

# **Отчёт по лабораторной работе №6**

**Операционные системы**

Ильина Любовь Александровна

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Выводы	26

# Список иллюстраций

3.1	Создание файла командой touch . . . . .	8
3.2	Просмотр файла командой cat . . . . .	8
3.3	Просмотр файла командой less . . . . .	8
3.4	Просмотр файла командой less . . . . .	9
3.5	Просмотр первых 10 строк командой head . . . . .	9
3.6	Просмотр первых 3 строк командой head с опцией -3 . . . . .	10
3.7	Просмотр последних 15 строк командой tail с опцией -15 . . . . .	11
3.8	Копирование файла io.h из каталога /usr/include/sys/io.h в домашний, переименование файла в equipment . . . . .	11
3.9	Создание директории ski.places, перемещение в нее файла equipment, переименование файла в equiplist . . . . .	12
3.10	Создание файла abc1, копирование файла в ~/ski.places, переименование файла . . . . .	12
3.11	Создание каталога equipment, перемещение файлов в каталог ~/ski.places/equipment . . . . .	12
3.12	Создание, перемещение нового каталога в ~/ski.places, переименование каталога в plans . . . . .	13
3.13	Создание файлов с указанными правами . . . . .	13
3.14	Ввод команды less для просмотра содержимого файла /etc/password . . . . .	14
3.15	Просмотр содержимого файла /etc/password командой less . . . . .	14
3.16	Копирование, перемещение файла, копирование, перемещение, переименование каталога . . . . .	15
3.17	Изменение прав файла feathers . . . . .	15
3.18	Изменение прав каталога play . . . . .	16
3.19	Просмотр описания и опций команды mount . . . . .	16
3.20	Просмотр описания и опций команды fsck . . . . .	17
3.21	Просмотр описания и опций команды mkfs . . . . .	18
3.22	Просмотр описания и опций команды kill . . . . .	19
3.23	Просмотр файловых систем . . . . .	20
3.24	Просмотр структуры файловой системы . . . . .	21

## **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

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:
  - 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.
  - 2.2. В домашнем каталоге создайте директорию `~/ski.places`.
  - 2.3. Переместите файл `equipment` в каталог `~/ski.places`.
  - 2.4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
  - 2.5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
  - 2.6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
  - 2.7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
  - 2.8. 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:
  - 3.1. `drwxr-r- ... australia`
  - 3.2. `drwx-x-x ... play`
  - 3.3. `-r-xr-r- ... my_os`
  - 3.4. `-rw-rw-r- ... feathers`При необходимости создайте нужные файлы.
4. Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
  - 4.1. Просмотрите содер-

- жимое файла `/etc/passwd`. 4.2. Скопируйте файл `~/feathers` в файл `~/file.old`. 4.3. Переместите файл `~/file.old` в каталог `~/play`. 4.4. Скопируйте каталог `~/play` в каталог `~/fun`. 4.5. Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`. 4.6. Лишите владельца файла `~/feathers` права на чтение. 4.7. Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`? 4.8. Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`? 4.9. Дайте владельцу файла `~/feathers` право на чтение. 4.10. Лишите владельца каталога `~/play` права на выполнение. 4.11. Перейдите в каталог `~/play`. Что произошло? 4.12. Дайте владельцу каталога `~/play` право на выполнение.
5. Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.

## 3 Выполнение лабораторной работы

1. Выполните все примеры, приведённые в первой части описания лабораторной работы. (рис. 3.1) - 3.6)).

```
[lailjina@lailjina report]$ touch test6  
[lailjina@lailjina report]$ ls  
bib image Makefile pandoc report.docx report.md report.pdf test6
```

Рис. 3.1: Создание файла командой touch

```
[lailjina@lailjina report]$ cat report.md  
---  
## Front matter  
title: "Отчёт по лабораторной работе № 5"  
subtitle: "Операционные системы"  
author: "Ильина Любовь Александровна"  
  
## Generic options  
lang: ru-RU  
toc-title: "Содержание"  
  
## Bibliography  
bibliography: bib/cite.bib  
csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl  
  
## Pdf output format  
toc: true # Table of contents  
toc-depth: 2
```

Рис. 3.2: Просмотр файла командой cat

```
[lailjina@lailjina report]$ less report.md
```

Рис. 3.3: Просмотр файла командой less





```
lailjina@lailjina:~/work/study/2022-2023/Операционные системы/os-intro/labs/lab05/report _ □ ×
File Edit View Search Terminal Help
---
## Front matter
title: "Отчёт по лабораторной работе № 5"
subtitle: "Операционные системы"
author: "Ильина Любовь Александровна"

## Generic options
lang: ru-RU
toc-title: "Содержание"

## Bibliography
bibliography: bib/cite.bib
csl: pandoc/csl/gost-r-7-0-5-2008-numeric.csl

## Pdf output format
toc: true # Table of contents
toc-depth: 2
lof: true # List of figures
lot: true # List of tables
fontsize: 12pt
linestretch: 1.5
papersize: a4
documentclass: scrreprt
## I18n polyglossia
polyglossia-lang:
  name: russian
  options:
report.md
```

Рис. 3.4: Просмотр файла командой less

```
[lailjina@lailjina report]$ head report.md
---
## Front matter
title: "Отчёт по лабораторной работе № 5"
subtitle: "Операционные системы"
author: "Ильина Любовь Александровна"

## Generic options
lang: ru-RU
toc-title: "Содержание"

[lailjina@lailjina report]$ █
```

Рис. 3.5: Просмотр первых 10 строк командой head

```
[lailjina@lailjina report]$ head -3 report.md
---
## Front matter
title: "Отчёт по лабораторной работе № 5"
[lailjina@lailjina report]$ tail -15 report.md
```

6. Чтобы определить, какие команды выполнил пользователь в сеансе работы, необходимо воспользоваться командой «history».

7. Чтобы исправить или запустить на выполнение команду, которую пользователь уже использовал в сеансе работы, необходимо: в первом случае: воспользоваться конструкцией `!:s//`, во втором случае: `!`. Примеры на рис. @fig:023

8. Чтобы записать в одной строке несколько команд, необходимо между ними поставить `;` . Например, `cd /tmp; ls`.

9. Символ обратного слэша `\` позволяет использовать управляющие символы ( `"."`, `"/"`, `"$"`, `"*"`, `"["`, `"]"`, `"^"`, `"&"`) без их интерпретации командной оболочкой; процедура добавления данного символа перед управляющими символами называется экранированием символов. Например, команда `ls newdir/morefun` отобразит содержимое каталога `newdir/morefun`.

10. Команда `ls -l` отображает список каталогов и файлов с подробной информацией о них (тип файла, право доступа, число ссылок, владелец, размер, дата последней ревизии, имя файла или каталога).

11. Полный, абсолютный путь от корня файловой системы — этот путь начинается от корня `/` и описывает весь путь к файлу или каталогу; Относительный путь — это путь к файлу относительно текущего каталога (каталога, где находится пользователь). Например, `cd /newdir/morefun` — абсолютный путь, `cd newdir` — относительный путь.

12. Чтобы получить необходимую информацию о команде, необходимо воспользоваться конструкцией `man [имя_команды]`, либо использовать опцию `help`, которая предусмотрена для некоторых команд.

13. Для автоматического дополнения вводимых команд служит клавиша `Tab`. Вывод: В ходе выполнения данной лабораторной работы я приобрела практические навыки взаимодействия с системой посредством командной строки.

Рис. 3.6: Просмотр первых 3 строк командой `head` с опцией `-3`

```
[lailjina@lailjina report]$ tail -15 report.md
```

6. Чтобы определить, какие команды выполнил пользователь в сеансе работы, необходимо воспользоваться командой «history».

7. Чтобы исправить или запустить на выполнение команду, которую пользователь уже и спользовал в сеансе работы, необходимо: в первом случае: воспользоваться конструкцией !:s//, во втором случае: !. Примеры на рис.@fig:023

8. Чтобы записать в одной строке несколько команд, необходимо между ними поставить ; . Например, «cd /tmp; ls».

9. Символ обратного слэша \ позволяет использовать управляющие символы ( ".", "/", "\$", "\*", "[", "]", "^", "&") без их интерпретации командной оболочкой; процедура добавления данного символа перед управляющими символами называется экранированием символов. Например, команда «ls newdir/morefun» отобразит содержимое каталога newdir/morefun.

10. Команда «ls -l» отображает список каталогов и файлов с подробной информацией о них (тип файла, право доступа, число ссылок, владелец, размер, дата последней ревизии, имя файла или каталога).

11. Полный, абсолютный путь от корня файловой системы – этот путь начинается от корня "/" и описывает весь путь к файлу или каталогу; Относительный путь – это путь к файлу относительно текущего каталога (каталога, где находится пользователь). Например, «cd /newdir/morefun» – абсолютный путь, «cd newdir» – относительный путь.

12. Чтобы получить необходимую информацию о команде, необходимо воспользоваться конструкцией man [имя\_команды], либо использовать опцию help, которая предусмотрена для некоторых команд.

13. Для автоматического дополнения вводимых команд служит клавиша Tab. Вывод: В ходе выполнения данной лабораторной работы я приобрела практические навыки взаимодействия с системой посредством командной строки.

Рис. 3.7: Просмотр последних 15 строк командой tail с опцией -15

2. Выполним следующие действия: 2.1. Скопируем файл /usr/include/sys/io.h в домашний каталог и назовем его equipment. Перед этим убедимся, что файл io.h есть в каталоге /usr/include/sys/. (рис. 3.8)

```
[lailjina@lailjina ~]$ cd /usr/include/sys
[lailjina@lailjina sys]$ find io.h
io.h
[lailjina@lailjina sys]$ cp io.h /home/lailjina/equipment
[lailjina@lailjina sys]$ cd /h^C
[lailjina@lailjina sys]$ cd /home/lailjina
[lailjina@lailjina ~]$ cd ..
[lailjina@lailjina home]$ ls
lailjina
[lailjina@lailjina home]$ cd lailjina
[lailjina@lailjina ~]$ ls
Desktop  equipment  newdir     Python-3.10.2  python3-virtualenv  Videos
Documents  get-pip.py  pandoc     Python-3.10.2.tgz  Templates            work
Downloads  Music       Pictures   Python-3.8.1    texlive              Work
env        myenv      Public     Python-3.8.1.tgz  usr
```

Рис. 3.8: Копирование файла io.h из каталога /usr/include/sys/io.h в домашний, переименование файла в equipment

- 2.2. В домашнем каталоге создаем директорию ~/ski.plases 2.3. Пере-

местим файл `equipment` в каталог `~/ski.places`. 2.4. Переименуем файл `~/ski.places/equipment` в `~/ski.places/equiplist`. (рис. 3.9)

```
[lailjina@lailjina ~]$ mkdir ski.places
[lailjina@lailjina ~]$ mv equipment ski.places
[lailjina@lailjina ~]$ cd ski.places
[lailjina@lailjina ski.places]$ mv ^C
[lailjina@lailjina ski.places]$ mv equipment equiplist
[lailjina@lailjina ski.places]$ ls
equiplist
```

Рис. 3.9: Создание директории `ski.places`, перемещение в нее файла `equipment`, переименование файла в `equiplist`

2.5. Создадим в домашнем каталоге файл `abc1` и скопируем его в каталог `~/ski.places`, назовем его `equiplist2`. (рис. 3.10)

```
[lailjina@lailjina ~]$ touch abc1
[lailjina@lailjina ~]$ cp abc1 /ski.places
cp: cannot create regular file '/ski.places': Permission denied
[lailjina@lailjina ~]$ cp abc1 ski.places
[lailjina@lailjina ~]$ cd ski.places
[lailjina@lailjina ski.places]$ mv abc1 equiplist2
[lailjina@lailjina ski.places]$ ls
equiplist equiplist2
```

Рис. 3.10: Создание файла `abc1`, копирование файла в `~/ski.places`, переименование файла

2.6. Создадим каталог с именем `equipment` в каталоге `~/ski.places`. 2.7. Переместим файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`. (рис. 3.11)

```
[lailjina@lailjina ~]$ cd ski.places
[lailjina@lailjina ski.places]$ mkdir equipment
[lailjina@lailjina ski.places]$ mv equiplist equiplist2 equipment
[lailjina@lailjina ski.places]$ ls equipment
equiplist equiplist2
```

Рис. 3.11: Создание каталога `equipment`, перемещение файлов в каталог `~/ski.places/equipment`

2.8. Создадим и переместим каталог `~/newdir` в каталог `~/ski.places` и назовем его `plans`. (рис. 3.12)

```

[lailjina@lailjina ~]$ mkdir newdir
[lailjina@lailjina ~]$ mv newdir ski.places/plans
[lailjina@lailjina ~]$ ls ski.places
equipment  plans

```

Рис. 3.12: Создание, перемещение нового каталога в ~/ski.places, переименование каталога в plans

3. Определим опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет: 3.1. `drwxr-r-` ... `australia` 3.2. `drwx-x-x` ... `play` 3.3. `-r-xr-r-` ... `my_os` 3.4. `-rw-rw-r-` ... `feathers` При необходимости создадим нужные файлы. (рис. 3.13)

```

[lailjina@lailjina ~]$ mkdir australia
[lailjina@lailjina ~]$ chmod 744 australia
[lailjina@lailjina ~]$ mkdir play
[lailjina@lailjina ~]$ chmod 711 play
[lailjina@lailjina ~]$ touch my_os
[lailjina@lailjina ~]$ chmod 544 my_os
[lailjina@lailjina ~]$ touch feathers
[lailjina@lailjina ~]$ chmod 664 feathers
[lailjina@lailjina ~]$ ls -l
total 50428
-rw-rw-r--. 1 lailjina lailjina      0 Apr  2 14:08 abc1
drwxr--r--. 2 lailjina lailjina      6 Apr  2 14:56 australia
drwxr-xr-x. 2 lailjina lailjina      6 Feb 24 10:55 Desktop
drwxr-xr-x. 3 lailjina lailjina    108 Apr  2 12:51 Documents
drwxr-xr-x. 4 lailjina lailjina   4096 Apr  1 02:56 Downloads
drwxrwxr-x. 5 lailjina lailjina     100 Mar 23 17:46 env
-rw-rw-r--. 1 lailjina lailjina      0 Apr  2 14:57 feathers
-rw-rw-r--. 1 lailjina lailjina 2574273 Mar 25 16:01 get-pip.py
drwxr-xr-x. 2 lailjina lailjina      6 Feb 24 10:55 Music
drwxrwxr-x. 5 lailjina lailjina      74 Mar 25 19:18 myenv
-r-xr--r--. 1 lailjina lailjina      0 Apr  2 14:57 my_os
drwxrwxr-x. 4 lailjina lailjina     32 Mar 17 02:00 pandoc
drwxr-xr-x. 2 lailjina lailjina      6 Feb 24 10:55 Pictures
drwx--x--x. 2 lailjina lailjina      6 Apr  2 14:56 play
drwxr-xr-x. 2 lailjina lailjina      6 Feb 24 10:55 Public
drwxr-xr-x. 16 lailjina lailjina    4096 Mar 23 15:00 Python-3.10.2
-rw-rw-r--. 1 lailjina lailjina 25067363 Jan 13  2022 Python-3.10.2.tgz
drwxr-xr-x. 20 lailjina lailjina    4096 Mar 23 18:04 Python-3.8.1
-rw-rw-r--. 1 lailjina lailjina 23978360 Mar 23 15:19 Python-3.8.1.tgz
drwxrwxr-x. 5 lailjina lailjina     100 Mar 23 16:11 python3-virtualenv
drwxrwxr-x. 4 lailjina lailjina      36 Apr  2 14:28 ski.places
drwxr-xr-x. 2 lailjina lailjina      6 Feb 24 10:55 Templates
drwxrwxr-x. 4 lailjina lailjina     37 Mar 16 13:54 texlive
drwxrwxr-x. 4 lailjina lailjina     30 Mar 16 16:03 usr
drwxr-xr-x. 2 lailjina lailjina      6 Feb 24 10:55 Videos
drwxrwxr-x. 3 lailjina lailjina     19 Mar  8 12:57 work
drwxrwxr-x. 3 lailjina lailjina     29 Mar 16 16:25 Work

```

Рис. 3.13: Создание файлов с указанными правами

4. Выполним упражнения ниже: 4.1. Просмотрим содержимое файла /etc/passwd командой less. (рис. 3.14)-3.15))

```
[lailijina@lailijina etc]$ less passwd
```

Рис. 3.14: Ввод команды less для просмотра содержимого файла /etc/passwd

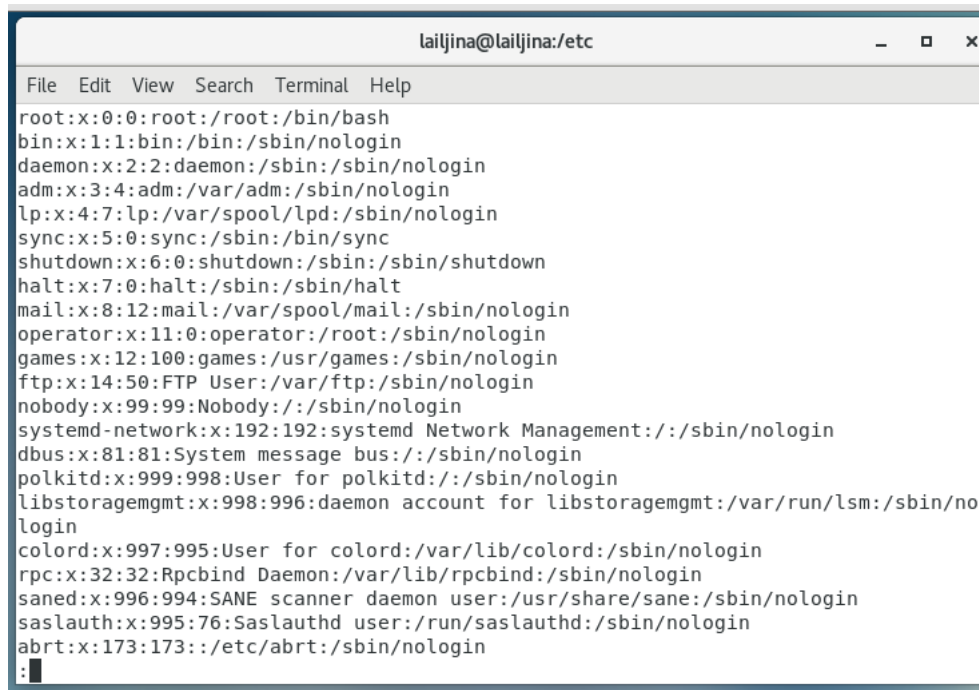


Рис. 3.15: Просмотр содержимого файла /etc/passwd командой less

4.2. Скопируем файл ~/feathers в файл ~/file.old. 4.3. Переместим файл ~/file.old в каталог ~/play. 4.4. Скопируем каталог ~/play в каталог ~/fun. 4.5. Переместим каталог ~/fun в каталог ~/play и назовем его games. (рис. 3.16)

```

[lailjina@lailjina ~]$ cp feathers file.old
[lailjina@lailjina ~]$ mv file.old play
[lailjina@lailjina ~]$ ls /play
ls: cannot access /play: No such file or directory
[lailjina@lailjina ~]$ ls play
file.old
[lailjina@lailjina ~]$ cp -r play fun
[lailjina@lailjina ~]$ mv fun play
[lailjina@lailjina ~]$ mv play/fun play/games
[lailjina@lailjina ~]$ ls play
file.old  games
[lailjina@lailjina ~]$ ls
abcl  env  my_os  Python-3.10.2  ski.plases  work
australia  feathers  pandoc  Python-3.10.2.tgz  Templates  Work
Desktop  get-pip.py  Pictures  Python-3.8.1  texlive
Documents  Music  play  Python-3.8.1.tgz  usr
Downloads  myenv  Public  python3-virtualenv  Videos
[lailjina@lailjina ~]$ ls play/games
file.old
[lailjina@lailjina ~]$ ls play
file.old  games
[lailjina@lailjina ~]$ ls play/games
file.old

```

Рис. 3.16: Копирование, перемещение файла, копирование, перемещение, переименование каталога

4.6. Лишим владельца файла ~/feathers права на чтение. 4.7. Попробуем просмотреть файл ~/feathers командой cat. 4.8. Попробуем скопировать файл ~/feathers. 4.9. Дадим владельцу файла ~/feathers право на чтение. (рис. 3.17)

```

[lailjina@lailjina ~]$ chmod u-r feathers
[lailjina@lailjina ~]$ ls -f feathers
feathers
[lailjina@lailjina ~]$ ls -l feathers
--w-rw-r--. 1 lailjina lailjina 0 Apr  2 14:57 feathers
[lailjina@lailjina ~]$ cat feathers
cat: feathers: Permission denied
[lailjina@lailjina ~]$ cp feathers feathers2
cp: cannot open 'feathers' for reading: Permission denied
[lailjina@lailjina ~]$ chmod u+r feathers
[lailjina@lailjina ~]$ ls -l feathers
-rw-rw-r--. 1 lailjina lailjina 0 Apr  2 14:57 feathers

```

Рис. 3.17: Изменение прав файла feathers

4.10. Лишим владельца каталога ~/play права на выполнение. 4.11. Передадим в каталог ~/play. Что произошло? 4.12. Дайте владельцу каталога ~/play право на выполнение. (рис. 3.18)

```

[lailjina@lailjina ~]$ chmod u-x play
[lailjina@lailjina ~]$ cd play
bash: cd: play: Permission denied
[lailjina@lailjina ~]$ chmod u+x play
[lailjina@lailjina ~]$ cd play

```

Рис. 3.18: Изменение прав каталога play

5. Прочитайте man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, приведя примеры. (рис. 3.19)-3.24)

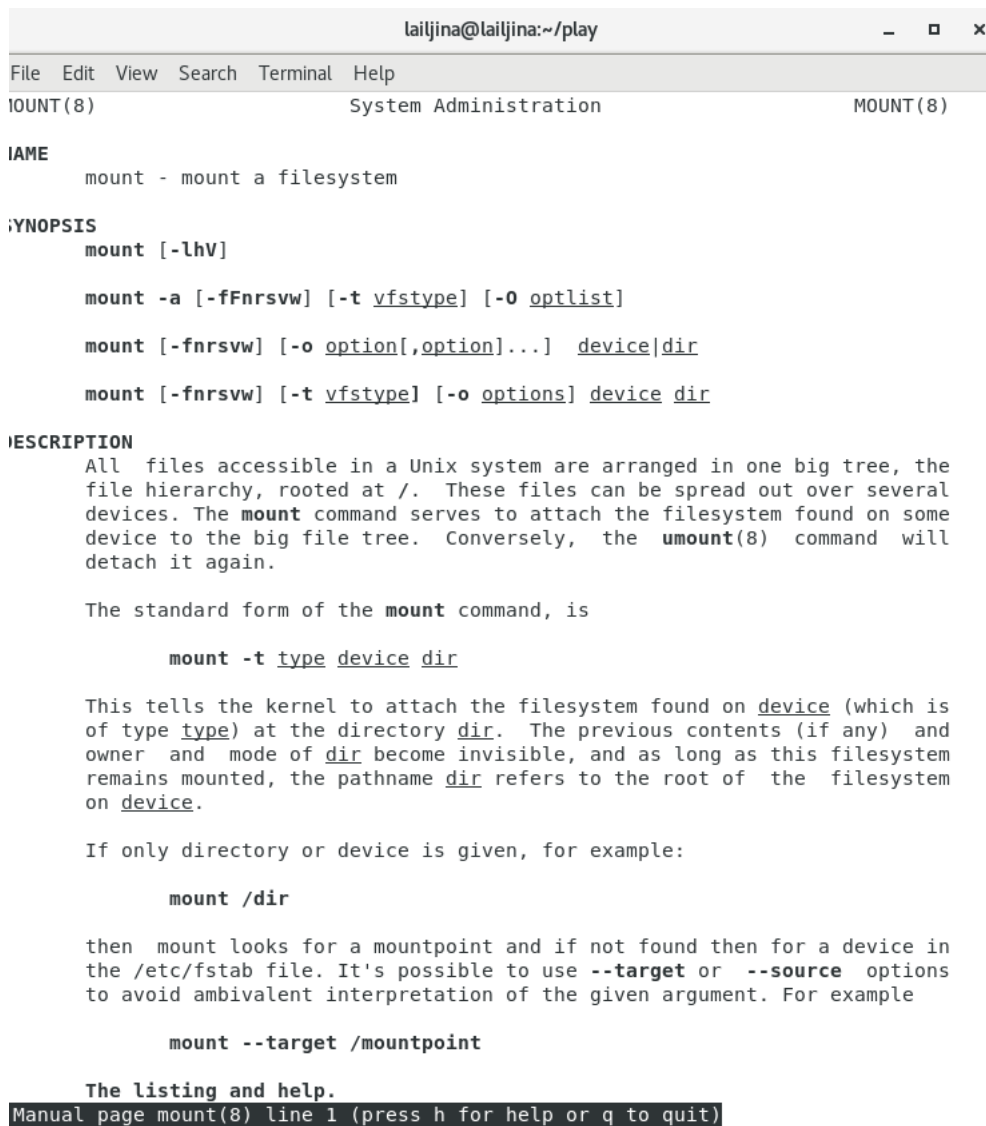


Рис. 3.19: Просмотр описания и опций команды mount



```
lailjina@lailjina:~/play
File Edit View Search Terminal Help
FSCK(8) System Administration FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

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

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystems can be a device name (e.g. /dev/hdc1, /dev/sdb2), a mount point (e.g. /, /usr, /home), or an ext2 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 code 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 code returned when multiple filesystems are checked is the bitwise OR of the exit codes for each filesystem that is checked.

    In actuality, fsck is simply a front-end for the various filesystem checkers (fsck.fstype) available under Linux. The filesystem-specific checker is searched for in /sbin first, then in /etc/fs and /etc, and finally in the directories listed in the PATH environment variable. Please see the filesystem-specific checker manual pages for further details.

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

Рис. 3.20: Просмотр описания и опций команды fsck

```
lailjina@lailjina:~/play
File Edit View Search Terminal Help
MKFS(8) System Administration MKFS(8)

NAME
    mkfs - build a Linux filesystem

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

DESCRIPTION
    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 code 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 in a number of directories, like perhaps /sbin,
/sbin/fs, /sbin/fs.d, /etc/fs, /etc (the precise list is defined at compile
    time but at least contains /sbin and /sbin/fs), and finally in the
    directories listed in the PATH environment variable. 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. Although not guaranteed, the following options are supported
        by most filesystem builders.

    -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
```

Рис. 3.21: Просмотр описания и опций команды mkfs

```
lailjina@lailjina:~/play
File Edit View Search Terminal Help
KILL(1) User Commands KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-s signal|-p] [-q sigval] [-a] [--] pid...
    kill -l [signal]

DESCRIPTION
    The command kill sends the specified signal to the specified process or process group. If no signal is specified, the TERM signal is sent. The TERM signal will kill processes which do not catch this signal. For other processes, it may be necessary to use the KILL (9) signal, since this signal cannot be caught.

    Most modern shells have a builtin kill function, with a usage rather similar to that of the command described here. The '-a' and '-p' options, and the possibility to specify processes by command name are a local extension.

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

OPTIONS
    pid... Specify the list of processes that kill should signal. Each pid can be one of five things:

        n      where n is larger than 0. The process with pid n will be signaled.

        0      All processes in the current process group are signaled.

        -1     All processes with pid larger than 1 will be signaled.

        -n    where n is larger than 1. All processes in process group n are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either the signal must be specified first, or the argument must be preceded by a '--' option, otherwise it will be taken as the signal to send.
```

Рис. 3.22: Просмотр описания и опций команды kill

Контрольные вопросы:

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу. Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, используют команду «df -Th». На моем компьютере есть следующие файловые системы: xfs, devtmpfs,tmpfs. (рис. 3.23)

```
[root@lailjina lailjina]# df -Th
Filesystem                                Type      Size  Used Avail Use% Mounted on
devtmpfs                                 devtmpfs  3.9G   0    3.9G   0% /dev
tmpfs                                    tmpfs     3.9G   0    3.9G   0% /dev/shm
tmpfs                                    tmpfs     3.9G  9.5M   3.9G   1% /run
tmpfs                                    tmpfs     3.9G   0    3.9G   0% /sys/fs/cgroup
/dev/mapper/centos_lailjina-root        xfs       37G   28G   9.4G   75% /
/dev/sda1                               xfs      1014M  214M   801M   22% /boot
tmpfs                                    tmpfs     799M   52K   799M   1% /run/user/1000
```

Рис. 3.23: Просмотр файловых систем

XFS обладает преимуществами такие как журналирование метаданных для быстрого восстановления, но, кроме того, здесь поддерживается распределение потоков ввода/вывода по группам что сильно увеличивает производительность чтения и записи данных. Но это работает только для больших файлов. Также вы можете увеличить размер файловой системы или выполнить дефрагментацию, даже если она смонтирована. `devtmpfs` позволяет ядру создать экземпляр `tmpfs` с именем `devtmpfs` при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором / минором будет предоставлять узел устройства в `devtmpfs`. `devtmpfs` монтируется на `/dev` и содержит специальные файлы устройств для всех устройств. `tmpfs` – временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система, предназначенная для быстрого и ненадёжного хранения временных данных, подходит для `/tmp` и массовой сборки пакетов/образов, предполагает наличие достаточного объёма виртуальной памяти. Файловая система `tmpfs` предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска.

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

С помощью команды `findmnt` выясним директории первого уровня.(рис. 3.24)

```
[lailjina@lailjina ~]$ findmnt
```

TARGET	SOURCE	FSTYPE	OPTIONS
/	/dev/mapper/centos_lailjina-root	xfs	rw,relatime,seclabel,a
├─/sys	sysfs	sysfs	rw,nosuid,nodev,noexec
│   └─/sys/kernel/security	securityfs	security	rw,nosuid,nodev,noexec
│   └─/sys/fs/cgroup	tmpfs	tmpfs	ro,nosuid,nodev,noexec
│       └─/sys/fs/cgroup/systemd	cgroup	cgroup	rw,nosuid,nodev,noexec
│           └─/sys/fs/cgroup/freezer	cgroup	cgroup	rw,nosuid,nodev,noexec
│               └─/sys/fs/cgroup/blkio	cgroup	cgroup	rw,nosuid,nodev,noexec
│                   └─/sys/fs/cgroup/memory	cgroup	cgroup	rw,nosuid,nodev,noexec
│                       └─/sys/fs/cgroup/cpu,cpuacct	cgroup	cgroup	rw,nosuid,nodev,noexec
│                           └─/sys/fs/cgroup/perf_event	cgroup	cgroup	rw,nosuid,nodev,noexec
│                               └─/sys/fs/cgroup/cpuset	cgroup	cgroup	rw,nosuid,nodev,noexec
│                                   └─/sys/fs/cgroup/hugetlb	cgroup	cgroup	rw,nosuid,nodev,noexec
│                                       └─/sys/fs/cgroup/net_cls,net_prio	cgroup	cgroup	rw,nosuid,nodev,noexec
│   └─/sys/fs/cgroup/pids	cgroup	cgroup	rw,nosuid,nodev,noexec
│   └─/sys/fs/cgroup/devices	cgroup	cgroup	rw,nosuid,nodev,noexec
└─/sys/fs/pstore	pstore	pstore	rw,nosuid,nodev,noexec
└─/sys/kernel/config	configfs	configfs	rw,relatime
└─/sys/fs/selinux	selinuxfs	selinuxfs	rw,relatime
└─/sys/kernel/debug	debugfs	debugfs	rw,relatime
└─/sys/fs/fuse/connections	fusectl	fusectl	rw,relatime
└─/proc	proc	proc	rw,nosuid,nodev,noexec
└─/proc/sys/fs/binfmt_misc	systemd-1	autofs	rw,relatime,fd=25,pgrp
└─/dev	devtmpfs	devtmpfs	rw,nosuid,seclabel,siz
└─/dev/shm	tmpfs	tmpfs	rw,nosuid,nodev,seclab
└─/dev/pts	devpts	devpts	rw,nosuid,noexec,relat
└─/dev/mqueue	mqueue	mqueue	rw,relatime,seclabel
└─/dev/hugepages	hugetlbfs	hugetlbfs	rw,relatime,seclabel
└─/run	tmpfs	tmpfs	rw,nosuid,nodev,seclab
└─/run/user/1000	tmpfs	tmpfs	rw,nosuid,nodev,relati
└─/run/user/1000/gvfs	gvfsd-fuse	fuse.gvf	rw,nosuid,nodev,relati
└─/boot	/dev/sda1	xfs	rw,relatime,seclabel,a
└─/var/lib/nfs/rpc_pipefs	sunrpc	rpc_pipe	rw,relatime

```
[lailjina@lailjina ~]$
```

Рис. 3.24: Просмотр структуры файловой системы

Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел –/ (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам.

“/” – корень. Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом

каталоге. /sys (system) - Информация о системе. Назначение каталогов Linux из этой папки - получение информации о системе непосредственно от ядра. Это еще одна файловая система организуемая ядром и позволяющая просматривать и изменить многие параметры работы системы, например, работу swapon, контролировать вентиляторы и многое другое. "/PROC" – информация о процессах. По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его Pid, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов. "/DEV" – файлы устройств в Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры – это просто файлы в каталоге /dev/. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов. "/RUN" - процессы. Каталог, содержащий PID файлы процессов, похожий на "/var/run", но в отличие от него, он размещен в TMPFS, а поэтому после перезагрузки все файлы теряются. /BOOT" – файлы загрузчика. Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящиеся в каталоге /boot/grub. "/VAR" – переменные файлы. Название каталога "/var" говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кешы, базы данных и так далее

3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе? Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount.
4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы? Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера.

Чтобы устранить повреждения файловой системы, необходимо использовать команду `fsck`.

5. Как создаётся файловая система? Файловую систему можно создать, используя команду `mkfs`
6. Дайте характеристику командам, которые позволяют просмотреть текстовые файлы. Команда `cat` читает данные из файла или стандартного ввода и выводит их на экран. Синтаксис утилиты: `cat [опции] файл1` Основные опции `cat`: `b` – нумеровать только непустые строки `-E` – показывать символ `$` в конце каждой строки `-n` – нумеровать все строки `-s` – удалять пустые повторяющиеся строки `-T` – отображать табуляции в виде `^I` `-h` – отобразить справку `-v` – версия утилиты Команда `nl`. Команда `nl` действует аналогично команде `cat`, но выводит еще номера строк в столбце слева. Команда `less`. Существенно более развитая команда для пролистывания текста. При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста. Синтаксис `less` аналогичный синтаксису команды `cat`. Некоторые опции `less`: `-g` – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения) `-N` – показывать номера строк `head`. Команда `head` выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита. Синтаксис аналогичный синтаксису команды `cat`. Основные опции `head`: `-c` (`-bytes`) – позволяет задавать количество текста не в строках, а в байтах `-n` (`-lines`) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию `-q` (`-quiet`, `-silent`) – выводит только текст, недобавляя к нему название файла `-v` (`-verbose`) – перед текстом выводит название файла `-z` (`-zero-terminated`) – символы перехода на новую строку заменяет символами завершения строк Команда `tail` позволяет выводить

заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды `cat`. Основные опции `tail`: `-c` –выводить указанное количество байт с конца файла `-f` –обновлять информацию по мере появления новых строк в файле `-n` –выводить указанное количество строк из конца файла `-pid` –используется с опцией `-f`, позволяет завершить работу утилиты, когда завершится указанный процесс `-q` –не выводить имена файлов `-retry` –повторять попытки открыть файл, если он недоступен `-v` –выводить подробную информацию о файле.

7. Приведите основные возможности команды `cp` в Linux. Команда `cp` позволяет полностью копировать файлы и директории. Синтаксис: `cp [опции] файл-источник файл-приемник`. После выполнения команды файл-источник будет полностью перенесен в файл-приемник. Если в конце указан слэш, файл будет записан в заданную директорию с оригинальным именем. Основные опции: `-attributes-only` –не копировать содержимое файла, а только флаги доступа и владельца `-f`, `-force` –перезаписывать существующие файлы `-i`, `-interactive` –спрашивать, нужно ли перезаписывать существующие файлы `-L` –копировать не символические ссылки, а то, на что они указывают `-n` –не перезаписывать существующие файлы `-P` –не следовать символическим ссылкам `-r` –копировать папку Linux рекурсивно `-s` –не выполнять копирование файлов в Linux, а создавать символические ссылки `-u` –скопировать файл, только если он был изменён `-x` –не выходить за пределы этой файловой системы `-p` –сохранять владельца, временные метки и флаги доступа при копировании `-t` –считать файл-приемник директорией и копировать файл-источник в эту директорию.
8. Назовите и дайте характеристику командам перемещения и переименования файлов и каталогов. Команда `mv` используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для



переименования файлов и директорий. Синтаксис: `mv [-опции] старый_файл новый_файл` Основные опции: `-help` – выводит на экран официальную документацию об утилите `-version` – отображает версию `mv` `-b` – создает копию файлов, которые были перемещены или перезаписаны `-f` – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла `-i` – наоборот, будет спрашивать разрешение у владельца `-n` – отключает перезапись уже существующих объектов `-strip-trailing-slashes` – удаляет завершающий символ `/` у файла при его наличии `-t [директория]` – перемещает все файлы в указанную директорию `-u` – осуществляет перемещение только в том случае, если исходный файл новее объекта назначения `-v` – отображает сведения о каждом элементе во время обработки команды Команда `rename` также предназначена, чтобы переименовать файл. Синтаксис: `rename [опции] старое_имя новое_имя_файлы`. Основные опции: `-v` – вывести список обработанных файлов `-n` – тестовый режим, на самом деле никакие действия выполнены не будут `-f` – принудительно перезаписывать существующие файлы.

9. Что такое права доступа? Как они могут быть изменены? Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации. Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Синтаксис команды: `chmod режим имя_файла`. Режим имеет следующие компоненты структуры и способ записи: `=` установить право, `-` лишить права, `+` дать право, `r` – чтение, `w` – запись, `x` – выполнение, `u` (user) – владелец файла, `g` (group) – группа, к которой принадлежит владелец файла, `o` (others) – все остальные.

## 4 Выводы

Ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.