

Лабораторная работа №5

Анализ файловой системы Linux.

Старовойтов Егор Сергеевич

Содержание

Цель работы	4
Задание	4
Теоретическое введение	5
Формат команды.....	5
Команда ls.....	7
Команда mkdir	8
Команда rm	9
Команды для работы с файлами и каталогами	10
touch	10
cat	10
less.....	10
head.....	10
tail.....	10
Копирование файлов и каталогов	10
cp.....	10
Перемещение и переименование файлов и каталогов.....	11
Права доступа	12
Изменение прав доступа.....	12
Анализ файловой системы	14
mount.....	14
df	15
Выполнение лабораторной работы.....	15
Шаг 1 - Выполните все примеры, приведённые в первой части описания лабораторной работы.	15
Пример 1.....	15
Пример 2.....	16
Пример 3.....	16

Пример 4.....	16
Пример 5.....	17
Пример 6.....	17
Пример 7.....	17
Пример 8.....	18
Пример 9.....	18
Пример 10	18
Пример 11	18
Пример 12	18
Пример 13	19
Пример 14	19
Пример 15	19
Пример 16	20
Пример 17	20
Пример 18	21
Шаг 2 - Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:.....	21
Шаг 2.1	21
Шаг 2.2	22
Шаг 2.3	22
Шаг 2.4	23
Шаг 2.5	23
Шаг 2.6	23
Шаг 2.7	24
Шаг 2.8	24
Шаг 3 - Определите опции команды <code>chmod</code> , необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет.....	25
Шаг 3.1 - <code>drwxr-r- ... australia</code>	25
Шаг 3.2 - <code>drwx-x-x ... play</code>	25
Шаг 3.3 - <code>-r-xr-r- ... my_os</code>	25
Шаг 3.4 - <code>-rw-rw-r- ... feathers</code>	25
Шаг 4 - Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:.....	25
Шаг 4.1	25

Шаг 4.2	26
Шаг 4.3	26
Шаг 4.4	26
Шаг 4.5	26
Шаг 4.6	27
Шаг 4.7	27
Шаг 4.8	27
Шаг 4.9	27
Шаг 4.10	27
Шаг 4.11	28
Шаг 4.12	28
Шаг 5 - Прочитайте man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте,приведя примеры.	28
mount.....	28
fsck	29
mkfs	30
kill	31
Вывод.....	32
Контрольные вопросы	33
1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.....	33
2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.....	33
3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?.....	34
4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?	34
5. Как создаётся файловая система?	35
6. Дайте характеристику командам для просмотра текстовых файлов.	35
7. Приведите основные возможности команды cp в Linux.	35
8. Приведите основные возможности команды mv в Linux.	35
9. Что такое права доступа? Как они могут быть изменены?	35

Цель работы

Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

Задание

1. Выполните все примеры, приведённые в первой части описания лабораторной работы.
2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
 1. Скопируйте файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него.
 2. В домашнем каталоге создайте директорию `~/ski.places`.
 3. Переместите файл `equipment` в каталог `~/ski.places`.
 4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
 6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
 7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 8. Создайте и переместите каталог `~/newdir` в каталог `~/ski.places` и назовите его `plans`.
3. Определите опции команды **chmod**, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:
 1. `drwxr-r- ... australia`
 2. `drwx-x-x ... play`
 3. `-r-xr-r- ... my_os`
 4. `-rw-rw-r- ... feathers`

При необходимости создайте нужные файлы.

4. Прделайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 1. Просмотрите содержимое файла `/etc/password`.
 2. Скопируйте файл `~/feathers` в файл `~/file.old`.
 3. Переместите файл `~/file.old` в каталог `~/play`.

4. Скопируйте каталог `~/play` в каталог `~/fun`.
5. Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`.
6. Лишите владельца файла `~/feathers` права на чтение. 7. Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`?
7. Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`?
8. Дайте владельцу файла `~/feathers` право на чтение.
9. Лишите владельца каталога `~/play` права на выполнение.
10. Перейдите в каталог `~/play`. Что произошло?
11. Дайте владельцу каталога `~/play` право на выполнение.
5. Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.

Теоретическое введение

Формат команды

Командой в операционной системе называется записанный по специальным правилам текст (возможно с аргументами), представляющий собой указание на выполнение какой-либо функций (или действий) в операционной системе. Обычно первым словом идёт имя команды, остальной текст — аргументы или опции, конкретизирующие действие.

Общий формат команд можно представить следующим образом:

```
## Команда man
Команда ``man`` используется для просмотра (оперативная помощь) в
диалоговом режиме руководства (manual) по основным командам операционной
системы
типа Linux.
Формат команды: ``man <команда>``
```

Пример (вывод информации о команде `man`): ```man man```.

Для управления просмотром результата выполнения команды `man` можно использовать

следующие клавиши:

- Space — перемещение по документу на одну страницу вперёд;
- Enter — перемещение по документу на одну строку вперёд;
- q — выход из режима просмотра описания.

```
## Команда cd.
```

Команда `cd` используется для перемещения по файловой системе операционной системы типа Linux.

Замечание 1. Файловая система ОС типа Linux – иерархическая система каталогов, подкаталогов и файлов, которые обычно организованы и сгруппированы по функциональному признаку.

Самый верхний каталог в иерархии называется корневым и обозначается символом /. Корневой каталог содержит системные файлы и другие каталоги.

Формат команды:

```
```cd [путь_к_каталогу]```
```

Для перехода в домашний каталог пользователя следует использовать команду ````cd```` без параметров или ````cd ~````.

Например, команда

```
```cd /afs/dk.sci.pfu.edu.ru/home```
```

позволяет перейти в каталог `/afs/dk.sci.pfu.edu.ru/home` (если такой существует),

а для того, чтобы подняться выше на одну директорию, следует использовать:

```
```cd ..```
```

Подробнее об опциях команды `**cd**` смотри в справке с помощью команды `man`:

```
```man cd```
```

Команда `pwd`

Для определения абсолютного пути к текущему каталогу используется команда `pwd` (`print working directory`).

Пример (абсолютное имя текущего каталога пользователя `dharm`):

```
```bash
```

```
pwd
```

результат:

```
1 /afs/dk.sci.pfu.edu.ru/home/d/h/dharma
```

Сокращения имён файлов. В работе с командами, в качестве аргументов которых выступает путь к какому-либо каталогу или файлу, можно использовать сокращённую запись пути. Символы сокращения приведены в табл. 4.1.

Таблица 4.1 - ~ Домашний каталог - . Текущий каталог - .. Родительский каталог

Например, в команде `cd` для перемещения по файловой системе сокращённую запись пути можно использовать следующим образом (команды чередуются с выводом результата выполнения команды `pwd`):

```
pwd
```

```
/afs/dk.sci.pfu.edu.ru/home/d/h/dharma
```

```
cd ..
pwd
```

```
/afs/dk.sci.pfu.edu.ru/home/d/h
```

```
cd ../..
pwd
```

```
/afs/dk.sci.pfu.edu.ru/home
```

```
cd ~/work
pwd
```

```
/afs/dk.sci.pfu.edu.ru/home/d/h/dharma/work
```

## Команда ls

Команда ls используется для просмотра содержимого каталога.

Формат команды: ls [-опции] [путь]

Пример:

```
cd
cd ..
pwd
```

```
/afs/dk.sci.pfu.edu.ru/home/d/h
```

```
ls
```

```
dharma
```

Некоторые файлы в операционной системе скрыты от просмотра и обычно используются для настройки рабочей среды. Имена таких файлов начинаются с точки. Для того, чтобы отобразить имена скрытых файлов, необходимо использовать команду **ls** с опцией **a**: **ls -a**.

Можно также получить информацию о типах файлов (каталог, исполняемый файл, ссылка), для чего используется опция **F**. При использовании этой опции в поле имени выводится символ, который определяет тип файла (см. табл. 4.2) Таблица 4.2 - Каталог / - Исполняемый файл \* - Ссылка @

Чтобы вывести на экран подробную информацию о файлах и каталогах, необходимо использовать опцию **l**. При этом о каждом файле и каталоге будет выведена следующая информация: - тип файла, - право доступа, - число ссылок, - владелец, - размер, - дата последней ревизии, - имя файла или каталога.

Пример:

```
cd /
ls
```

Результат:

```
bin boot dev etc home lib media mnt
opt proc root sbin sys tmp usr var
```

В этом же каталоге команда `ls -alF` даст примерно следующий результат:

```
drwxr-xr-x 21 root root 4096 Jan. 17 09:00 ./
drwxr-xr-x 21 root root 4096 Jan. 17 09:00 ../
drwxr-xr-x 2 root root 4096 Jan. 18 15:57 bin/
drwxr-xr-x 2 root root 4096 Apr. 14 2008 boot/
drwxr-xr-x 20 root root 14120 Feb. 17 10:48 dev/
drwxr-xr-x 170 root root 12288 Feb. 17 09:19 etc/
drwxr-xr-x 6 root root 4096 Aug. 5 2009 home/
lrwxrwxrwx 1 root root 5 Jan. 12 22:01 lib -> lib64/
drwxr-xr-x 8 root root 4096 Jan. 30 21:41 media/
drwxr-xr-x 5 root root 4096 Jan. 17 2010 mnt/
drwxr-xr-x 25 root root 4096 Jan. 16 09:55 opt/
dr-xr-xr-x 163 root root 0 Feb. 17 13:17 proc/
drwxr-xr-x 31 root root 4096 Feb. 15 23:57 root/
drwxr-xr-x 2 root root 12288 Jan. 18 15:57 sbin/
drwxr-xr-x 12 root root 0 Feb. 17 13:17 sys/
drwxrwxrwt 12 root root 500 Feb. 17 16:35 tmp/
drwxr-xr-x 22 root root 4096 Jan. 18 09:26 usr/
drwxr-xr-x 17 root root 4096 Jan. 14 17:38 var/
```

## Команда `mkdir`

Команда `mkdir` используется для создания каталогов.

Формат команды: `mkdir имя_каталога1 [имя_каталога2...]`

Пример создания каталога в текущем каталоге:

```
cd
pwd
```

```
/afs/dk.sci.pfu.edu.ru/home/d/h/dharma
```

```
ls
```

```
Desktop public tmp
GNUstep public_html work
```

```
mkdir abc
ls
```



```
abc GNUstep public_html work
Desktop public tmp
```

Замечание 2. Для того чтобы создать каталог в определённом месте файловой системы, должны быть правильно установлены права доступа.

Можно создать также подкаталог в существующем подкаталоге:

```
mkdir parentdir
mkdir parentdir/dir
```

При задании нескольких аргументов создаётся несколько каталогов:

```
cd parentdir
mkdir dir1 dir2 dir3
```

Можно использовать группировку: `mkdir parentdir/{dir1,dir2,dir3}`

Если же требуется создать подкаталог в каталоге, отличном от текущего, то путь к нему требуется указать в явном виде:

```
mkdir ../dir1/dir2
или
mkdir ~/dir1/dir2
```

Интересны следующие опции: - **--mode** (или **-m**) — установка атрибутов доступа; - **--parents** (или **-p**) — создание каталога вместе с родительскими по отношению к нему каталогами.

Атрибуты задаются в численной или символьной нотации: `mkdir --mode=777 dir` или `mkdir -m a+rwX dir`

Опция **--parents** (краткая форма **-p**) позволяет создавать иерархическую цепочку подкаталогов, создавая все промежуточные каталоги: `mkdir -p ~/dir1/dir2/dir3`

## Команда **rm**

Команда **rm** используется для удаления файлов и/или каталогов.

Формат команды: `rm [-опции] [файл]`

Если требуется, чтобы выдавался запрос подтверждения на удаление файла, то необходимо использовать опцию **-i**.

Чтобы удалить каталог, содержащий файлы, нужно использовать опцию **-r**. Без указания этой опции команда не будет выполняться.

Пример:

```
cd
mkdir abs
rm abc
```

`rm: abc is a directory`

`rm -r abc`

Если каталог пуст, то можно воспользоваться командой **rmdir**. Если удаляемый каталог содержит файлы, то команда не будет выполнена — нужно использовать `rm -r` имя\_каталога.

## Команды для работы с файлами и каталогами

### touch

Для создания текстового файла можно использовать команду **touch**. Формат команды: `touch` имя-файла

### cat

Для просмотра файлов небольшого размера можно использовать команду **cat**.

Формат команды: `cat` имя-файла

### less

Для просмотра файлов постранично удобнее использовать команду **less**.

Формат команды: `less` имя-файла

Следующие клавиши используются для управления процессом просмотра: - Space — переход к следующей странице, - ENTER — сдвиг вперёд на одну строку, - b — возврат на предыдущую страницу, - h — обращение за подсказкой, - q — выход из режима просмотра файла.

### head

Команда **head** выводит по умолчанию первые 10 строк файла.

Формат команды: `head [-n]` имя-файла, где `n` — количество выводимых строк.

### tail

Команда **tail** выводит по умолчанию 10 последних строк файла.

Формат команды: `tail [-n]` имя-файла, где `n` — количество выводимых строк.

## Копирование файлов и каталогов

### cp

Команда **cp** используется для копирования файлов и каталогов.

Формат команды: `cp [-опции]` исходный\_файл целевой\_файл

Примеры:

1. Копирование файла в текущем каталоге. Скопировать файл ~/abc1 в файл april и в файл may:

```
cd
touch abc1
cp abc1 april
cp abc1 may
```

2. Копирование нескольких файлов в каталог. Скопировать файлы april и may в каталог monthly:

```
mkdir monthly
cp april may monthly
```

3. Копирование файлов в произвольном каталоге. Скопировать файл monthly/may в файл с именем june:

```
cp monthly/may monthly/june
ls monthly
```

Опция **i** в команде **cp** выведет на экран запрос подтверждения о перезаписи файла. Для рекурсивного копирования каталогов, содержащих файлы, используется команда **cp** с опцией **r**.

Примеры:

1. Копирование каталогов в текущем каталоге. Скопировать каталог monthly в каталог monthly.00:

```
mkdir monthly.00
cp -r monthly monthly.00
```

2. Копирование каталогов в произвольном каталоге. Скопировать каталог monthly.00 в каталог /tmp

```
cp -r monthly.00 /tmp
```

## Перемещение и переименование файлов и каталогов

Команды **mv** и **mvdir** предназначены для перемещения и переименования файлов и каталогов.

Формат команды **mv**: **mv** [-опции] старый\_файл новый\_файл

Примеры: 1. Переименование файлов в текущем каталоге. Изменить название файла april на july в домашнем каталоге:

```
cd
mv april july
```

2. Перемещение файлов в другой каталог. Переместить файл july в каталог monthly.00:

```
mv july monthly.00
ls monthly.00
```

Результат:

```
april july june may
```

Если необходим запрос подтверждения о перезаписи файла, то нужно использовать опцию `i`.

3. Переименование каталогов в текущем каталоге. Переименовать каталог `monthly.00` в `monthly.01`

```
mv monthly.00 monthly.01
```

4. Перемещение каталога в другой каталог. Переместить каталог `monthly.01` в каталог `reports`:

```
mkdir reports
```

```
mv monthly.01 reports
```

5. Переименование каталога, не являющегося текущим. Переименовать каталог `reports/monthly.01` в `reports/monthly`:

```
mv reports/monthly.01 reports/monthly
```

## Права доступа

Каждый файл или каталог имеет права доступа. В сведениях о файле или каталоге указываются: - тип файла (символ `-`) обозначает файл, а символ `(d)` — каталог); - права для владельца файла (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует); - права для членов группы (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует); - права для всех остальных (`r` — разрешено чтение, `w` — разрешена запись, `x` — разрешено выполнение, `-` — право доступа отсутствует).

Примеры:

1. Для файла (крайнее левое поле имеет значение `-`) владелец файла имеет право на чтение и запись (`rw-`), группа, в которую входит владелец файла, может читать файл (`r-`), все остальные могут читать файл (`r-`): `-rw-r--r--`
2. Только владелец файла имеет право на чтение, изменение и выполнение файла: `-rwx-----`.
3. Владелец каталога (крайнее левое поле имеет значение `d`) имеет право на просмотр, изменение и доступа в каталог, члены группы могут входить и просматривать его, все остальные — только входить в каталог: `drwxr-x--x`.

## Изменение прав доступа

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой **chmod**. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

Формат команды: `chmod режим имя_файла`

Режим (в формате команды) имеет следующие компоненты структуры и способ записи: - = установить право - - лишить права - + дать право - r чтение - w запись - u (user) владелец файла - g (group) группа, к которой принадлежит владелец файла - o (others) все остальные

В работе с правами доступа можно использовать их цифровую запись (восьмеричное значение) вместо символьной. Формы записи прав доступа

Двоичная	Восьмеричная	Символьная
111	7	gwx
110	6	gw
101	5	r-x
100	4	r-
011	3	-wx
010	2	-w
001	1	-x
000	0	—

Примеры:

1. Требуется создать файл ~/may с правом выполнения для владельца:

```
cd
touch may
ls -l may
chmod u+x may
ls -l may
```

2. Требуется лишить владельца файла ~/may права на выполнение:

```
chmod u-x may
ls -l may
```

3. Требуется создать каталог monthly с запретом на чтение для членов группы и всех остальных пользователей:

```
cd
mkdir monthly
chmod g-r, o-r monthly
```

4. Требуется создать файл ~/abc1 с правом записи для членов группы:

```
1 cd
2 touch abc1
3 chmod g+w abc1
```

## Анализ файловой системы

Файловая система в Linux состоит из файлов и каталогов. Каждому физическому носителю соответствует своя файловая система.

Существует несколько типов файловых систем. Перечислим наиболее часто встречающиеся типы: - ext2fs (second extended filesystem); - ext3fs (third extended file system); - ext4 (fourth extended file system); - ReiserFS; - xfs; - fat (file allocation table); - ntfs (new technology file system).

### mount

Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой **mount** без параметров. В результате её применения можно получить примерно следующее:

```
mount
```

```
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec)
udev on /dev type tmpfs (rw,nosuid)
devpts on /dev/pts type devpts (rw,nosuid,noexec)
/dev/sda1 on /mnt/a type ext3 (rw,noatime)
/dev/sdb2 on /mnt/docs type reiserfs (rw,noatime)
shm on /dev/shm type tmpfs (rw,noexec,nosuid,nodev)
usbfs on /proc/bus/usb type usbfs
(rw,noexec,nosuid,devmode=0664,devgid=85)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc
(rw,noexec,nosuid,nodev)
nfsd on /proc/fs/nfs type nfsd (rw,noexec,nosuid,nodev)
```

В данном случае указаны имена устройств, названия соответствующих им точек монтирования (путь), тип файловой системы и параметрами монтирования.

В контексте команды **mount** устройство — специальный файл устройства, с помощью которого операционная система получает доступ к аппаратному устройству.

Файлы устройств обычно располагаются в каталоге `/dev`, имеют сокращённые имена (например, `sdaN`, `sdbN` или `hdaN`, `hdbN`, где `N` — порядковый номер устройства, `sd` — устройства SCSI, `hd` — устройства MFM/IDE).

Точка монтирования — каталог (путь к каталогу), к которому присоединяются файлы устройств.

Другой способ определения смонтированных в операционной системе файловых систем — просмотр файла `/etc/fstab`. Сделать это можно например с помощью команды

```
cat /etc/fstab
```

```
/dev/hda1 / ext2 defaults 1 1
/dev/hda5 /home ext2 defaults 1 2
/dev/hda6 swap swap defaults 0 0
/dev/hdc /mnt/cdrom auto umask=0,user,noauto,ro,exec,users 0 0
```

```
none /mnt/floppy supermount dev=/dev/fd0,fs=ext2:vfat,--,
sync,umask=0 0 0
none /proc proc defaults 0 0
none /dev/pts devpts mode=0622 0 0
```

В каждой строке этого файла указано: - имя устройство; - точка монтирования; - тип файловой системы; - опции монтирования; - специальные флаги для утилиты dump; - порядок проверки целостности файловой системы с помощью утилиты fsck.

## df

Для определения объёма свободного пространства на файловой системе можно воспользоваться командой df, которая выведет на экран список всех файловых систем в соответствии с именами устройств, с указанием размера и точки монтирования.

Например:

```
df
```

```
Filesystem 1024-blocks Used Available Capacity Mounted on
/dev/hda3 297635 169499 112764 60% /
```

С помощью команды fsck можно проверить (а в ряде случаев восстановить) целостность файловой системы:

Формат команды: fsck имя\_устройства

Пример:

```
fsck /dev/sda1
```

## Выполнение лабораторной работы

**Шаг 1 - Выполните все примеры, приведённые в первой части описания лабораторной работы.**

### Пример 1

Копирование файла в текущем каталоге. Скопировать файл ~/abc1 в файл april и в файл may:

```
liveuser@localhost-live:~
[liveuser@localhost-live ~]$ cd
[liveuser@localhost-live ~]$ touch abc1
[liveuser@localhost-live ~]$ cp abc1 april
[liveuser@localhost-live ~]$ cp abc1 may
[liveuser@localhost-live ~]$ ls
abc1 Desktop Downloads Music Public Videos
april Documents may Pictures Templates
[liveuser@localhost-live ~]$
```

Пример 1

### Пример 2

Копирование нескольких файлов в каталог. Скопировать файлы april и may в каталог monthly:

```
[liveuser@localhost-live ~]$ mkdir montly
[liveuser@localhost-live ~]$ cp april may montly
[liveuser@localhost-live ~]$ ls
abc1 Desktop Downloads montly Pictures Templates
april Documents may Music Public Videos
[liveuser@localhost-live ~]$ cd montly
[liveuser@localhost-live montly]$ ls
april may
```

Пример 2

### Пример 3

Копирование файлов в произвольном каталоге. Скопировать файл monthly/may в файл с именем june:

```
[liveuser@localhost-live ~]$ cp montly/may montly/june
[liveuser@localhost-live ~]$ cd montly
[liveuser@localhost-live montly]$ ls
april june may
```

Пример 3

### Пример 4

Копирование каталогов в текущем каталоге. Скопировать каталог monthly в каталог monthly.00:

```
[liveuser@localhost-live ~]$ mkdir montly.00
[liveuser@localhost-live ~]$ cp -r montly montly.00
[liveuser@localhost-live ~]$ ls
abc1 Desktop Downloads montly Music Public Videos
april Documents may montly.00 Pictures Templates
```

Пример 4



### Пример 5

Копирование каталогов в произвольном каталоге. Скопировать каталог monthly.00 в каталог /tmp

```
[liveuser@localhost-live ~]$ cp -r montly.00 /tmp
[liveuser@localhost-live ~]$ tree
.
├── abc1
├── april
├── Desktop
├── Documents
├── Downloads
├── may
├── montly
│ ├── april
│ ├── june
│ └── may
├── montly.00
│ └── montly
│ ├── april
│ ├── june
│ └── may
├── Music
├── Pictures
├── Public
├── Templates
└── Videos

11 directories, 9 files
[liveuser@localhost-live ~]$ cd /tmp
[liveuser@localhost-live tmp]$ ls
montly.00
systemd-private-e481671f4ead42138f9f4ca4d5d44368-chrond.service-mTrebh
systemd-private-e481671f4ead42138f9f4ca4d5d44368-colord.service-NbYuwG
```

Пример 5

### Пример 6

Переименование файлов в текущем каталоге. Изменить название файла april на july в домашнем каталоге:

```
[liveuser@localhost-live tmp]$ cd
[liveuser@localhost-live ~]$ mv april july
[liveuser@localhost-live ~]$ ls
abc1 Documents july montly Music Public Videos
Desktop Downloads may montly.00 Pictures Templates
[liveuser@localhost-live ~]$
```

Пример 6

### Пример 7

Перемещение файлов в другой каталог. Переместить файл july в каталог monthly.00:

```
[liveuser@localhost-live ~]$ mv july montly.00
[liveuser@localhost-live ~]$ ls montly.00
july montly
```

Пример 7

### Пример 8

Переименование каталогов в текущем каталоге. Переименовать каталог monthly.00 в monthly.01

```
[liveuser@localhost-live ~]$ mv montly.00 montly.01
[liveuser@localhost-live ~]$ ls
abcl Documents may montly.01 Pictures Templates
Desktop Downloads montly Music Public Videos
[liveuser@localhost-live ~]$
```

### Пример 8

### Пример 9

Перемещение каталога в другой каталог. Переместить каталог monthly.01 в каталог reports:

```
[liveuser@localhost-live ~]$ mkdir reports
[liveuser@localhost-live ~]$ mv montly.01 reports
[liveuser@localhost-live ~]$ ls
abcl Documents may Music Public Templates
Desktop Downloads montly Pictures reports Videos
[liveuser@localhost-live ~]$ ls reports
montly.01
[liveuser@localhost-live ~]$
```

### Пример 9

### Пример 10

Переименование каталога, не являющегося текущим. Переименовать каталог reports/monthly.01 в reports/monthly:

```
[liveuser@localhost-live ~]$ mv reports/monthly.01 reports/monthly
[liveuser@localhost-live ~]$ ls reports
montly
[liveuser@localhost-live ~]$
```

### Пример 10

### Пример 11

Требуется создать файл ~/may с правом выполнения для владельца:

```
[liveuser@localhost-live ~]$ cd
[liveuser@localhost-live ~]$ touch may
[liveuser@localhost-live ~]$ ls -l may
-rw-rw-r--. 1 liveuser liveuser 0 May 2 09:38 may
[liveuser@localhost-live ~]$ chmod u+x may
[liveuser@localhost-live ~]$ ls -l may
-rwxrw-r--. 1 liveuser liveuser 0 May 2 09:38 may
[liveuser@localhost-live ~]$
```

### Пример 11

### Пример 12

Требуется лишить владельца файла ~/may права на выполнение:

```
[liveuser@localhost-live ~]$ chmod u-x may
[liveuser@localhost-live ~]$ ls -l may
-rw-rw-r--. 1 liveuser liveuser 0 May 2 09:38 may
[liveuser@localhost-live ~]$
```

*Пример 12*

### Пример 13

Требуется создать каталог `monthly` с запретом на чтение для членов группы и всех остальных пользователей:

```
[liveuser@localhost-live ~]$ mkdir monthly
[liveuser@localhost-live ~]$ chmod g-r, o-r monthly
chmod: invalid mode: 'g-r,'
Try 'chmod --help' for more information.
```

*Пример 13*

### Пример 14

Требуется создать файл `~/abc1` с правом записи для членов группы:

```
[liveuser@localhost-live ~]$ cd
[liveuser@localhost-live ~]$ touch abc1
[liveuser@localhost-live ~]$ chmod g+w abc1
[liveuser@localhost-live ~]$ ls -l abc1
-rw-rw-r--. 1 liveuser liveuser 0 May 2 09:45 abc1
[liveuser@localhost-live ~]$
```

*Пример 14*

### Пример 15

Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой `mount` без параметров. В результате её применения можно получить примерно следующее

```
[liveuser@localhost-live ~]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=1970628k,nr_inodes=492657,mode=755,inode64)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,seclabel,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,seclabel,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=801656k,nr_inodes=819200,mode=755,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,seclabel,nsdelegate,memory_recursiveprot)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime,seclabel)
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
/dev/sr0 on /run/initramfs/live type iso9660 (ro,relatime,nojoliet,check=s,map=n,blocksize=2048,iocharset=utf8)
/dev/mapper/live-rw on / type ext4 (rw,relatime,seclabel)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=30,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=18275)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
debugfs on /sys/kernel/debug type debugfs (rw,nosuid,nodev,noexec,relatime,seclabel)
tracefs on /sys/kernel/tracing type tracefs (rw,nosuid,nodev,noexec,relatime,seclabel)
fusectl on /sys/fs/fuse/connections type fusectl (rw,nosuid,nodev,noexec,relatime)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,seclabel,size=2004140k,nr_inodes=409600,inode64)
vartmp on /var/tmp type tmpfs (rw,relatime,seclabel,inode64)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=400824k,nr_inodes=100206,mode=700,uid=1000,gid=1000,inode64)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1000,group_id=1000)
/dev/sr1 on /run/media/liveuser/VBox_GAs_6.1.34 type iso9660 (ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,iocharset=utf8,uhelper=udisks2)
[liveuser@localhost-live ~]$
```

### Пример 15

### Пример 16

Другой способ определения смонтированных в операционной системе файловых систем — просмотр файла `/etc/fstab`. Сделать это можно например с помощью команды `cat`:

```
[liveuser@localhost-live ~]$ cat /etc/fstab
vartmp /var/tmp tmpfs defaults 0 0
[liveuser@localhost-live ~]$
```

### Пример 16

### Пример 17

Для определения объёма свободного пространства на файловой системе можно воспользоваться командой `df`, которая выведет на экран список всех файловых систем в соответствии с именами устройств, с указанием размера и точки монтирования. Например:

```
[liveuser@localhost-live ~]$ df
Filesystem 1K-blocks Used Available Use% Mounted on
devtmpfs 1970628 0 1970628 0% /dev
tmpfs 2004136 0 2004136 0% /dev/shm
tmpfs 801656 9552 792104 2% /run
/dev/sr0 1962240 1962240 0 100% /run/initramfs/live
/dev/mapper/live-rw 7640588 5501212 2122992 73% /
tmpfs 2004140 52 2004088 1% /tmp
vartmp 2004136 0 2004136 0% /var/tmp
tmpfs 400824 136 400688 1% /run/user/1000
/dev/sr1 60096 60096 0 100% /run/media/liveuser/VBox_GAs_6.1.34
[liveuser@localhost-live ~]$
```

Пример 17

### Пример 18

С помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы:

```
[liveuser@localhost-live ~]$ fsck /dev/sda1
fsck from util-linux 2.37.2
e2fsck 1.46.3 (27-Jul-2021)
fsck.ext2: No such file or directory while trying to open /dev/sda1
Possibly non-existent device?
[liveuser@localhost-live ~]$
```

Пример 18

**Шаг 2 - Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:**

#### Шаг 2.1

Скопируйте файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Если файла `io.h` нет, то используйте любой другой файл в каталоге `/usr/include/sys/` вместо него.

Для копирования файла я воспользовался командой `cp`, после чего убедился в успешности операции с помощью команд `ls` и `cat`.

```
[liveuser@localhost-live ~]$ cp /usr/include/sys/io.h ~/equipment
[liveuser@localhost-live ~]$ ls
abcl Documents equipment monthly Music Public Templates
Desktop Downloads may montly Pictures reports Videos
[liveuser@localhost-live ~]$ cat equipment
/* Copyright (C) 1996-2021 Free Software Foundation, Inc.
 This file is part of the GNU C Library.

 The GNU C Library is free software; you can redistribute it and/or
 modify it under the terms of the GNU Lesser General Public
 License as published by the Free Software Foundation; either
 version 2.1 of the License, or (at your option) any later version.

 The GNU C Library is distributed in the hope that it will be useful,
 but WITHOUT ANY WARRANTY; without even the implied warranty of
 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 Lesser General Public License for more details.

 You should have received a copy of the GNU Lesser General Public
 License along with the GNU C Library; if not, see
 <https://www.gnu.org/licenses/>. */

#ifndef _SYS_IO_H
#define _SYS_IO_H 1

#include <features.h>

__BEGIN_DECLS

/* If TURN_ON is TRUE, request for permission to do direct i/o on the
 port numbers in the range [FROM, FROM+NUM-1]. Otherwise, turn I/O
 permission off for that range. This call requires root privileges.

 Portability note: not all Linux platforms support this call. Most
 platforms based on the PC I/O architecture probably will, however.
 E.g., Linux/Alpha for Alpha PCs supports this. */
extern int ioperm (unsigned long int __from, unsigned long int __num,
 int __turn_on) __THROW;

/* Set the I/O privilege level to LEVEL. If LEVEL>3, permission to
 access any I/O port is granted. This call requires root
```

## Задание 2.1

### Шаг 2.2

В домашнем каталоге создайте директорию ~/ski.plases.

Я создал нужную директорию с помощью команды mkdir и убедился в результате с помощью команды ls.

```
[liveuser@localhost-live ~]$ mkdir ski.plases
[liveuser@localhost-live ~]$ ls
abcl Documents equipment monthly Music Public ski.plases Videos
Desktop Downloads may montly Pictures reports Templates
[liveuser@localhost-live ~]$
```

## Задание 2.2

### Шаг 2.3

Переместите файл equipment в каталог ~/ski.plases.

Для перемещения файла я воспользовался командой `mv`, а затем проверил результат с помощью `ls`.

```
[liveuser@localhost-live ~]$ mv equipment ski.places
[liveuser@localhost-live ~]$ ls
abc1 Documents may montly Pictures reports Templates
Desktop Downloads monthly Music Public ski.places Videos
[liveuser@localhost-live ~]$ ls ski.places
equipment
[liveuser@localhost-live ~]$
```

### Задание 2.3

#### Шаг 2.4

Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.

Для переименования файла я воспользовался командой `mv`, а затем проверил результат с помощью `ls`.

```
[liveuser@localhost-live ~]$ mv ski.places/equipment ski.places/equiplist
[liveuser@localhost-live ~]$ ls ski.places
equiplist
[liveuser@localhost-live ~]$
```

### Задание 2.4

#### Шаг 2.5

Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.

С помощью команды `touch` я создал файл `abc1`, затем с помощью команды `cp` скопировал его в каталог `~/ski.places` под новым именем `equiplist2`. С помощью команды `ls` проверил результат.

```
[liveuser@localhost-live ~]$ touch abc1
[liveuser@localhost-live ~]$ cp abc1 ski.places/equiplist2
[liveuser@localhost-live ~]$ ls ski.places
equiplist equiplist2
[liveuser@localhost-live ~]$
```

### Задание 2.5

#### Шаг 2.6

Создайте каталог с именем `equipment` в каталоге `~/ski.places`.

С помощью команды `cd` я перешел в `~/ski.places`, где с помощью `mkdir` создал каталог `equipment`.

```
[liveuser@localhost-live ski.plases]$ mkdir equipment
[liveuser@localhost-live ski.plases]$ mv equiplist equipment
[liveuser@localhost-live ski.plases]$ mv equiplist2 equipment
[liveuser@localhost-live ski.plases]$ tree
.
└─ equipment
 ├── equiplist
 └── equiplist2

1 directory, 2 files
[liveuser@localhost-live ski.plases]$
```

## Задание 2.6

### Шаг 2.7

Переместите файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment

С данным заданием справляется двухкратный вызов команды mv, а спомощью tree легко увидеть результат.

```
[liveuser@localhost-live ski.plases]$ mkdir equipment
[liveuser@localhost-live ski.plases]$ mv equiplist equipment
[liveuser@localhost-live ski.plases]$ mv equiplist2 equipment
[liveuser@localhost-live ski.plases]$ tree
.
└─ equipment
 ├── equiplist
 └── equiplist2

1 directory, 2 files
[liveuser@localhost-live ski.plases]$
```

## Задание 2.7

### Шаг 2.8

8. Создайте и переместите каталог ~/newdir в каталог ~/ski.plases и назовите его plans.

С помощью команды mkdir я создал каталог newdir, после чего переместил его в ~/ski.plases с помощью команды mv. С помощью ls увидел результат.

```
[liveuser@localhost-live ~]$ mkdir newdir
[liveuser@localhost-live ~]$ mv newdir ski.plases/plans
[liveuser@localhost-live ~]$ ls
abc1 Documents may montly Pictures reports Templates
Desktop Downloads monthly Music Public ski.plases Videos
[liveuser@localhost-live ~]$ ls ski.plases
equipment plans
[liveuser@localhost-live ~]$
```

## Задание 2.8



**Шаг 3 - Определите опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет.**

На данном этапе я задавал права доступа к файлам в численном виде.

#### Шаг 3.1 - drwxr--r-- ... australia

```
[liveuser@localhost-live ~]$ touch australia
[liveuser@localhost-live ~]$ chmod 744 australia
[liveuser@localhost-live ~]$ ls -l australia
-rwxr--r--. 1 liveuser liveuser 0 May 2 10:23 australia
[liveuser@localhost-live ~]$
```

#### Задание 3.1

#### Шаг 3.2 - drwx--x--x ... play

```
[liveuser@localhost-live ~]$ touch play
[liveuser@localhost-live ~]$ chmod 711 play
[liveuser@localhost-live ~]$ ls -l play
-rwx--x--x. 1 liveuser liveuser 0 May 2 10:24 play
[liveuser@localhost-live ~]$
```

#### Задание 3.2

#### Шаг 3.3 - -r-xr--r-- ... my\_os

```
[liveuser@localhost-live ~]$ chmod 544 my_os
[liveuser@localhost-live ~]$ ls -l my_os
-r-xr--r--. 1 liveuser liveuser 0 May 2 10:25 my_os
[liveuser@localhost-live ~]$
```

#### Задание 3.3

#### Шаг 3.4 - -rw-rw-r-- ... feathers

```
[liveuser@localhost-live ~]$ chmod 664 feathers
[liveuser@localhost-live ~]$ ls -l feathers
-rw-rw-r--. 1 liveuser liveuser 0 May 2 10:21 feathers
[liveuser@localhost-live ~]$
```

#### Задание 3.4

**Шаг 4 - Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:**

#### Шаг 4.1

Просмотрите содержимое файла /etc/passwd.

Я ввел cat /etc/passwd, но искомого файла не оказалось.

```
[liveuser@localhost-live ~]$ cat /etc/passwd
cat: /etc/passwd: No such file or directory
[liveuser@localhost-live ~]$
```

#### Задание 4.1

## Шаг 4.2

Скопируйте файл ~/feathers в файл ~/file.old.

Я использовал команду `cp`.

```
[liveuser@localhost-live ~]$ cp feathers file.old
[liveuser@localhost-live ~]$ ls
abcl Documents file.old montly Pictures reports Videos
australia Downloads may Music play ski.plases
Desktop feathers monthly my_os Public Templates
[liveuser@localhost-live ~]$
```

### Задание 4.2

## Шаг 4.3

Скопируйте файл ~/file.old в каталог ~/play.

Я использовал команду `cp`.

```
[liveuser@localhost-live ~]$ cp file.old play/
```

### Задание 4.3

## Шаг 4.4

Скопируйте каталог ~/play в каталог ~/fun.

Я использовал команду `cp` с опцией `-r`.

```
[liveuser@localhost-live ~]$ cp -r ~/play ~/fun
[liveuser@localhost-live ~]$ ls
abcl Documents file.old monthly my_os Public Templates
australia Downloads fun montly Pictures reports Videos
Desktop feathers may Music play ski.plases
[liveuser@localhost-live ~]$
```

### Задание 4.4

## Шаг 4.5

Переместите каталог ~/fun в каталог ~/play и назовите его `games`.

Я использовал команду `mv`, результат проверил с помощью команды `tree`.

```
[liveuser@localhost-live ~]$ mv ~/fun ~/play/games
[liveuser@localhost-live ~]$ cd play
[liveuser@localhost-live play]$ tree
.
├── file.old
└── games
 └── file.old

1 directory, 2 files
[liveuser@localhost-live play]$
```

### Задание 4.5

## Шаг 4.6

Лишите владельца файла ~/feathers права на чтение.

Я использовал команду chmod с аргументом u-r.

```
[liveuser@localhost-live ~]$ chmod u-r feathers
```

### Задание 4.6

## Шаг 4.7

Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?

Я получил ошибку “cat: feathers: Permission denied”, что означает “отказано в доступе”.

```
[liveuser@localhost-live ~]$ chmod u-r feathers
[liveuser@localhost-live ~]$ cat feathers
cat: feathers: Permission denied
[liveuser@localhost-live ~]$
```

### Задание 4.7

## Шаг 4.8

Что произойдёт, если вы попытаетесь скопировать файл ~/feathers?

Я получил ошибку “cp: cannot open 'feathers' for reading: Permission denied”, - программа cp не смогла прочитать файл, так как у пользователя отсутствует на это право.

```
[liveuser@localhost-live ~]$ cp feathers feathers_copy
cp: cannot open 'feathers' for reading: Permission denied
[liveuser@localhost-live ~]$
```

### Задание 4.8

## Шаг 4.9

Дайте владельцу файла ~/feathers право на чтение.

Я использовал команду chmod с аргументом u+r. В изменении прав убедился с помощью ls -l.

```
[liveuser@localhost-live ~]$ chmod u+r feathers
[liveuser@localhost-live ~]$ ls -l feathers
-rw-rw-r--. 1 liveuser liveuser 0 May 2 10:21 feathers
[liveuser@localhost-live ~]$
```

### Задание 4.9

## Шаг 4.10

Лишите владельца каталога ~/play права на выполнение.

Я использовал команду `chmod` с аргументом `u-x`.

```
[liveuser@localhost-live ~]$ chmod u-x play
```

*Задание 4.10*

### Шаг 4.11

Перейдите в каталог `~/play`. Что произошло?

При попытке перейти в каталог мне было отказано в доступе.

```
[liveuser@localhost-live ~]$ cd play
bash: cd: play: Permission denied
[liveuser@localhost-live ~]$
```

*Задание 4.11*

### Шаг 4.12

Дайте владельцу каталога `~/play` право на выполнение.

Я использовал команду `chmod` с аргументом `u+x`.

```
[liveuser@localhost-live ~]$ chmod u+x play
```

*Задание 4.12*

**Шаг 5 - Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.**

### `mount`

Команда **`mount`** позволяет объединить несколько файловых систем в единое дерево каталогов. Для подмонтирования нового устройства нужно написать `mount` файл\_устройства пака\_назначения.

```
MOUNT(8) System Administration MOUNT(8)

NAME
 mount - mount a filesystem

SYNOPSIS
 mount [-h|-V]

 mount [-l] [-t fstype]

 mount -a [-fFnrsvw] [-t fstype] [-O optlist]

 mount [-fnrsvw] [-o options] device|mountpoint

 mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

 mount --bind|--rbind|--move olddir newdir

 mount --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable]
mountpoint

DESCRIPTION
 All files accessible in a Unix system are arranged in one big tree, the file
 hierarchy, rooted at /. These files can be spread out over several devices. The
 mount command serves to attach the filesystem found on some device to the big file
 tree. Conversely, the umount(8) command will detach it again. The filesystem is
 used to control how data is stored on the device or provided in a virtual way by
 network or other services.

 The standard form of the mount command is:

 mount -t type device dir

 This tells the kernel to attach the filesystem found on device (which is of type
 type) at the directory dir. The option -t type is optional. The mount command is
 usually able to detect a filesystem. The root permissions are necessary to mount a
 filesystem by default. See section "Non-superuser mounts" below for more details.
 The previous contents (if any) and owner and mode of dir become invisible, and as
 long as this filesystem remains mounted, the pathname dir refers to the root of the
 filesystem on device.

Manual page mount(8) line 1 (press h for help or q to quit)
```

*man mount*

## fsck

**fsck** - это утилита для проверки и восстановления файловых систем Linux. Обычно команда fsck автоматически запускается по возможности в параллельном режиме при загрузке ОС. По этой причине обычно нет необходимости запускать ее через командную строку.

```
FSCK(8) System Administration FSCK(8)

NAME
 fsck - check and repair a Linux filesystem

SYNOPSIS
 fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--]
 [fs-specific-options]

DESCRIPTION
 fsck is used to check and optionally repair one or more Linux filesystems.
 filesystem can be a device name (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g.,
 /, /usr, /home), or an filesystem label or UUID specifier (e.g.,
 UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the fsck
 program will try to handle filesystems on different physical disk drives in
 parallel to reduce the total amount of time needed to check all of them.

 If no filesystems are specified on the command line, and the -A option is not
 specified, fsck will default to checking filesystems in /etc/fstab serially. This
 is equivalent to the -As options.

 The exit status returned by fsck is the sum of the following conditions:

 0
 No errors

 1
 Filesystem errors corrected

 2
 System should be rebooted

 4
 Filesystem errors left uncorrected

 8
 Operational error

 16
 Usage or syntax error

Manual page fsck(8) line 1 (press h for help or q to quit)
```

*man fsck*

## mkfs

Команда **mkfs** используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жесткого диска. В качестве аргумента может выступать название устройства (например /dev/sda1) или точка монтирования (например /, /home).

```

MKFS(8) System Administration MKFS(8)

NAME
 mkfs - build a Linux filesystem

SYNOPSIS
 mkfs [options] [-t type] [fs-options] device [size]

DESCRIPTION
 This mkfs frontend is deprecated in favour of filesystem specific mkfs.<type>
 utils.

 mkfs is used to build a Linux filesystem on a device, usually a hard disk
 partition. The device argument is either the device name (e.g., /dev/hda1,
 /dev/sdb2), or a regular file that shall contain the filesystem. The size
 argument is the number of blocks to be used for the filesystem.

 The exit status returned by mkfs is 0 on success and 1 on failure.

 In actuality, mkfs is simply a front-end for the various filesystem builders
 (mkfs.fs<type>) available under Linux. The filesystem-specific builder is
 searched for via your PATH environment setting only. Please see the
 filesystem-specific builder manual pages for further details.

OPTIONS
 -t, --type type
 Specify the type of filesystem to be built. If not specified, the
 default filesystem type (currently ext2) is used.

 fs-options
 Filesystem-specific options to be passed to the real filesystem
 builder.

 -v, --verbose
 Produce verbose output, including all filesystem-specific commands
 that are executed. Specifying this option more than once inhibits
 execution of any filesystem-specific commands. This is really only
 useful for testing.

 -V, --version
 Display version information and exit. (Option -V will display
 version information only when it is the only parameter, otherwise it
 will work as

Manual page mkfs(8) line 1 (press h for help or q to quit)

```

*man mkfs*

## kill

Команда **kill** позволяет отправить сигнал процессу, принимая на вход его PID идентификатор. Например, можно принудительно завершить процесс с PID=2000

набрав `kill -KILL 2000`

```
KILL(1) User Commands KILL(1)

NAME
 kill - terminate a process

SYNOPSIS
 kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--]
 pid|name...

 kill -l [number] | -L

DESCRIPTION
 The command kill sends the specified signal to the specified processes or process
 groups.

 If no signal is specified, the TERM signal is sent. The default action for this
 signal is to terminate the process. This signal should be used in preference to the
 KILL signal (number 9), since a process may install a handler for the TERM signal
 in order to perform clean-up steps before terminating in an orderly fashion. If a
 process does not terminate after a TERM signal has been sent, then the KILL signal
 may be used; be aware that the latter signal cannot be caught, and so does not give
 the target process the opportunity to perform any clean-up before terminating.

 Most modern shells have a builtin kill command, with a usage rather similar to that
 of the command described here. The --all, --pid, and --queue options, and the
 possibility to specify processes by command name, are local extensions.

 If signal is 0, then no actual signal is sent, but error checking is still
 performed.

ARGUMENTS
 The list of processes to be signaled can be a mixture of names and PIDs.

 pid
 Each pid can be expressed in one of the following ways:

 n
 where n is larger than 0. The process with PID n is signaled.

 0
 Manual page kill(1) line 1 (press h for help or q to quit)
```

## Вывод

Я познакомился с файловой системой Linux, ее структурой, именами и содержанием основных каталогов. Также приобрел практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами и по проверке диска и обслуживанию файловых систем.



## Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.

**Tmpfs** - временное файловое хранилище во многих Unix-подобных ОС и в частности Linux. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо физического диска.

2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.

```
bin dev home lib64 media opt root sbin sys usr
boot etc lib lost+found mnt proc run srv tmp var
[liveuser@localhost-live /]$
```

*Директории первого уровня*

- `/bin` - каталог, содержащий исполняемые файлы. Монтируется на корневую файловую систему, должен быть доступен даже если никакие другие файловые системы не смонтированы.
- `/dev` - содержит файлы физических устройств, которые могут входить в состав аппаратного обеспечения компьютера.
- `/home` - каталог, содержащий в себе домашние каталоги пользователей операционной системы, в которых хранятся их данные, настройки, пароли и т.д.
- `/lib64` - каталог, присутствующий на 64-битных системах, содержащий набор библиотек и компонентов компилятора языка C для 64-битных систем.
- `/media` - точка для автоматического монтирования различных устройств: USB-накопители, CD-ROM и т.д.
- `/opt` - содержит подкаталоги для дополнительных пакетов программного обеспечения.
- `/root` - Домашний каталог пользователя root. Он мог бы лежать в папке `/home`, но находится на первом уровне для большей надежности системы.
- `/sbin` - содержит исполняемые файлы, предназначенные для запуска пользователем при администрировании системы.
- `/sys` - точка монтирования виртуальной файловой системы `sysfs` с информацией об устройствах, драйверах, ядре ОС, гипервизоре и т.д.
- `/usr` - бинарные файлы, используемые только пользователями, например игры.

- /boot - файлы, нужные для запуска ОС (образы ядер Linux и файлы менеджеров загрузки).
- /etc - содержит основные конфигурационные файлы операционной системы и различных программ.
- /lib - директория, предназначенная для хранения системных библиотек и компонентов компилятора языка C, необходимых для работы программ из каталогов /bin и /sbin.
- /lost+found - При сбое в работе файловой системы и дальнейшей проверки файлов все найденные поврежденные файлы помещаются в каталог /lost+found, чтобы их можно было попытаться восстановить.
- /mnt - Точка ручного монтирования (используется для временного монтирования с применением команды mount).
- /proc - содержит файлы ядра и процессора. В эту директорию примонтирована виртуальная файловая система procfs, в которой содержатся специальные файлы, в которых находится информация о системе и выполняющихся процессах.
- /run - каталог для хранения вспомогательных временных файлов приложений.
- /srv - содержит данные сервисных служб, предоставляемых системой.
- /tmp - содержит временные файлы, которые удаляются при выключении или перезагрузке системы.
- /var - содержит журналы ОС, системные логи и cache-файлы.

### 3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?

Ответ: mkfs.

### 4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?

Основной причиной нарушения целостности файловой системы является остановка ОС в момент обновления метаданных файлов. Это может привести к дублированию или наоборот потере файлов из-за нарушения инварианта счетчика ссылок или другой важной информации.

Для диагностики и исправления ошибок файловой системы используется команда fsck.

## 5. Как создаётся файловая система?

Создать новую файловую систему можно с помощью команды `mkfs`.

## 6. Дайте характеристику командам для просмотра текстовых файлов.

- `cat` - подходит для просмотра файлов небольшого размера, выводит всё их содержимое в консоль.
- `less` - подходит для постраничного просмотра файла, можно переключаться между страницами.
- `head` - выводит несколько первых строк файла, по умолчанию - 10, но можно указать и другое число.
- `tail` - выводит несколько последних строк файла, по умолчанию - 10, но можно указать и другое число.

## 7. Приведите основные возможности команды `cp` в Linux.

Команда `cp` применяется для копирования файлов и каталогов, имеется возможность копирования несколько файлов в один каталог одной командой. Можно копировать файлы в произвольном каталоге, указывая полный или при возможности относительный путь. Для копирования каталогов вместе с их содержимым указывается опция `-r`. Подробнее об этой команде написано в теоретическом введении.

## 8. Приведите основные возможности команды `mv` в Linux.

Команда `mv` применится для перемещения и смены имени файлов и каталогов. Подробнее об этой команде написано в теоретическом введении.

## 9. Что такое права доступа? Как они могут быть изменены?

Права доступа - набор значений, устанавливающий возможности тех или иных пользователей читать, изменять или исполнять конкретный файл или директорию. Права доступа к файлу можно изменить с помощью команды `chmod`. Подробнее об этой команде написано в теоретическом введении.