

Лабораторная работа 5

Сафин Андрей Алексеевич

Содержание

| | | |
|----------|---------------------------------------|-----------|
| 1 | Цель работы | 5 |
| 2 | Задание | 6 |
| 3 | Теоретическое введение | 7 |
| 4 | Выполнение лабораторной работы | 8 |
| 5 | Ответы на контрольные вопросы | 15 |
| 6 | Выводы | 17 |

Список иллюстраций

| | | |
|------|---|----|
| 4.1 | Пример использования <code>cp</code> | 8 |
| 4.2 | Пример использования <code>cp -r</code> | 8 |
| 4.3 | Пример использования <code>mv</code> | 9 |
| 4.4 | Пример использования <code>chmod</code> | 9 |
| 4.5 | Пример использования <code>fsck</code> | 10 |
| 4.6 | Манипуляции, описанные во втором пункте задания | 10 |
| 4.7 | Манипуляции, описанные в четвертом пункте задания | 11 |
| 4.8 | <code>man mount</code> | 12 |
| 4.9 | <code>man fsck</code> | 13 |
| 4.10 | <code>man mkfs</code> | 14 |
| 4.11 | <code>man kill</code> | 14 |

Список таблиц

1 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.

2 Задание

Выполнить манипуляции с файлами и директориями, описанные в лабораторной работе.

3 Теоретическое введение

Файловая система - это способ физического хранения информации на жестком диске, которую группируют как файлы, а так же формат её хранения.

В рамках системы Linux для работы с файловой системой используются следующие команды:

`touch` - создание файла

`mkdir` - создание директории

`cat` - чтение файла

`cd` - смена директории

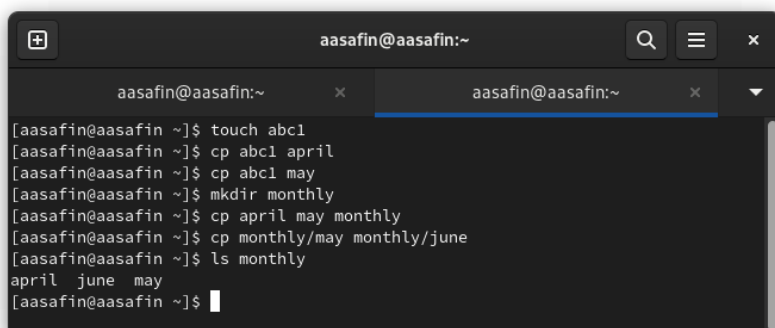
`mv` - перемещение файла/каталога с возможным переименованием

`cp` - копирование файла/каталога с возможным переименованием

`chmod` - изменение прав доступа к файлу/каталогу

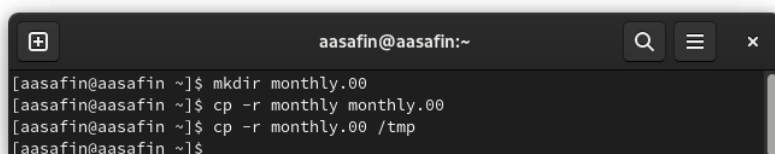
4 Выполнение лабораторной работы

1. Выполнены примеры использования команд для работы с файловой системой, описанные в лабораторной работе (рис. 4.1-4.5).



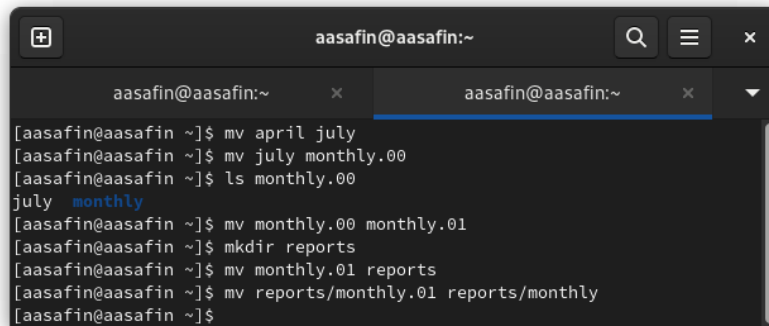
```
aasafin@aasafin:~  
[aasafin@aasafin ~]$ touch abc1  
[aasafin@aasafin ~]$ cp abc1 april  
[aasafin@aasafin ~]$ cp abc1 may  
[aasafin@aasafin ~]$ mkdir monthly  
[aasafin@aasafin ~]$ cp april may monthly  
[aasafin@aasafin ~]$ cp monthly/may monthly/june  
[aasafin@aasafin ~]$ ls monthly  
april  june  may  
[aasafin@aasafin ~]$
```

Рис. 4.1: Пример использования cp



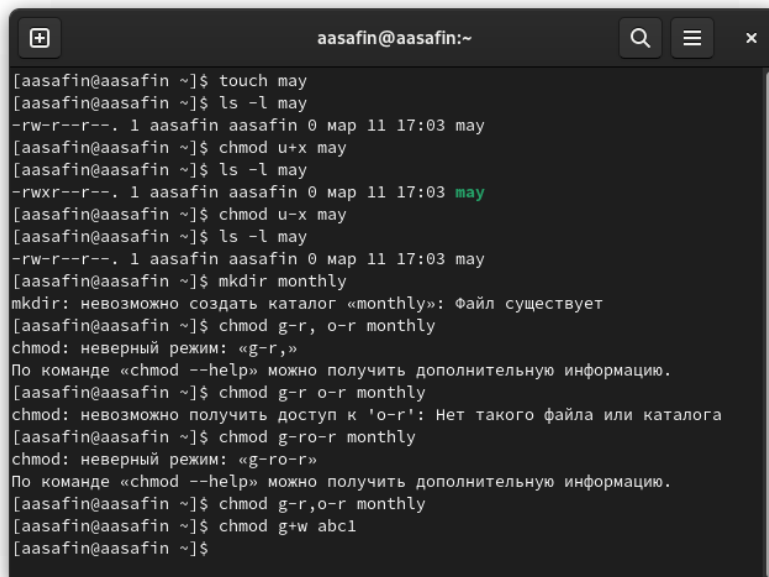
```
aasafin@aasafin:~  
[aasafin@aasafin ~]$ mkdir monthly.00  
[aasafin@aasafin ~]$ cp -r monthly monthly.00  
[aasafin@aasafin ~]$ cp -r monthly.00 /tmp  
[aasafin@aasafin ~]$
```

Рис. 4.2: Пример использования cp -r

A terminal window titled 'aasafin@aasafin:~' with a search icon, a menu icon, and a close icon in the top right corner. It contains two tabs, both labeled 'aasafin@aasafin:~'. The terminal shows a series of commands and their outputs: 'mv april july', 'mv july monthly.00', 'ls monthly.00' (output: 'july monthly'), 'mv monthly.00 monthly.01', 'mkdir reports', 'mv monthly.01 reports', 'mv reports/monthly.01 reports/monthly', and finally '\$'.

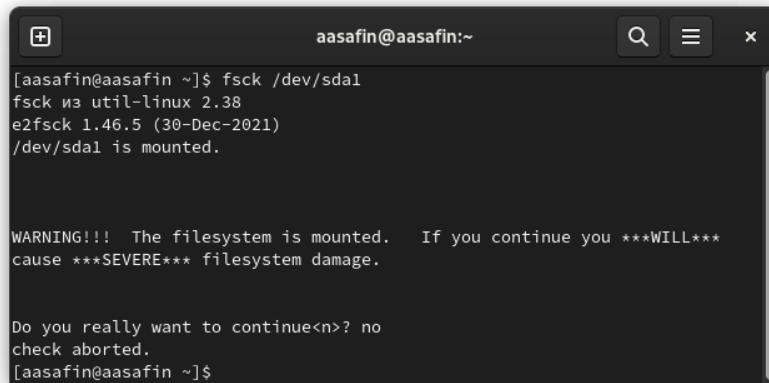
```
[aasafin@aasafin ~]$ mv april july
[aasafin@aasafin ~]$ mv july monthly.00
[aasafin@aasafin ~]$ ls monthly.00
july  monthly
[aasafin@aasafin ~]$ mv monthly.00 monthly.01
[aasafin@aasafin ~]$ mkdir reports
[aasafin@aasafin ~]$ mv monthly.01 reports
[aasafin@aasafin ~]$ mv reports/monthly.01 reports/monthly
[aasafin@aasafin ~]$
```

Рис. 4.3: Пример использования mv

A terminal window titled 'aasafin@aasafin:~' with a search icon, a menu icon, and a close icon in the top right corner. It contains a single tab labeled 'aasafin@aasafin:~'. The terminal shows a series of commands and their outputs: 'touch may', 'ls -l may' (output: '-rw-r--r--. 1 aasafin aasafin 0 map 11 17:03 may'), 'chmod u+x may', 'ls -l may' (output: '-rwxr--r--. 1 aasafin aasafin 0 map 11 17:03 may'), 'chmod u-x may', 'ls -l may' (output: '-rw-r--r--. 1 aasafin aasafin 0 map 11 17:03 may'), 'mkdir monthly' (output: 'mkdir: невозможно создать каталог «monthly»: Файл существует'), 'chmod g-r, o-r monthly' (output: 'chmod: неверный режим: «g-r,»'), 'chmod g-r o-r monthly' (output: 'chmod: невозможно получить доступ к 'o-r': Нет такого файла или каталога'), 'chmod g-ro-r monthly' (output: 'chmod: неверный режим: «g-ro-r»'), 'chmod g-r,o-r monthly', and 'chmod g+w abc1'.

```
[aasafin@aasafin ~]$ touch may
[aasafin@aasafin ~]$ ls -l may
-rw-r--r--. 1 aasafin aasafin 0 map 11 17:03 may
[aasafin@aasafin ~]$ chmod u+x may
[aasafin@aasafin ~]$ ls -l may
-rwxr--r--. 1 aasafin aasafin 0 map 11 17:03 may
[aasafin@aasafin ~]$ chmod u-x may
[aasafin@aasafin ~]$ ls -l may
-rw-r--r--. 1 aasafin aasafin 0 map 11 17:03 may
[aasafin@aasafin ~]$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
[aasafin@aasafin ~]$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
[aasafin@aasafin ~]$ chmod g-r o-r monthly
chmod: невозможно получить доступ к 'o-r': Нет такого файла или каталога
[aasafin@aasafin ~]$ chmod g-ro-r monthly
chmod: неверный режим: «g-ro-r»
По команде «chmod --help» можно получить дополнительную информацию.
[aasafin@aasafin ~]$ chmod g-r,o-r monthly
[aasafin@aasafin ~]$ chmod g+w abc1
[aasafin@aasafin ~]$
```

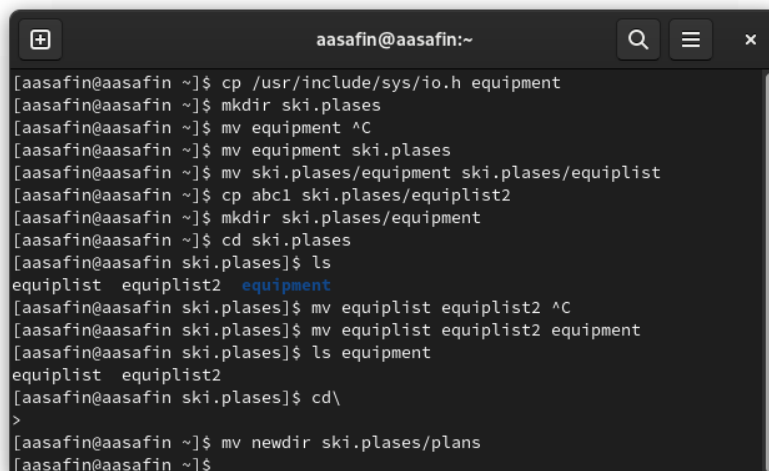
Рис. 4.4: Пример использования chmod



```
aasafin@aasafin:~  
[aasafin@aasafin ~]$ fsck /dev/sda1  
fsck из util-linux 2.38  
e2fsck 1.46.5 (30-Dec-2021)  
/dev/sda1 is mounted.  
  
WARNING!!! The filesystem is mounted.  If you continue you ***WILL***  
cause ***SEVERE*** filesystem damage.  
  
Do you really want to continue<n>? no  
check aborted.  
[aasafin@aasafin ~]$
```

Рис. 4.5: Пример использования fsck

2. Проведены манипуляции (такие, как создание, перемещение, копирование и переименовывание) с файлами и каталогами, описанные во втором пункте задания (рис. 4.6).

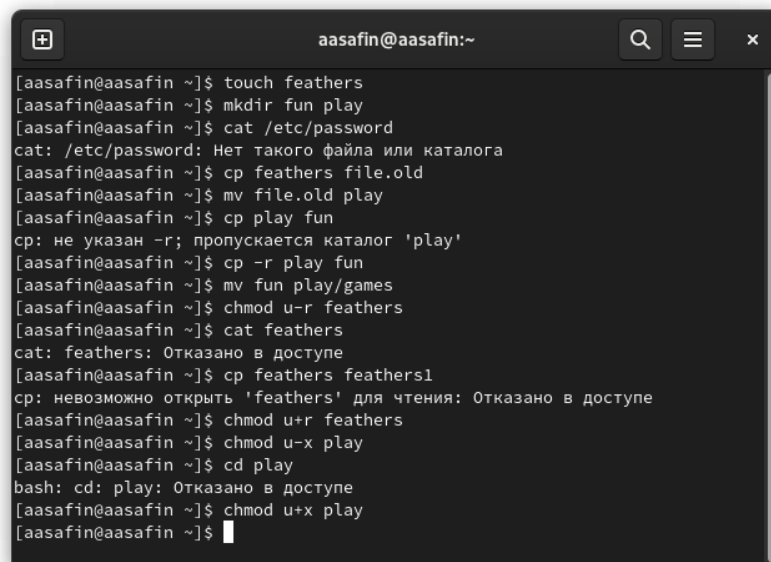


```
aasafin@aasafin:~  
[aasafin@aasafin ~]$ cp /usr/include/sys/io.h equipment  
[aasafin@aasafin ~]$ mkdir ski.plases  
[aasafin@aasafin ~]$ mv equipment ^C  
[aasafin@aasafin ~]$ mv equipment ski.plases  
[aasafin@aasafin ~]$ mv ski.plases/equipment ski.plases/equiplist  
[aasafin@aasafin ~]$ cp abc1 ski.plases/equiplist2  
[aasafin@aasafin ~]$ mkdir ski.plases/equipment  
[aasafin@aasafin ~]$ cd ski.plases  
[aasafin@aasafin ski.plases]$ ls  
equiplist  equiplist2  equipment  
[aasafin@aasafin ski.plases]$ mv equiplist equiplist2 ^C  
[aasafin@aasafin ski.plases]$ mv equiplist equiplist2 equipment  
[aasafin@aasafin ski.plases]$ ls equipment  
equiplist  equiplist2  
[aasafin@aasafin ski.plases]$ cd\  
>  
[aasafin@aasafin ~]$ mv newdir ski.plases/plans  
[aasafin@aasafin ~]$
```

Рис. 4.6: Манипуляции, описанные во втором пункте задания

3. Определены опции команды chmod, присваивающие перечисленным в задании лабораторной работы файлам соответствующие права: 744, 711, 544, 664.

4. Проведены манипуляции (такие, как создание, перемещение, копирование и переименовывание и присваивание прав) с файлами и каталогами, описанные во втором пункте задания (рис. 4.7). Как можно видеть, при отсутствии прав чтения файла невозможно использовать команды `cat` и `cp`, а при отсутствии права на выполнение директории нельзя её посещать.

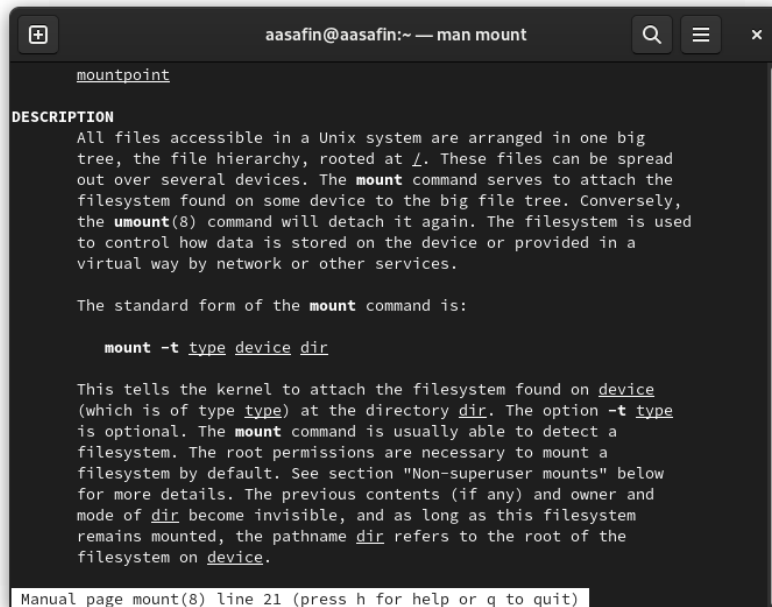


```
aasafin@aasafin:~  
[aasafin@aasafin ~]$ touch feathers  
[aasafin@aasafin ~]$ mkdir fun play  
[aasafin@aasafin ~]$ cat /etc/passwd  
cat: /etc/passwd: Нет такого файла или каталога  
[aasafin@aasafin ~]$ cp feathers file.old  
[aasafin@aasafin ~]$ mv file.old play  
[aasafin@aasafin ~]$ cp play fun  
cp: не указан -r; пропускается каталог 'play'  
[aasafin@aasafin ~]$ cp -r play fun  
[aasafin@aasafin ~]$ mv fun play/games  
[aasafin@aasafin ~]$ chmod u-r feathers  
[aasafin@aasafin ~]$ cat feathers  
cat: feathers: Отказано в доступе  
[aasafin@aasafin ~]$ cp feathers feathers1  
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе  
[aasafin@aasafin ~]$ chmod u+r feathers  
[aasafin@aasafin ~]$ chmod u-x play  
[aasafin@aasafin ~]$ cd play  
bash: cd: play: Отказано в доступе  
[aasafin@aasafin ~]$ chmod u+x play  
[aasafin@aasafin ~]$
```

Рис. 4.7: Манипуляции, описанные в четвертом пункте задания

5. Использована команда `man` к командам `mount`, `fsck`, `mkfs`, `kill`:

`Mount` прикрепляет определенную файловую систему к директории (рис. 4.8).



```
aasafin@aasafin:~ — man mount
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 21 (press h for help or q to quit)
```

Рис. 4.8: man mount

fsck проверяет и опционально восстанавливает файловые системы Linux (рис. 4.9).

```
aasafin@aasafin:~ — man fsck

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

32     Checking canceled by user request

128    Shared-library error

The exit status returned when multiple filesystems are checked is the
Manual page fsck(8) line 4 (press h for help or q to quit)
```

Рис. 4.9: man fsck

mkfs создает файловую систему на носителе или его части (от make filesystem) (рис. 4.10).

```

aasafin@aasafin:~ — man mkfs
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.fstype) 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.

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

Рис. 4.10: man mkfs

kill прекращает процессы, указанные в аргументе (рис. 4.11).

```

aasafin@aasafin:~ — man kill
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.

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

Рис. 4.11: man kill

5 Ответы на контрольные вопросы

1. tmpfs - временное файловое хранилище, предназначенное для монтирования файловой системы. devtmpfs - позволяет ядру создать экземпляр tmpfs с именем devtmpfs до регистрации устройств с драйверами ext4 - версия журналируемой файловой системы extended file system 2006-го года, широко используемая в системах UNIX. btrfs - файловая система для Linux основанная на B-деревьях и использующая принцип copy-on-write.
2. Файловая система представляет собой пространство раздела диска, разбитое на целое число КБ. Главным каталогом является корневой /, все данные примонтированы к нему и хранятся в подкаталогах. Доступ к данным получается при указании адреса соответствующего каталога.

BIN - исполняемые файлы

SBIN - системные исполняемые файлы

ETC - конфигурационные файлы

DEV - файлы устройств

PROC - информация о процессах

VAR - переменные файлы

TMP - временные файлы

USR - программы пользователя

HOME - домашняя папка, содержащая каталоги всех пользователей

BOOT - файлы загрузчика

LIB - системные библиотеки

OPT - дополнительные программы

MNT - каталог, в который монтируются внешние и дополнительные файловые системы

MEDIA - съемные носители

SRV - файлы серверов и сервисов

RUN - процессы

3. Команда mount

4. Целостность нарушается при прерывании процессов без возможности их подчистки (clean-up). Восстанавливается с помощью fsck.

5. С помощью mkfs.

6. cat - выводит содержимое less - выводит содержимое по частям, позволяя его листать head - выводит первые 10 строк tail - выводит последние 10 строк

7. cp позволяет копировать файлы в иные директории под старым либо новым именем. Так же может рекурсивно копировать директории (-r).

8. mv позволяет перемещать файлы в иные директории под старым либо новым именем. Так же может рекурсивно перемещать директории (-r). С её помощью можно просто переименовать файл.

9. Права доступа - это список действий с файлом или директорией, которые могут производить различные пользователи. Они изменяются с помощью команды chmod.

6 Выводы

Описанные в лабораторной работе действия произведены, навык работы с файловой системой приобретен.