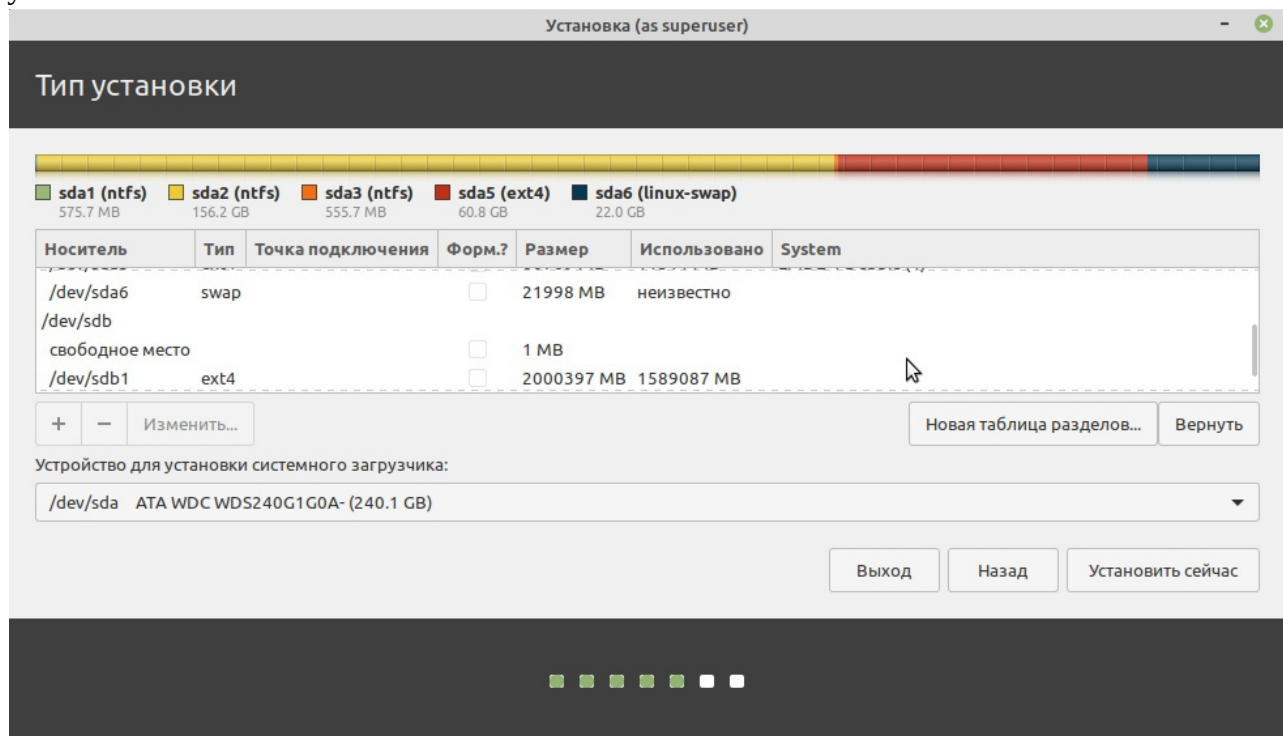
Создаваемые на диске разделы могут отличаться размером, файловой системой, назначением. Linux может занимать несколько разделов, каждый из которых служит для своих целей.

/ («корень») - Основной (корневой) раздел, в котором размещаются все каталоги системы.

/home - Домашний раздел, в котором размещаются пользовательские данные.

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 Windows), и строка «свободное место». Это место, которое заранее выделялось для установки Linux.



Сначала необходимо создать swap-раздел (раздел подкачки)

Для этого 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. Мы рекомендуем отводить для раздела подкачки 1-2 объёма оперативной памяти<sup>5</sup>.

2. Местоположение нового раздела. Начало этого пространства.

3. Использовать как: раздел подкачки.

Теперь нужно создать корневой раздел «/» и раздел «/home».

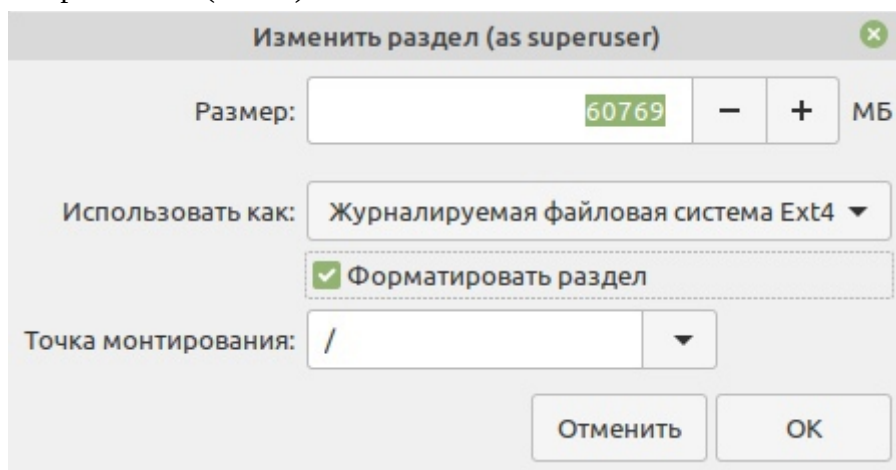
Для этого в таблице разделов снова выделить поле «свободное место» и нажать «+». В настройках:

1. Указать размер раздела в мегабайтах. (Минимум 20 Gb для «/» и оставшееся для «/home»);

2. Местоположение нового раздела: Начало этого пространства;

3. Использовать как: Журналируемая файловая система Ext4;

4. Точка монтирования: / (/home)



Изменить раздел (as superuser)

Размер: 60769 - + МБ

Использовать как: Журналируемая файловая система Ext4 ▼

Форматировать раздел

Точка монтирования: / ▼

Отменить ОК

Нажать кнопку «продолжить», выбрать часовой пояс и создать пользователя.

При создании пользователя указать:

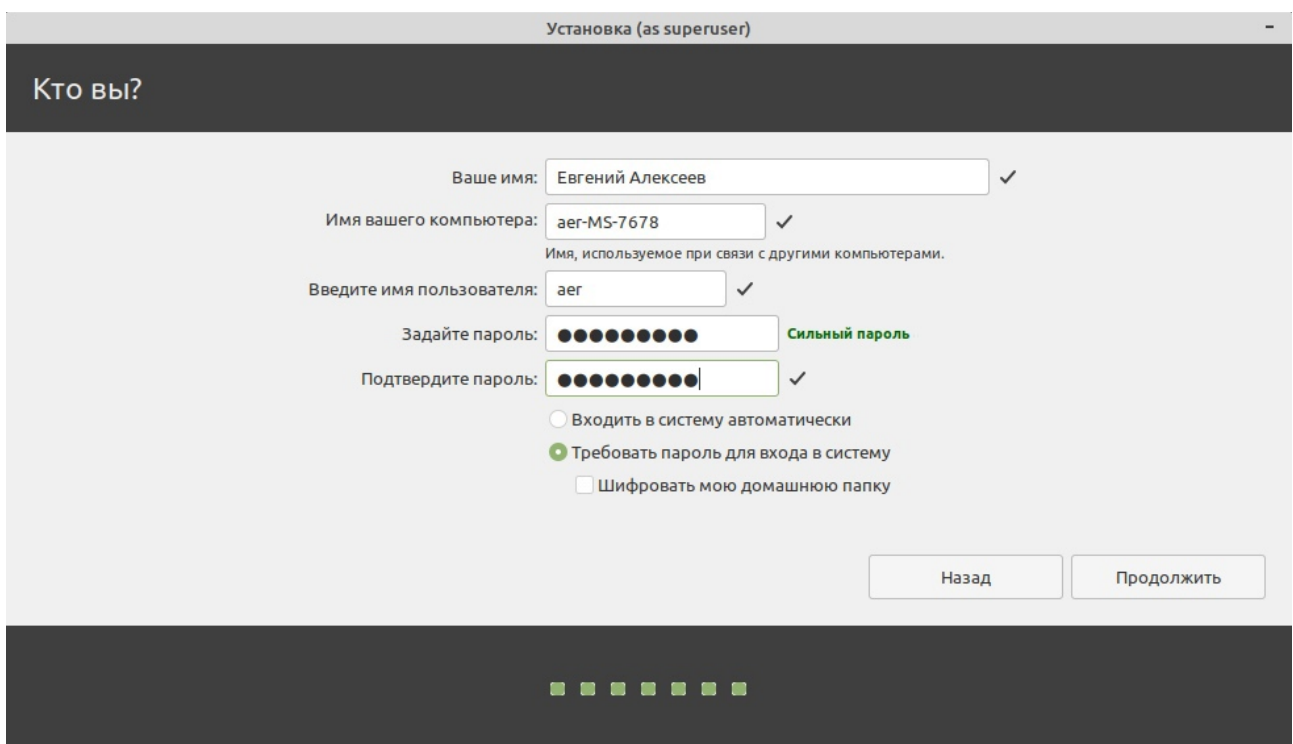
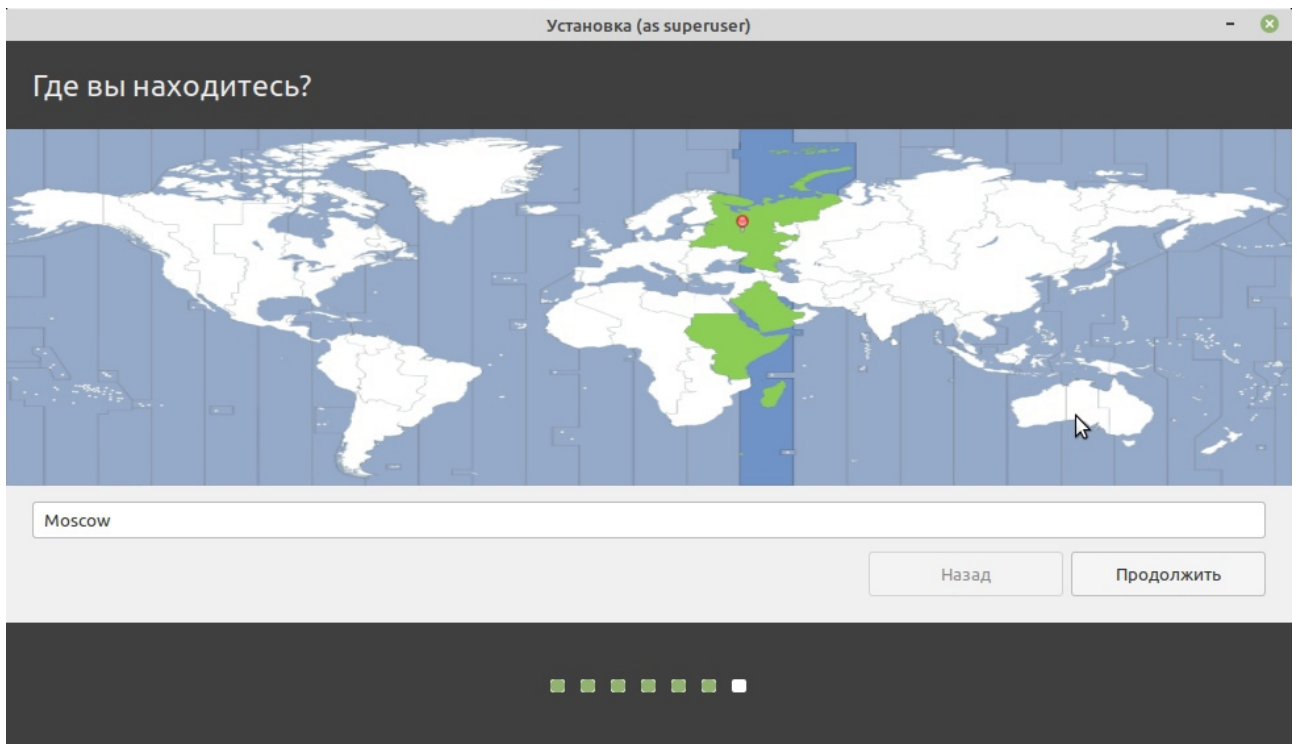
1. Ваше имя
2. Имя Вашего компьютера
3. Имя пользователя
4. Пароль и повтор пароля

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

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

<sup>5</sup> Если у Вас 16 Гб ОЗУ и более, то можно и не использовать раздел подкачки. В ряде дистрибутивов автоматически создаётся файл подкачки.





После указания всех настроек нажать кнопку «Продолжить».

Начинается процесс установки. 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).

## Добро пожаловать в Linux Mint

Добро пожаловать и спасибо за выбор Linux Mint! Пока система будет устанавливаться на компьютер, можно посмотреть это ознакомительное слайд-шоу.



► Установка системы

Skip

Видео установки Linux Ubuntu 20.04 можно увидеть тут [https://vk.com/video/@aer64?z=video7785282\\_456239349](https://vk.com/video/@aer64?z=video7785282_456239349).

## Глава 9. 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. – 352с.

## **Глава 10. Приложение. Лабораторные работы «Основы работы в ОС Linux»**

## **Лабораторная работа №1**

Установите ОС семейства Linux (предпочтительно Debian, Ubuntu, Linux Mint или их клоны) 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](https://disk.yandex.ru/d/oyeIzh586_qwSA) и <https://disk.yandex.ru/d/LV0VS5UTN9AmCw> ).  
Получите справочную информацию о командах (см. свой порядковый номер в списке) и перенаправьте ее в файл lab1.txt.

B-1. whatis, touch

B-2. apropos, cat

B-3. cp, file

B-4. mv, adduser

B-5. which, find

B-6. tac, id

B-7. rm, useradd

B-8. less, chmod

B-9. mkdir, chown

B-10. cd, chgrp

B-11. ls, su

B-12. rmdir, cal

B-13. tree, locate

B-14. sudo, rm-r

B-15. pwd, sudo

## Лабораторная работа №3

1. Создать каталог Var(номер варианта).
2. В каталоге создать дерево каталогов как показано на рисунке.
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 hooby.txt, name.txt, school.txt в файле all.txt каталога Info.
9. Удалите файлы hooby.txt, name.txt, school.txt.
10. Выведите содержимое файла all.txt на экран.
11. Скопировать файл all.txt в директорию A1. Изображено на рисунках 1-15.

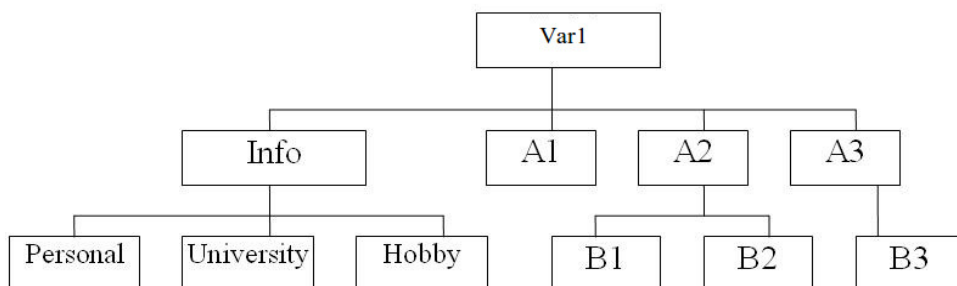


Рисунок 1 – Вариант 1

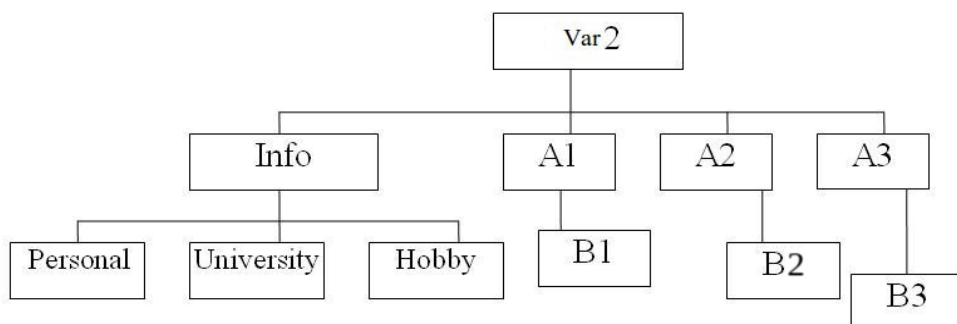


Рисунок 2 – Вариант 2

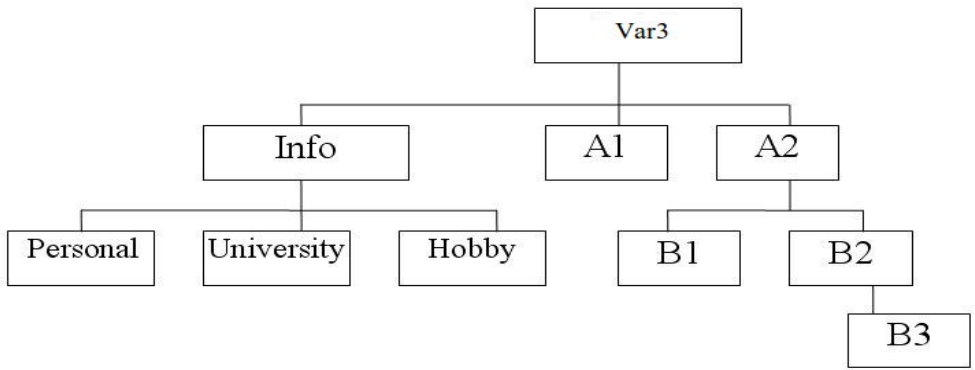


Рисунок 3 – Вариант 3

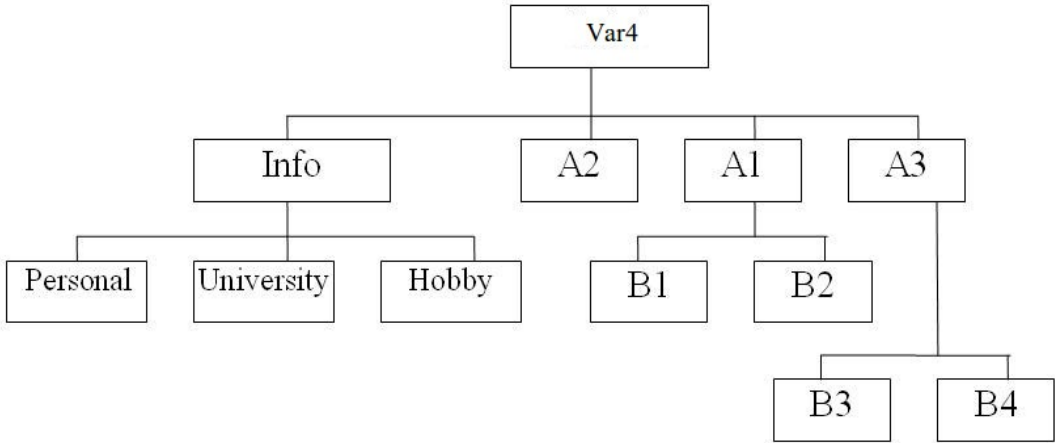


Рисунок 4 – Вариант 4

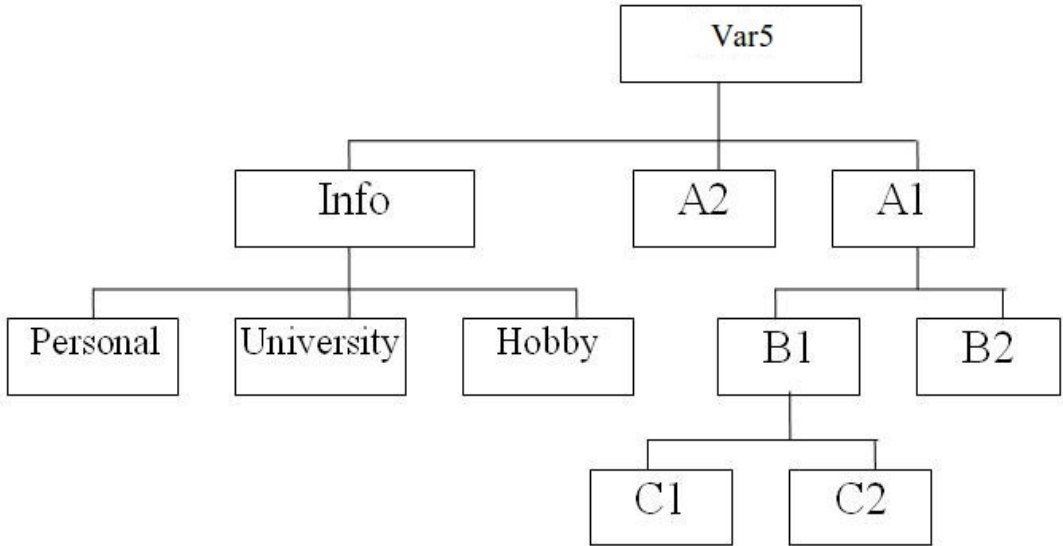


Рисунок 5 – Вариант 5



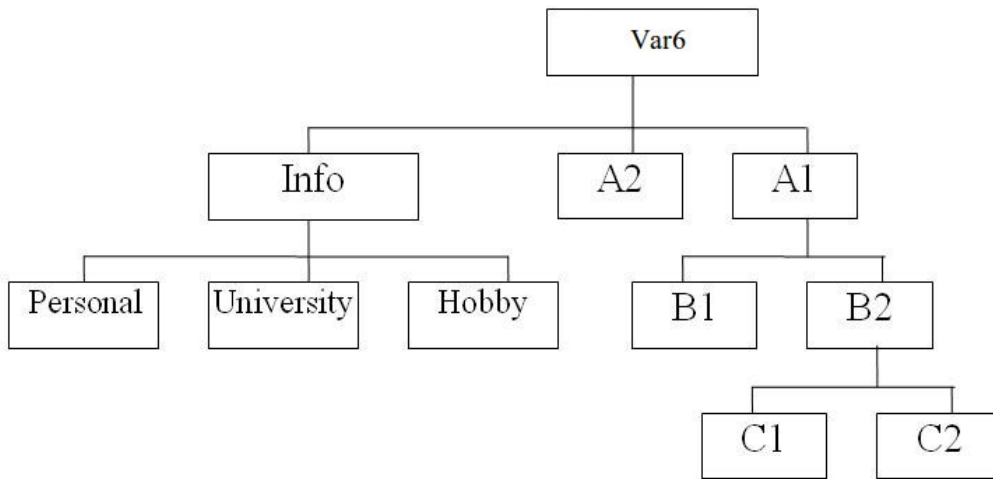


Рисунок 6 – Вариант 6

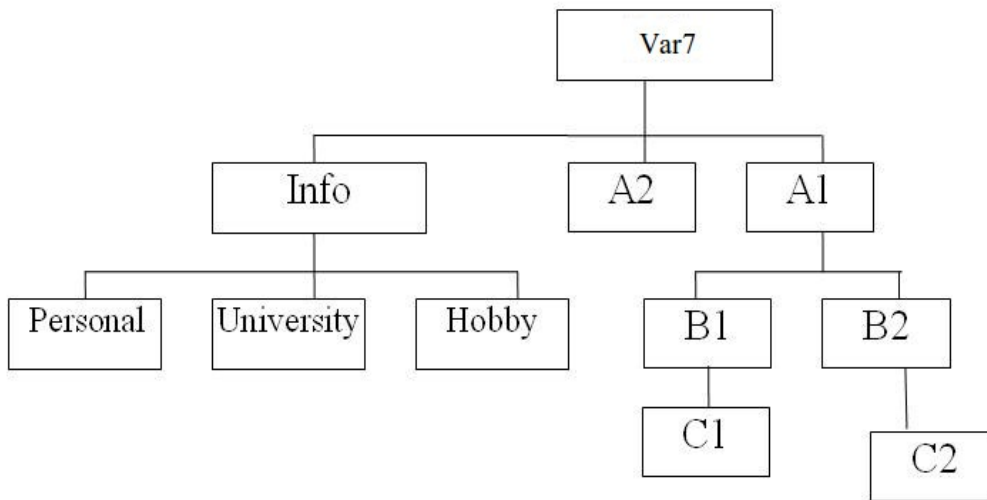


Рисунок 7 – Вариант 7

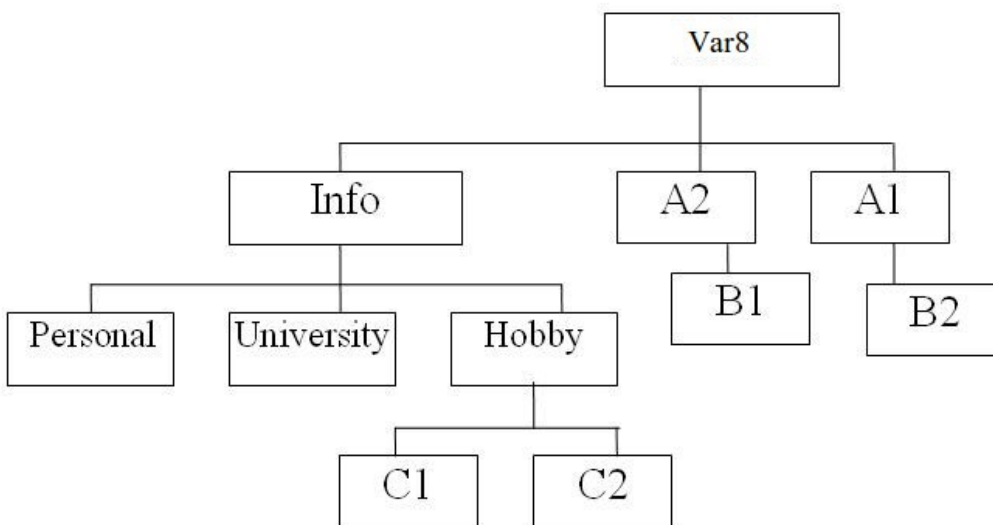


Рисунок 8 – Вариант 8

Рисунок 9 – Вариант 9

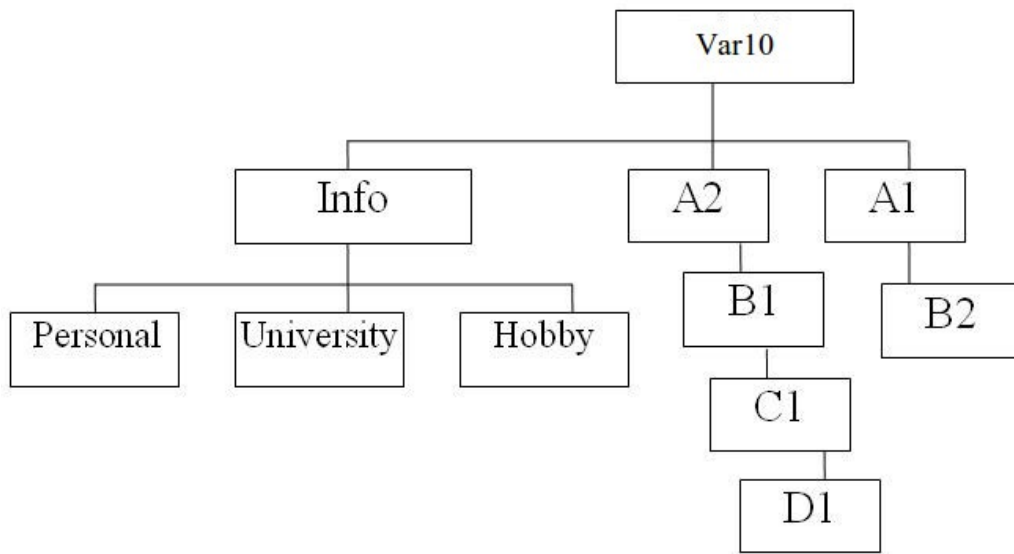
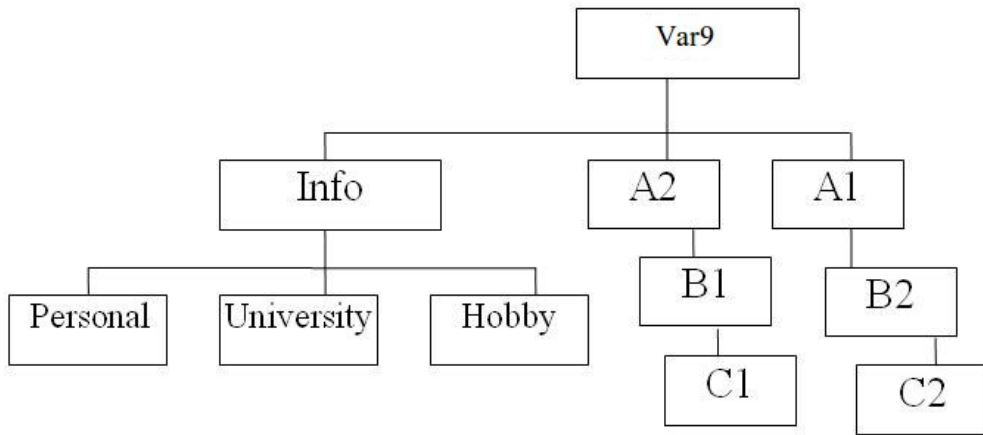


Рисунок 10 – Вариант 10

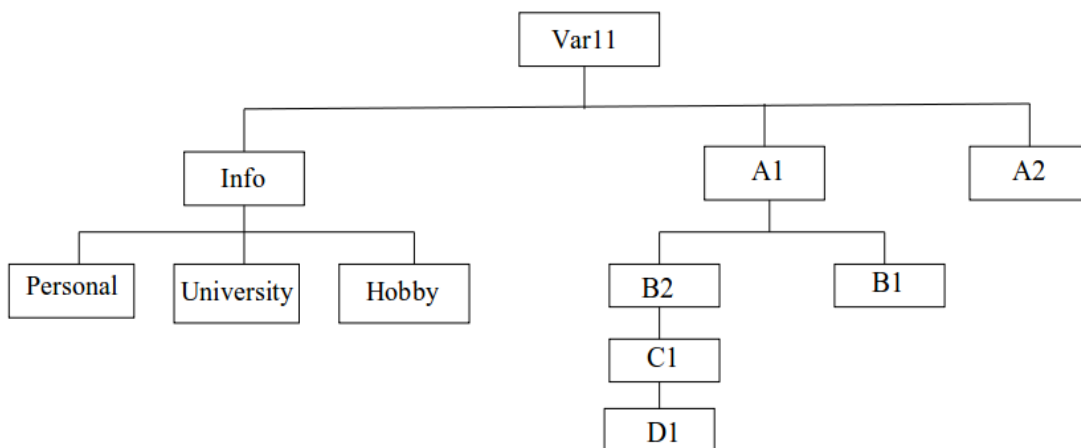


Рисунок 11 – Вариант 11

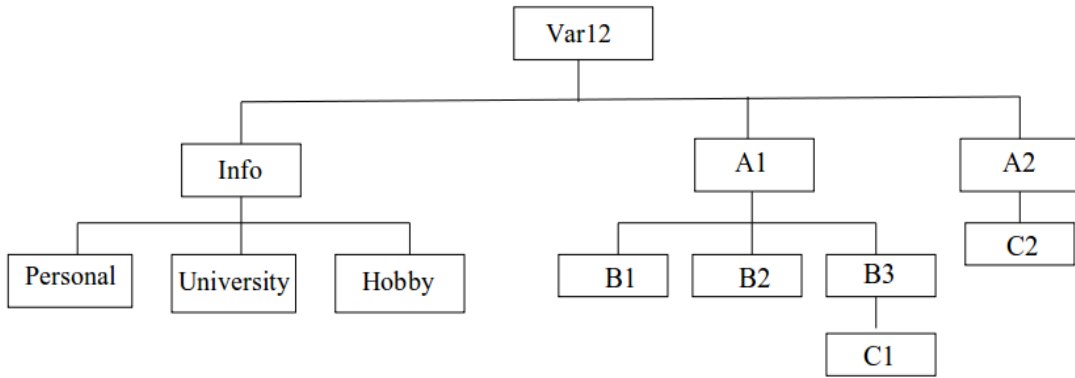


Рисунок 12 – Вариант 12

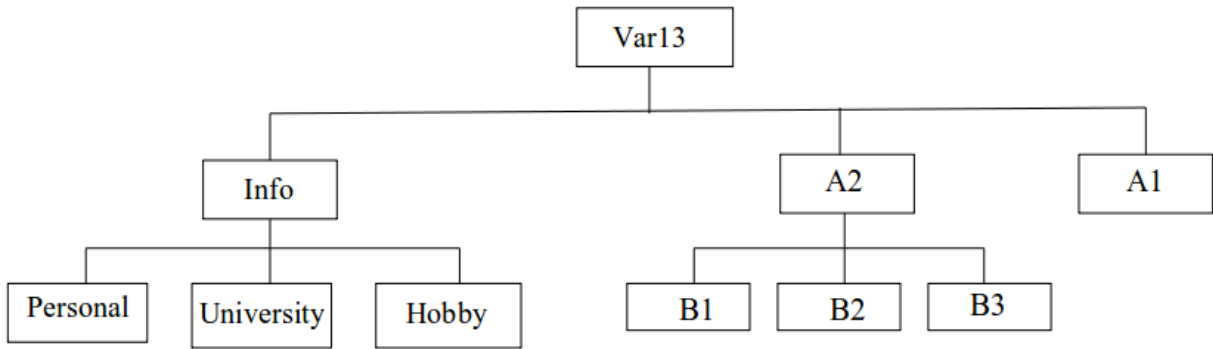


Рисунок 13 – Вариант 13

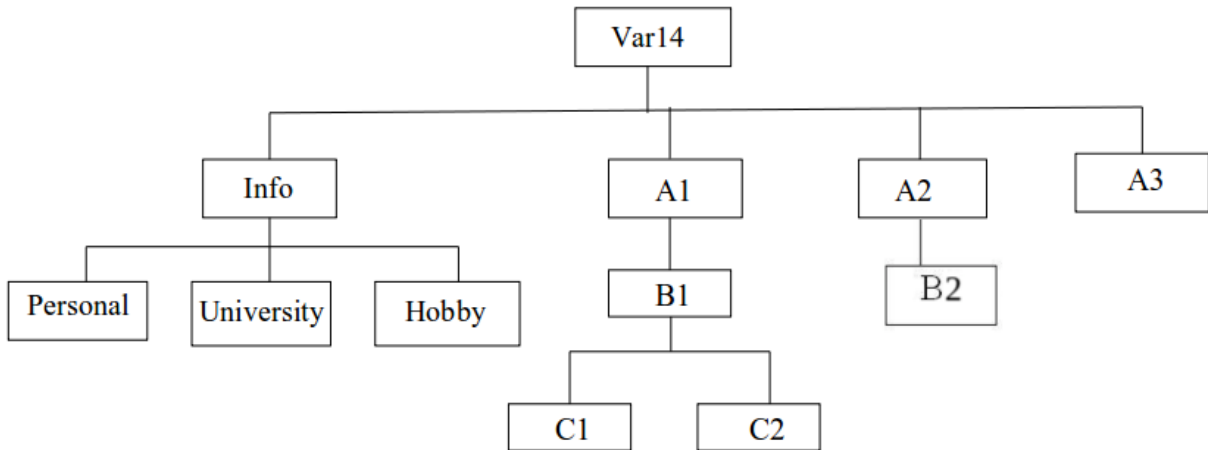


Рисунок 14 – Вариант 14

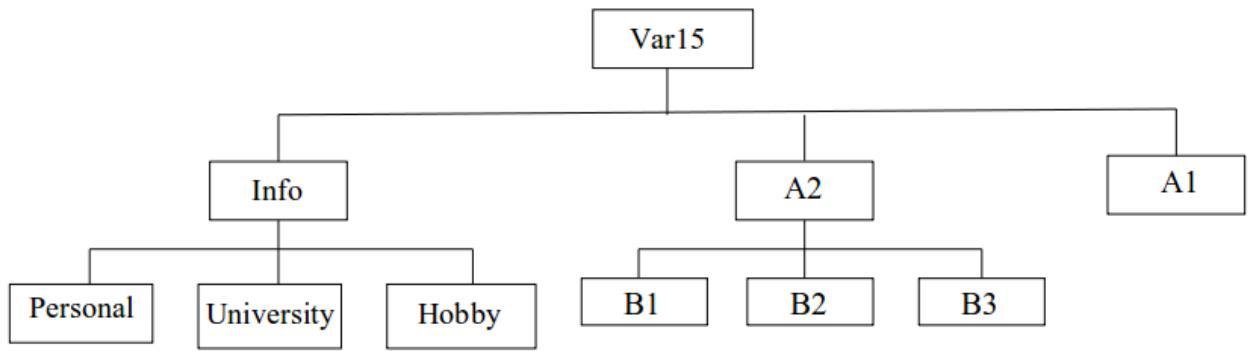


Рисунок 15 – Вариант 15

## Лабораторная работа №4

Используя задание №2, выполните:

1. Для Info установите права доступа «только чтение».
2. Перейдите в каталог Var и выполните:
  - 1.1. Создайте каталог test и перейдите в него.
  - 1.2. Создайте файл file, определите его права доступа и запретите любые действия с файлом. Попробуйте записать строку «тест» в file. Затем попробуйте прочитать файл.
  - 1.3. Измените права доступа для file так, чтобы только текущий пользователь мог осуществлять в него запись. Запишите строку «test» в файл. Попробуйте прочитать содержимое.
  - 1.4. Разрешите группе текущего пользователя читать содержимое файла. Попробуйте прочитать содержимое.
  - 1.5. Разрешите текущему пользователю читать содержимое файла
  - 1.6. Попробуйте прочитать содержимое – должна появиться ранее добавленная туда строка «test».
  - 1.7. Создайте каталог dir. Создайте в этом каталоге файл new\_file, запишите в него текст «file 2». Просмотрите содержимое каталога.
  - 1.8. Измените права доступа на каталог dir так, чтобы пользователи не имели права на исполнение (x). Попробуйте просмотреть содержимое файла new\_file 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:
  - В-1. Установите права 441 каталогу Primer.
  - В-2. Установите файлу primer1.txt права 454, удалите его.
  - В-3. Установите каталогу Primer и его содержимому права 344.
  - В-4. Установите всем файлам права 234.
  - В-5. Установите права 455 файлу primer2.txt.
  - В-6. Установите права 543 каталогу Primer.
  - В-7. Установите каталогу Primer права 552.
  - В-8. Установите каталогу Primer права 447.
  - В-9. Установите каталогу Primer права 444.
  - В-10. Установите обоим файлам права 333
  - В-11. Установите файлу primer1.txt права 411.
  - В-12. Установите созданным файлам права 300.

В-13. Установите каталогу Primer и его содержанию права 411.

В-14. Установите каталогу Primer права 432.

В-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

Изучить назначение и ключи команды `ln`. Создайте файл с описанием команд `dpkg`, `apt`, `apt-get`, `ln`. Создайте жесткую ссылку на него. Просмотреть содержимое файла, используя ссылку. Создайте символическую ссылку на исходный файл. С помощью команды `ls` просмотрите дескрипторы файла и ссылок.



## Лабораторная работа №7

В-1. Пользуясь командами терминала, установите пакет `ttf-mscorefonts-installer`. Каково назначение этого пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-2. Пользуясь командами терминала, установите пакет `dia`. Каково назначение этого пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-3. Пользуясь командами терминала, установите пакет `wxmaxima`. Каково назначение этого пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-4. Пользуясь командами терминала, установите пакет `g++`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-5. Пользуясь командами терминала установите пакет `geany`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-6. Пользуясь командами терминала, установите пакет `gfortran`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-7. Скачайте пакет `scilab` и установите его, пользуясь командами терминала. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-8. Пользуясь командами терминала, установите пакет `frs`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-9. Пользуясь командами терминала, установите пакет `Gimp`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-10. Пользуясь командами терминала установите пакет `Transmission`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-11. Пользуясь командами терминала, установите пакет `octave`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-12. Пользуясь командами терминала, установите пакет `Gradio`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-13. Пользуясь командами терминала, установите пакет `lazarus`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-14. Пользуясь командами терминала, установите пакет `g++`. Каково назначение пакета? Создайте текстовый файл Lab6 с ответом на вопрос.

В-15. Пользуясь командами терминала, установите пакет `vlc`. Каково назначение пакета? Создайте текстовый файл Lab6 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.

## Оглавление

Глава 1. Знакомство с ОС Linux.....	2
Глава 2. Файловая система ОС Linux.....	3
Глава 3. Терминал Linux и основные команды.....	4
Команды получения справочной информации.....	4
Команда info.....	5
Команда help.....	5
Команды для работы с файлами.....	5
Перенаправление информации в файл.....	7
Права доступа в ОС Linux.....	9
Глава 4. Создание нового пользователя. Изменение прав доступа.....	10
Основные команды для работы с правами доступа.....	11
Глава 5. Пакеты в ОС Linux.....	12
Установка приложений.....	13
Обновление базы данных пакетов.....	13
Глава 6. Настройка начального загрузчика.....	14
Глава 7. Жёсткие и символьные ссылки.....	15
Понятие ссылки.....	15
Жёсткая ссылка.....	15
Символьные ссылки.....	16
Глава 8. Установка Linux Mint.....	17
Создание образа ОС Linux.....	17
Запись образа на USB.....	18
Установка Linux Mint рядом с Windows (Пример Windows 10).....	18
Загрузка LiveCD.....	19
Установка Linux Mint.....	20
Глава 9. Литература.....	25
Глава 10. Приложение. Лабораторные работы «Основы работы в ОС Linux».....	26
Лабораторная работа №1.....	27
Лабораторная работа №2.....	28
Лабораторная работа №3.....	29
Лабораторная работа №5.....	37
Лабораторная работа №6.....	38
Лабораторная работа №7.....	39
Лабораторная работа №8.....	40