

**Федеральное государственное автономное  
Образовательное учреждение высшего образования  
Российский Университет Дружбы Народов**

Математический университет имени Никольского  
Факультет Физико-математических и Естественных наук  
Кафедра Прикладной математики и информатики

Отчет по лабораторной работе № 5  
“Анализ файловой системы Linux. Команды для работы с файлами и каталогами”

Выполнил:  
Студент группы НПИМбв-01-10  
Адхамова Луиза Шухратовна

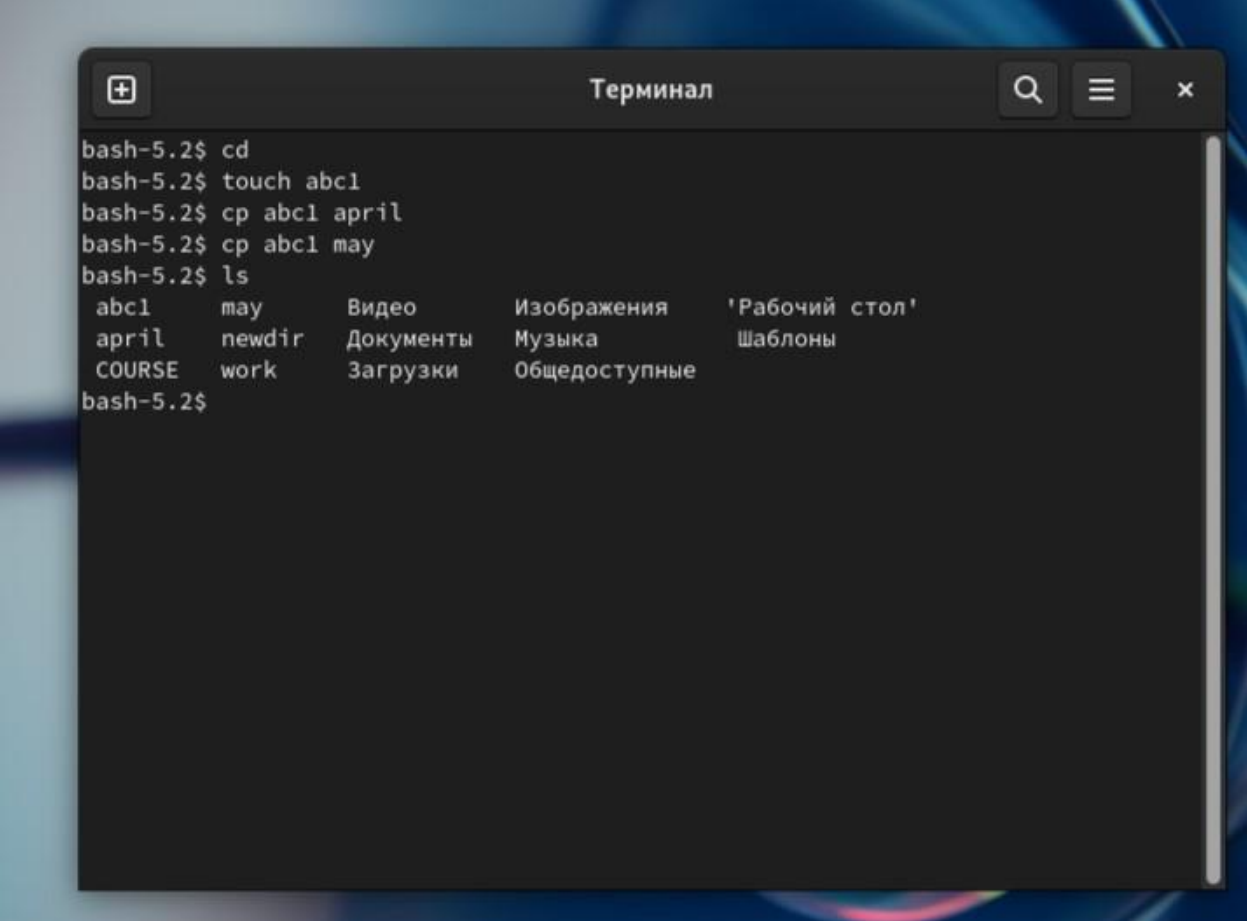
Москва  
2024 год

## 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.

## Выполнение:

1. Выполним все примеры, приведённые в первой части описания лабораторной работы:
  - 1.1. скопируем файл ~/abc1 в файл april и в файл may (рис. 1.1)



```
bash-5.2$ cd
bash-5.2$ touch abc1
bash-5.2$ cp abc1 april
bash-5.2$ cp abc1 may
bash-5.2$ ls
abc1      may      Видео    Изображения  'Рабочий стол'
april     newdir   Документы  Музыка        Шаблоны
COURSE    work     Загрузки  Общедоступные
bash-5.2$
```

Рисунок 1.1. Копирование файла abc1 в папки may и april.

- 1.2. Скопируем файлы april и may в каталог monthly (рис. 1.2)

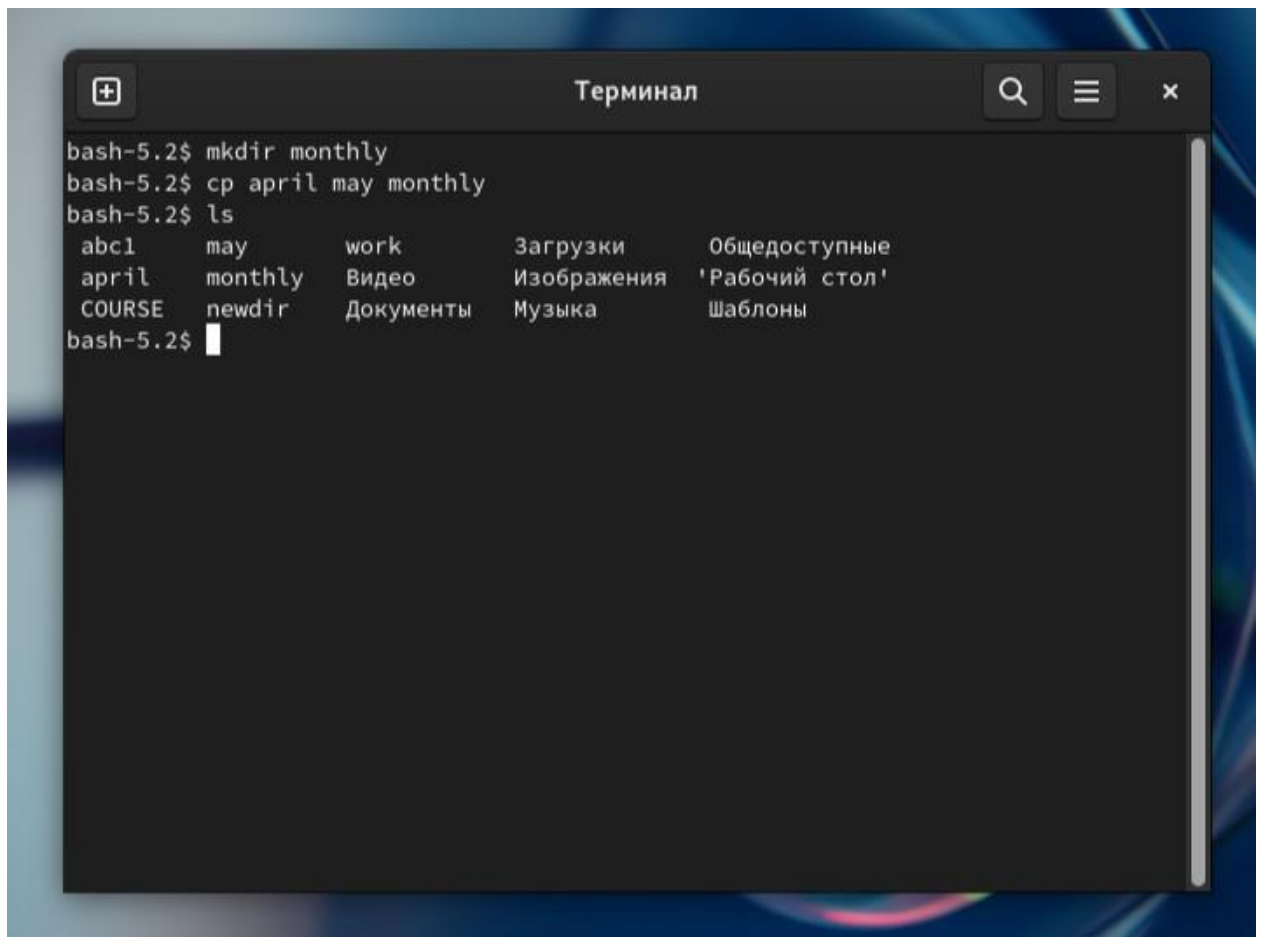
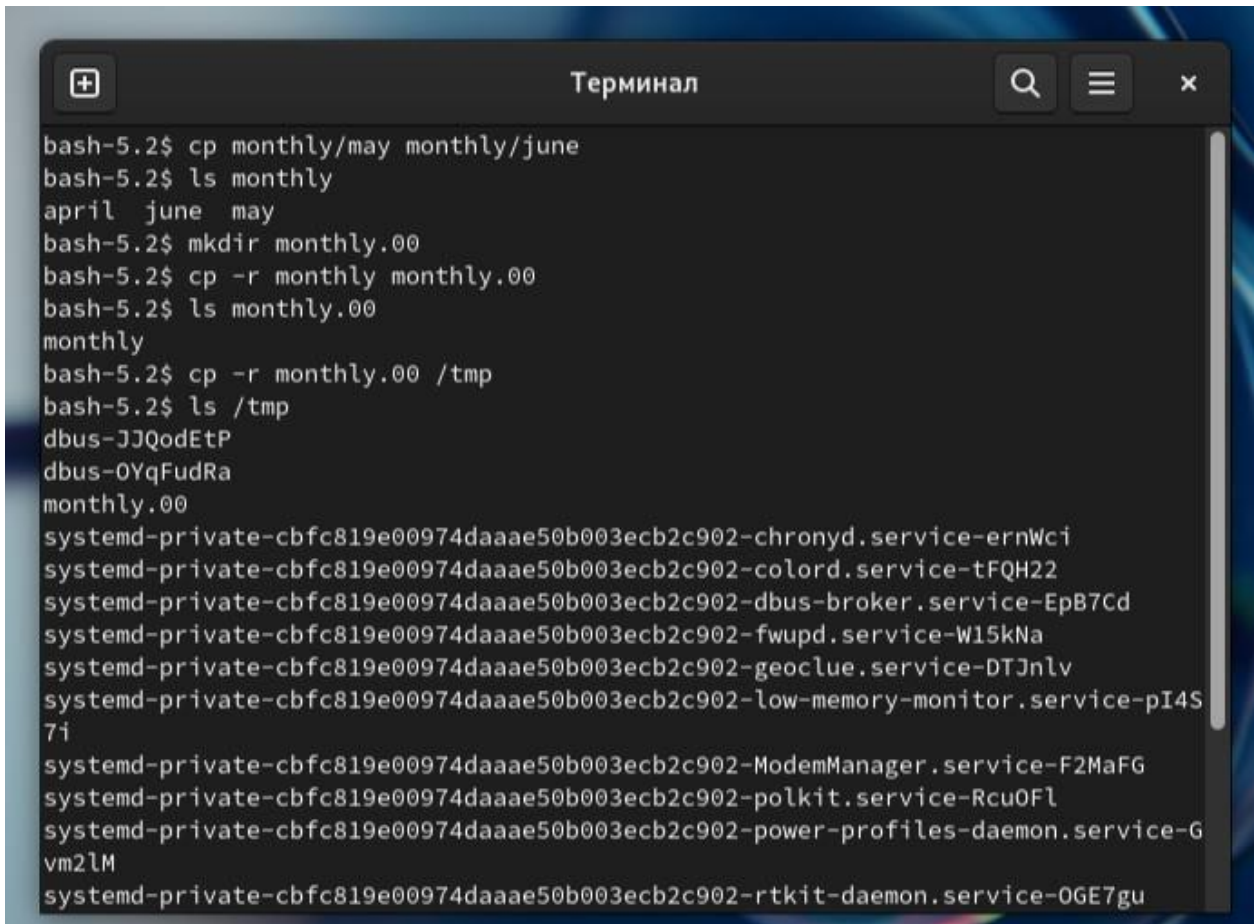


Рисунок 1.2. Создание папки monthly.

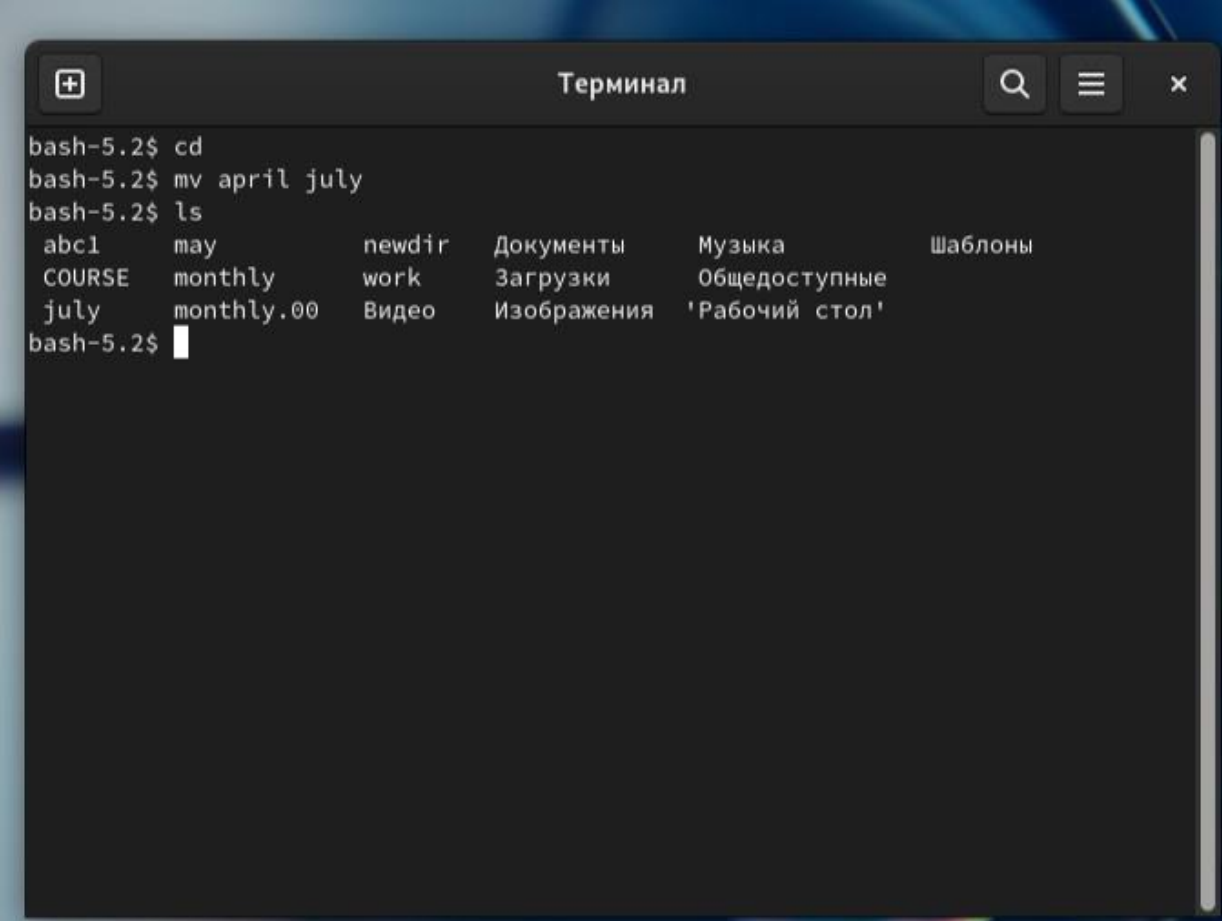
- 1.3. Скопируем файл `monthly/may` в файл с именем `june`. Также скопируем каталог `monthly` в каталог `monthly.00`, скопируем каталог `monthly.00` в каталог `/tmp` (рис. 1.3)



```
bash-5.2$ cp monthly/may monthly/june
bash-5.2$ ls monthly
april  june  may
bash-5.2$ mkdir monthly.00
bash-5.2$ cp -r monthly monthly.00
bash-5.2$ ls monthly.00
monthly
bash-5.2$ cp -r monthly.00 /tmp
bash-5.2$ ls /tmp
dbus-JJQodEtP
dbus-0YqFudRa
monthly.00
systemd-private-cbfc819e00974daaae50b003ecb2c902-chronyd.service-ernWci
systemd-private-cbfc819e00974daaae50b003ecb2c902-colord.service-tFQH22
systemd-private-cbfc819e00974daaae50b003ecb2c902-dbus-broker.service-EpB7Cd
systemd-private-cbfc819e00974daaae50b003ecb2c902-fwupd.service-W15kNa
systemd-private-cbfc819e00974daaae50b003ecb2c902-geoclue.service-DTJnlv
systemd-private-cbfc819e00974daaae50b003ecb2c902-low-memory-monitor.service-pI4S
7i
systemd-private-cbfc819e00974daaae50b003ecb2c902-ModemManager.service-F2MaFG
systemd-private-cbfc819e00974daaae50b003ecb2c902-polkit.service-Rcu0Fl
systemd-private-cbfc819e00974daaae50b003ecb2c902-power-profiles-daemon.service-G
vm2lM
systemd-private-cbfc819e00974daaae50b003ecb2c902-rtkit-daemon.service-0GE7gu
```

Рисунок 1.3. Копирование файла `monthly/may` в файл с именем `june`, копирование каталога `monthly` в каталог `monthly.00`, копирование каталога `monthly.00` в каталог `/tmp`

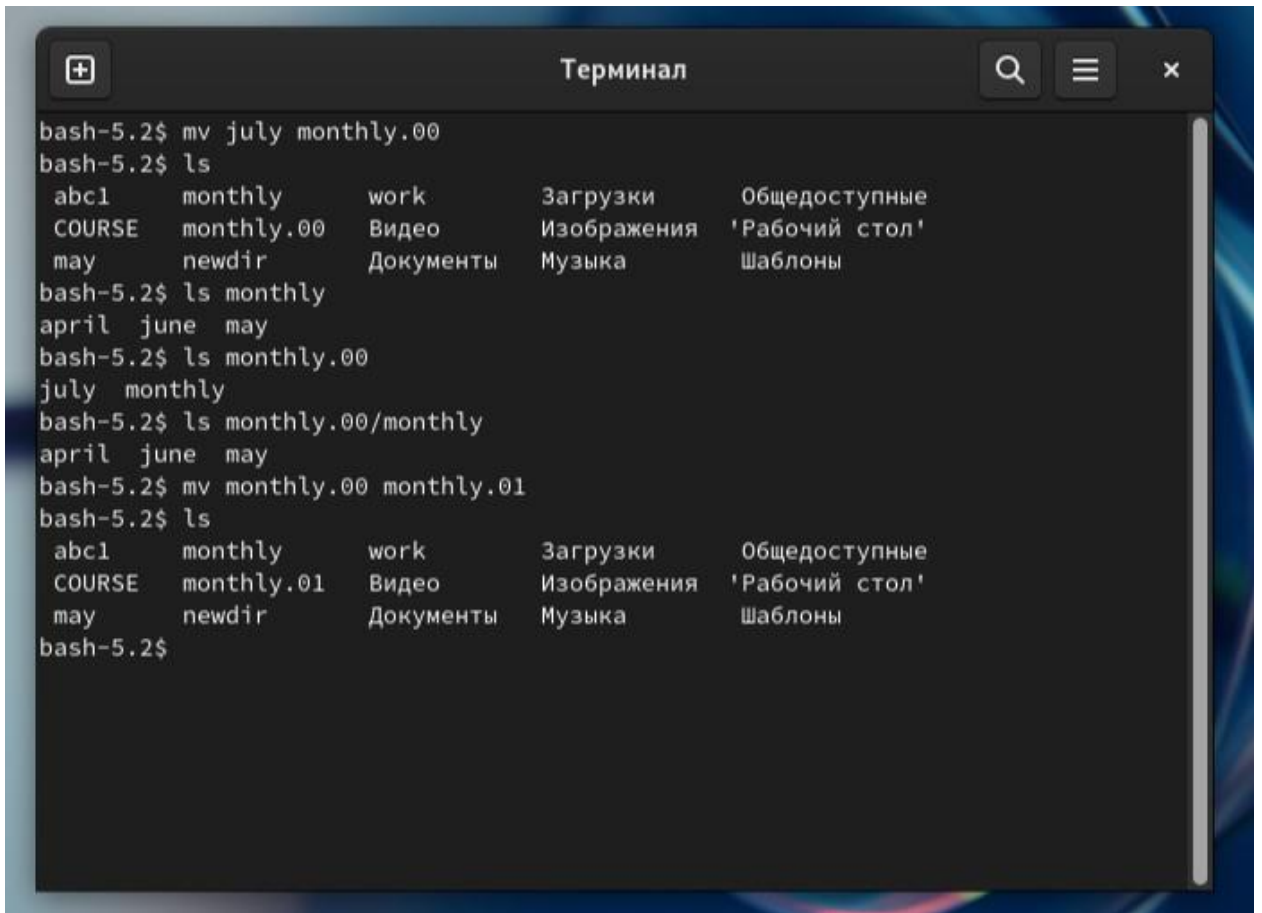
1.4. Изменим название файла `april` на `july` в домашнем каталоге. Переместим файл `july` в каталог `monthly.00` (рис 1.4)



```
bash-5.2$ cd
bash-5.2$ mv april july
bash-5.2$ ls
abc1      may      newdir   Документы  Музыка      Шаблоны
COURSE    monthly  work     Загрузки   Общедоступные
july      monthly.00  Видео    Изображения  'Рабочий стол'
```

Рисунок 1.4. Изменение названия файла `april` на `july` в домашнем каталоге.

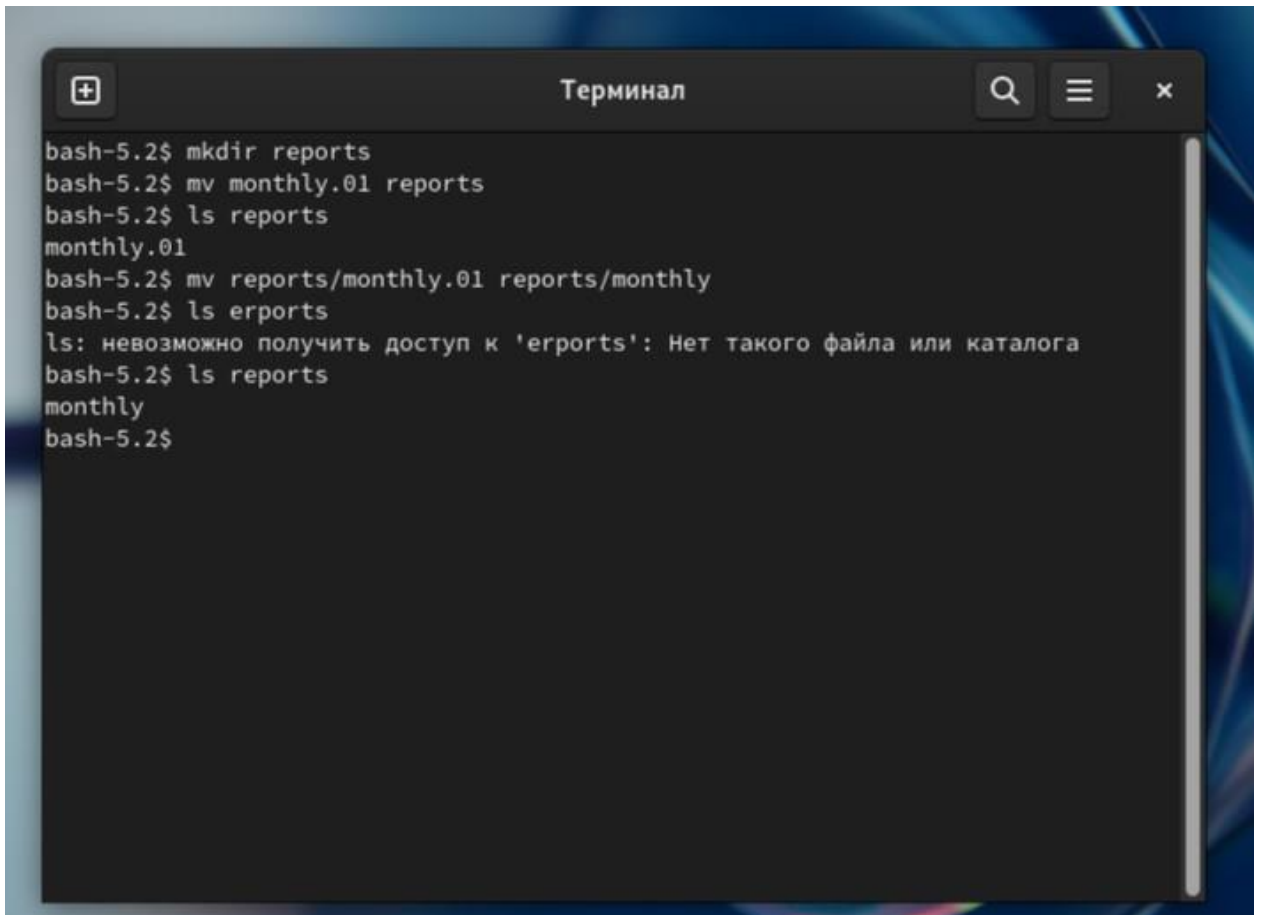
- 1.5. Переместим каталог `monthly.00` в `monthly.01`. Переименуем каталог `reports/monthly.01` в `reports/monthly` (рис. 1.5).



```
bash-5.2$ mv july monthly.00
bash-5.2$ ls
abc1      monthly   work      Загрузки  Общедоступные
COURSE    monthly.00 Видео     Изображения 'Рабочий стол'
may       newdir    Документы Музыка     Шаблоны
bash-5.2$ ls monthly
april  june  may
bash-5.2$ ls monthly.00
july  monthly
bash-5.2$ ls monthly.00/monthly
april  june  may
bash-5.2$ mv monthly.00 monthly.01
bash-5.2$ ls
abc1      monthly   work      Загрузки  Общедоступные
COURSE    monthly.01 Видео     Изображения 'Рабочий стол'
may       newdir    Документы Музыка     Шаблоны
bash-5.2$
```

Рисунок 1.5. Перемещение файла july в каталог monthly.00, переименование каталога monthly.00 в monthly.01

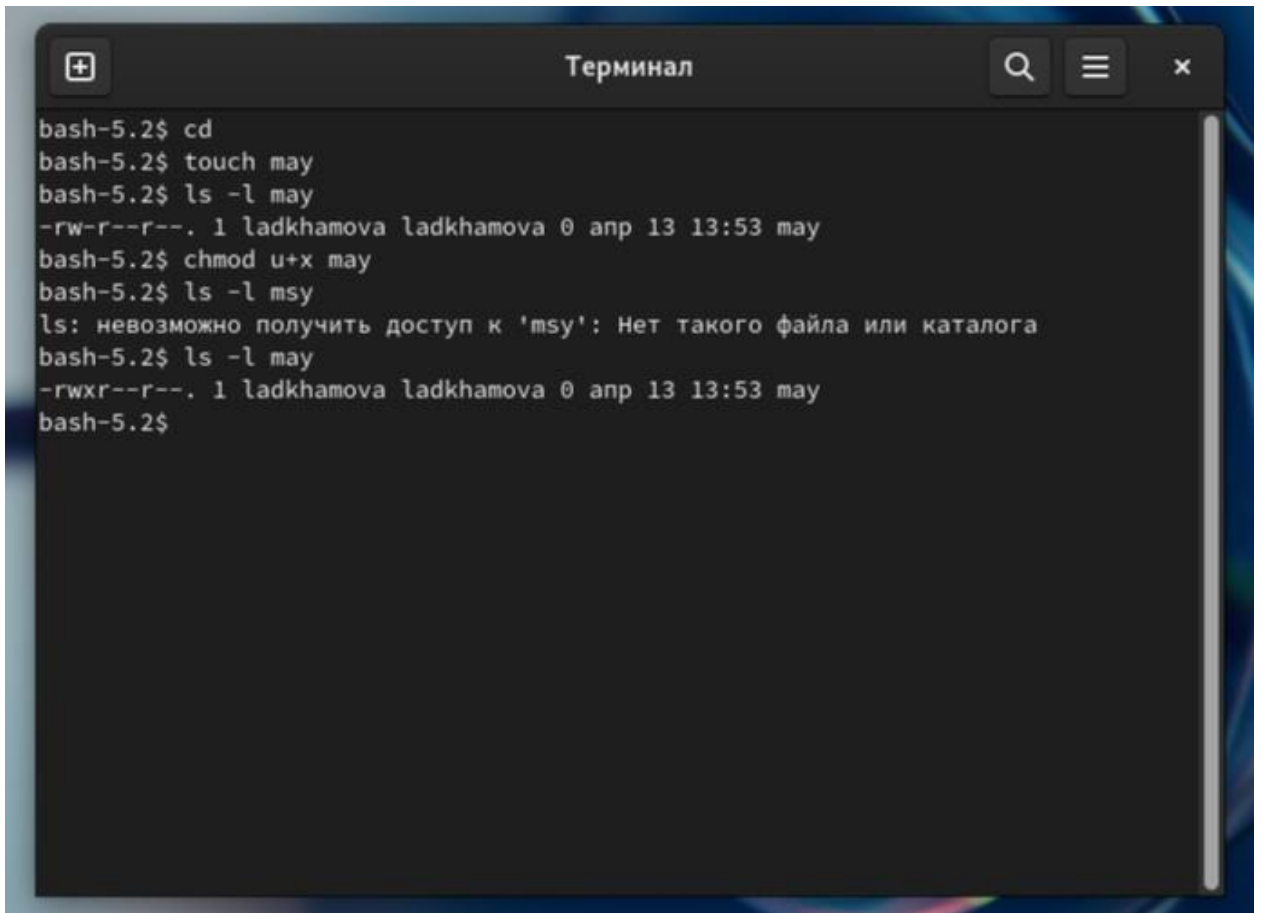
1.6. Переместим каталог monthly.01 в каталог reports. И переименуем monthly.01 в monthly (рис. 1.6).

A terminal window titled "Терминал" (Terminal) with standard macOS window controls (plus, search, menu, close). The terminal shows a series of commands and their outputs in a bash shell. The commands performed are: creating a directory 'reports', moving 'monthly.01' into it, listing its contents, moving 'reports/monthly.01' to 'reports/monthly', attempting to list 'erports' (which fails with an error), and finally listing 'reports' to confirm the new directory structure.

```
bash-5.2$ mkdir reports
bash-5.2$ mv monthly.01 reports
bash-5.2$ ls reports
monthly.01
bash-5.2$ mv reports/monthly.01 reports/monthly
bash-5.2$ ls erports
ls: невозможно получить доступ к 'erports': Нет такого файла или каталога
bash-5.2$ ls reports
monthly
bash-5.2$
```

Рисунок 1.6. Перемещение каталога monthly.01 в каталог reports, переименование каталога reports/monthly.01 в reports/monthly

1.7. Создадим файл ~/may с правом выполнения для владельца (рис. 1.7)

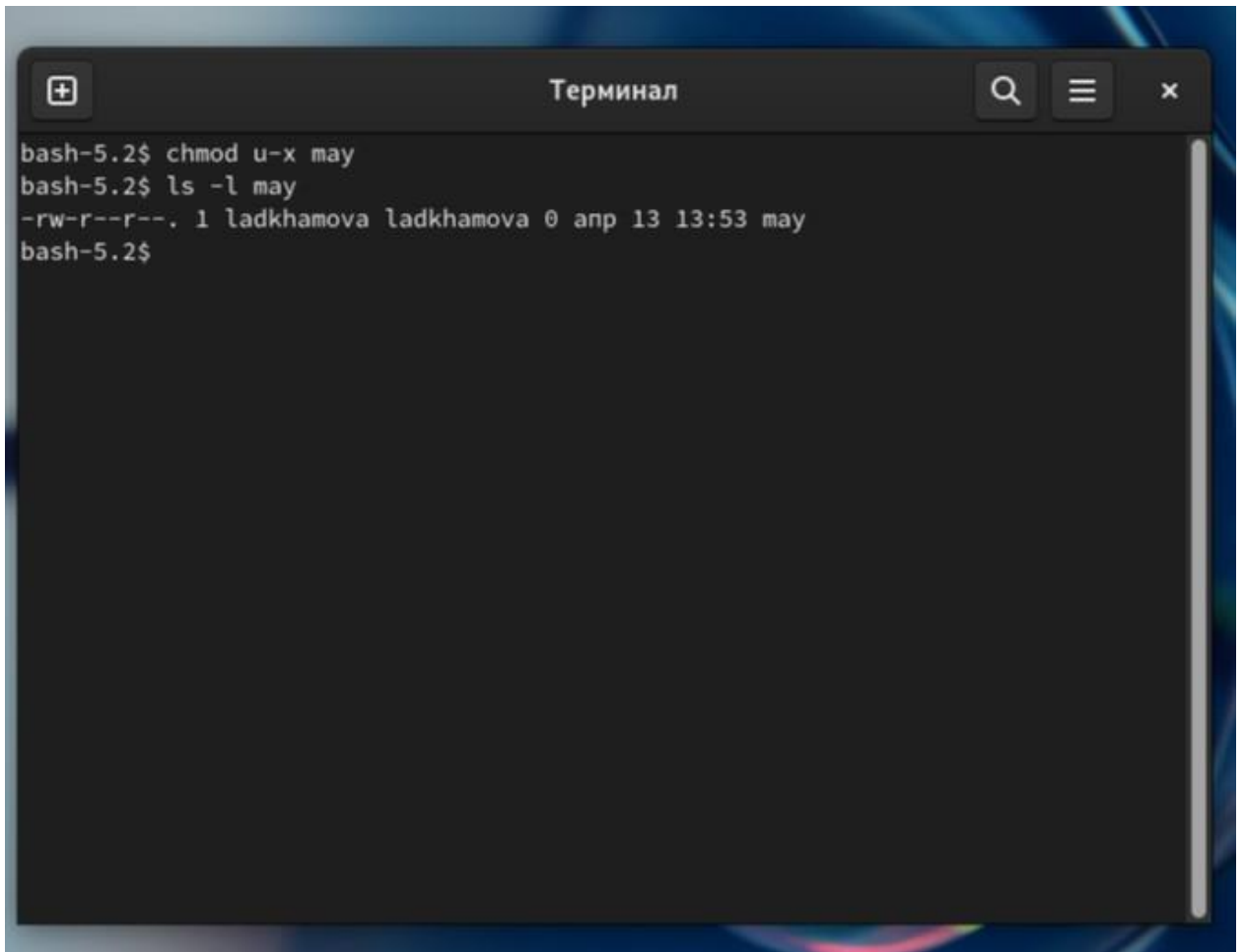
A screenshot of a macOS Terminal window titled "Терминал". The window has a dark background and a light gray title bar with standard macOS window controls (a plus icon, a search icon, a menu icon, and a close icon). The terminal shows a series of commands and their outputs. The user starts by running 'cd', then 'touch may', and 'ls -l may'. The output of 'ls -l may' shows a file named 'may' with permissions '-rw-r--r--', owned by 'ladkhamova', with a size of '1', and a timestamp of 'анр 13 13:53'. The user then runs 'chmod u+x may'. Next, they run 'ls -l msy', which results in an error message: 'ls: невозможно получить доступ к 'msy': Нет такого файла или каталога'. Finally, they run 'ls -l may' again, and the output shows the file 'may' now has permissions '-rwxr--r--'. The prompt 'bash-5.2\$' is visible at the end of the last command.

```
bash-5.2$ cd
bash-5.2$ touch may
bash-5.2$ ls -l may
-rw-r--r--. 1 ladkhamova ladkhamova 0 анр 13 13:53 may
bash-5.2$ chmod u+x may
bash-5.2$ ls -l msy
ls: невозможно получить доступ к 'msy': Нет такого файла или каталога
bash-5.2$ ls -l may
-rwxr--r--. 1 ladkhamova ladkhamova 0 анр 13 13:53 may
bash-5.2$
```

Рисунок 1.7. Требуется создать файл ~/may с правом выполнения для владельца.

1.8. А затем лишим владельца файла ~/may на выполнение (рис. 1.8)

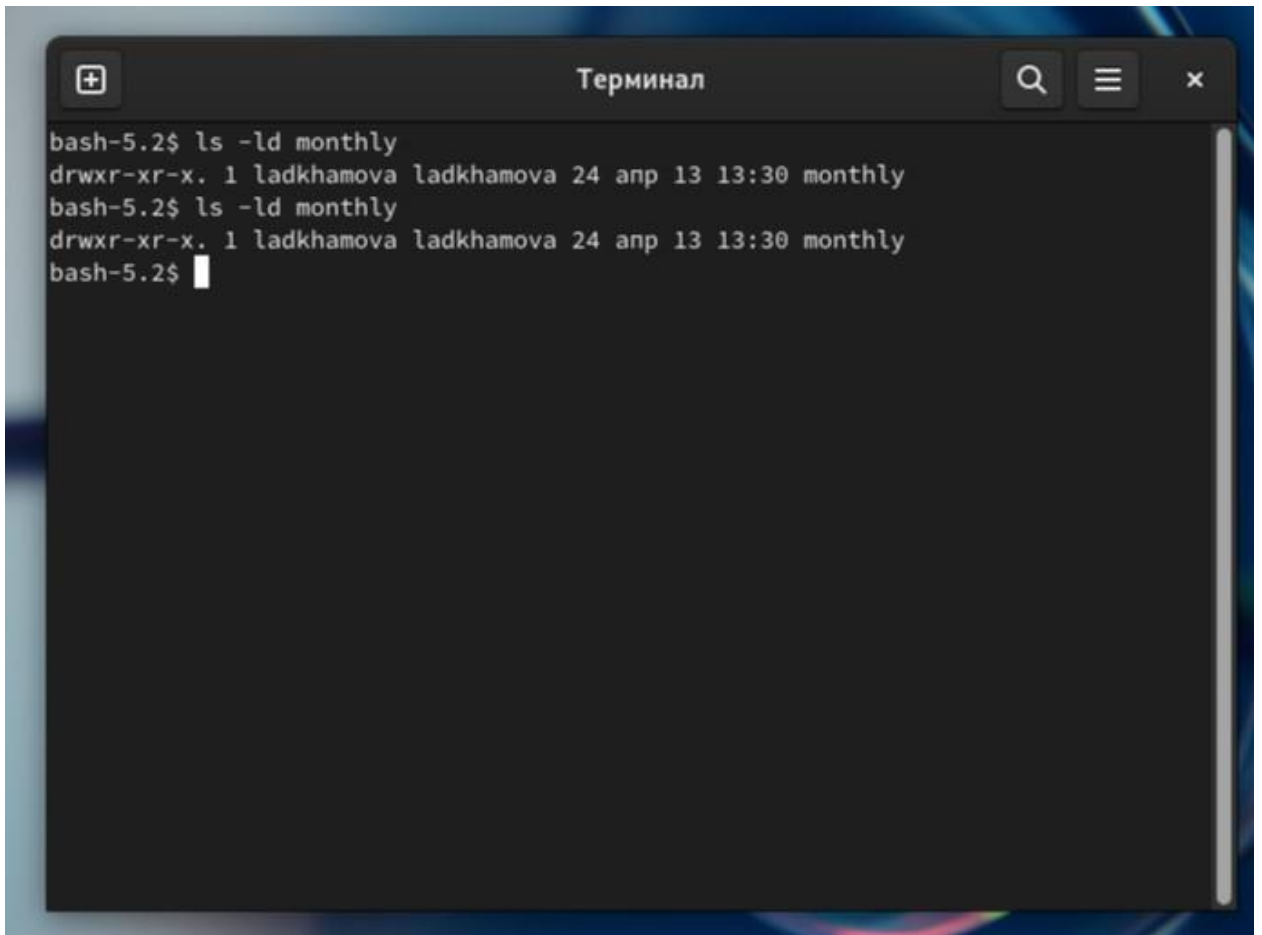




```
bash-5.2$ chmod u-x may
bash-5.2$ ls -l may
-rw-r--r--. 1 ladkhamova ladkhamova 0 apr 13 13:53 may
bash-5.2$
```

Рисунок 1.8. Требуется лишить владельца файла ~/may права на выполнение.

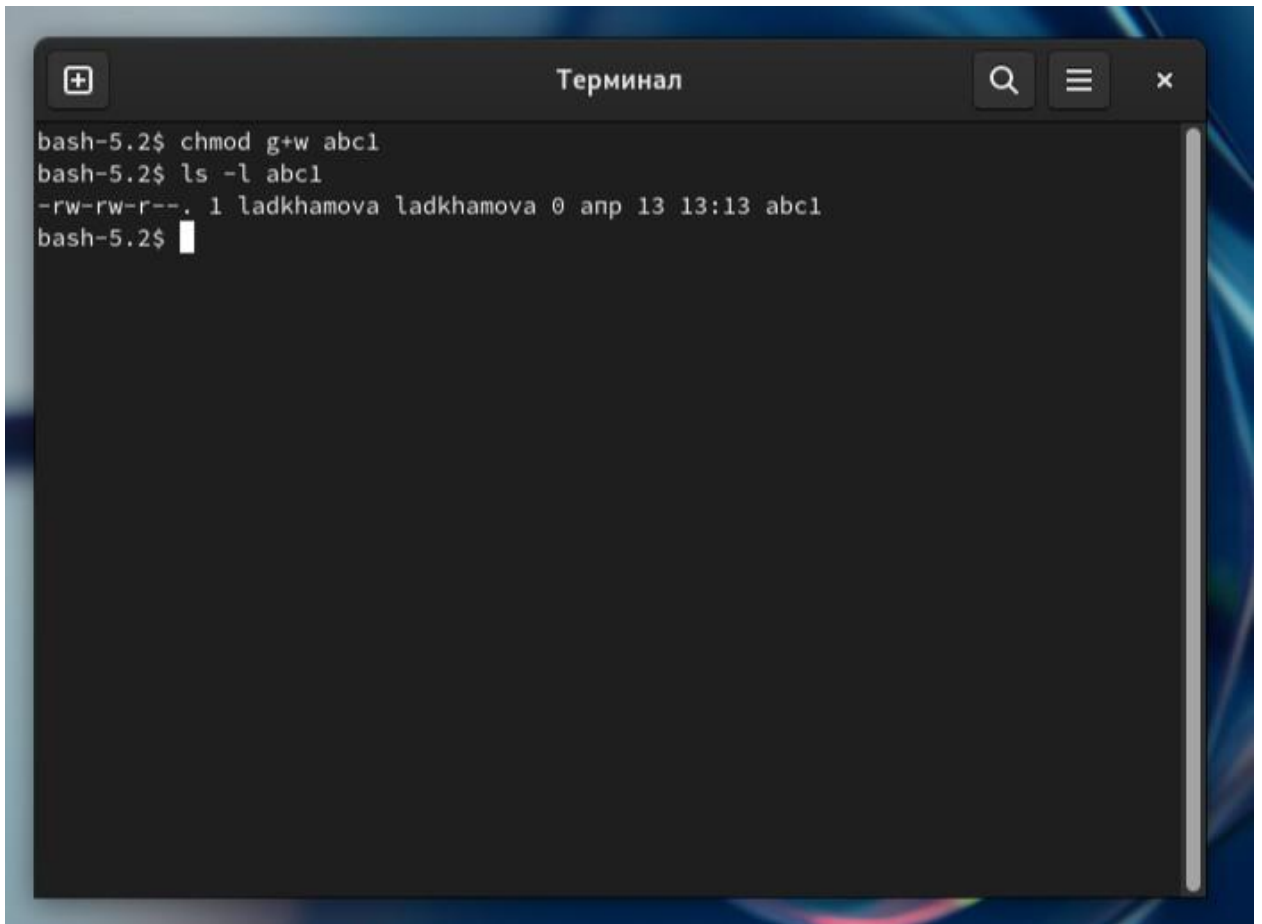
1.9. Создадим файл monthly с запретом на чтение для членов группы и всех остальных пользователей (рис 1.9):

A terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the title bar. The terminal shows the following commands and output:

```
bash-5.2$ ls -ld monthly
drwxr-xr-x. 1 ladkhamova ladkhamova 24 апр 13 13:30 monthly
bash-5.2$ ls -ld monthly
drwxr-xr-x. 1 ladkhamova ladkhamova 24 апр 13 13:30 monthly
bash-5.2$
```

Рисунок 1.9. Требуется создать каталог `monthly` с запретом на чтение для членов группы и всех остальных пользователей

1.10. Создадим файл `~/abc1` с правом записи для членов группы (рис. 1.10)

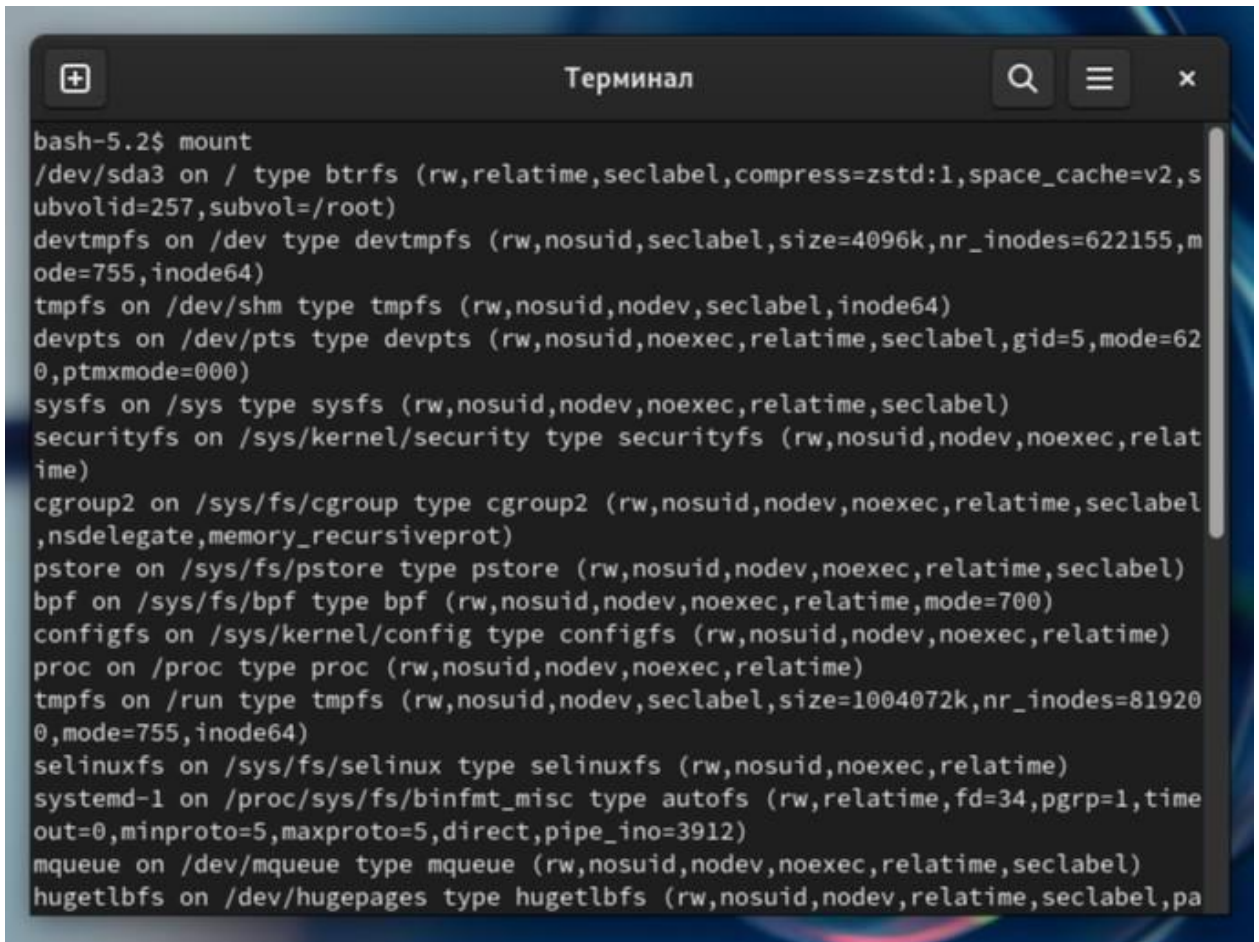
A terminal window titled "Терминал" (Terminal) with a dark background. The window contains the following text:

```
bash-5.2$ chmod g+w abc1
bash-5.2$ ls -l abc1
-rw-rw-r--. 1 ladkhamova ladkhamova 0 апр 13 13:13 abc1
bash-5.2$
```

The terminal window has a standard macOS-style title bar with a close button (X), a menu button (three horizontal lines), and a search button (magnifying glass). The background of the desktop is a blue and purple abstract pattern.

Рисунок 1.10. Требуется создать файл ~/abc1 с правом записи для членов группы

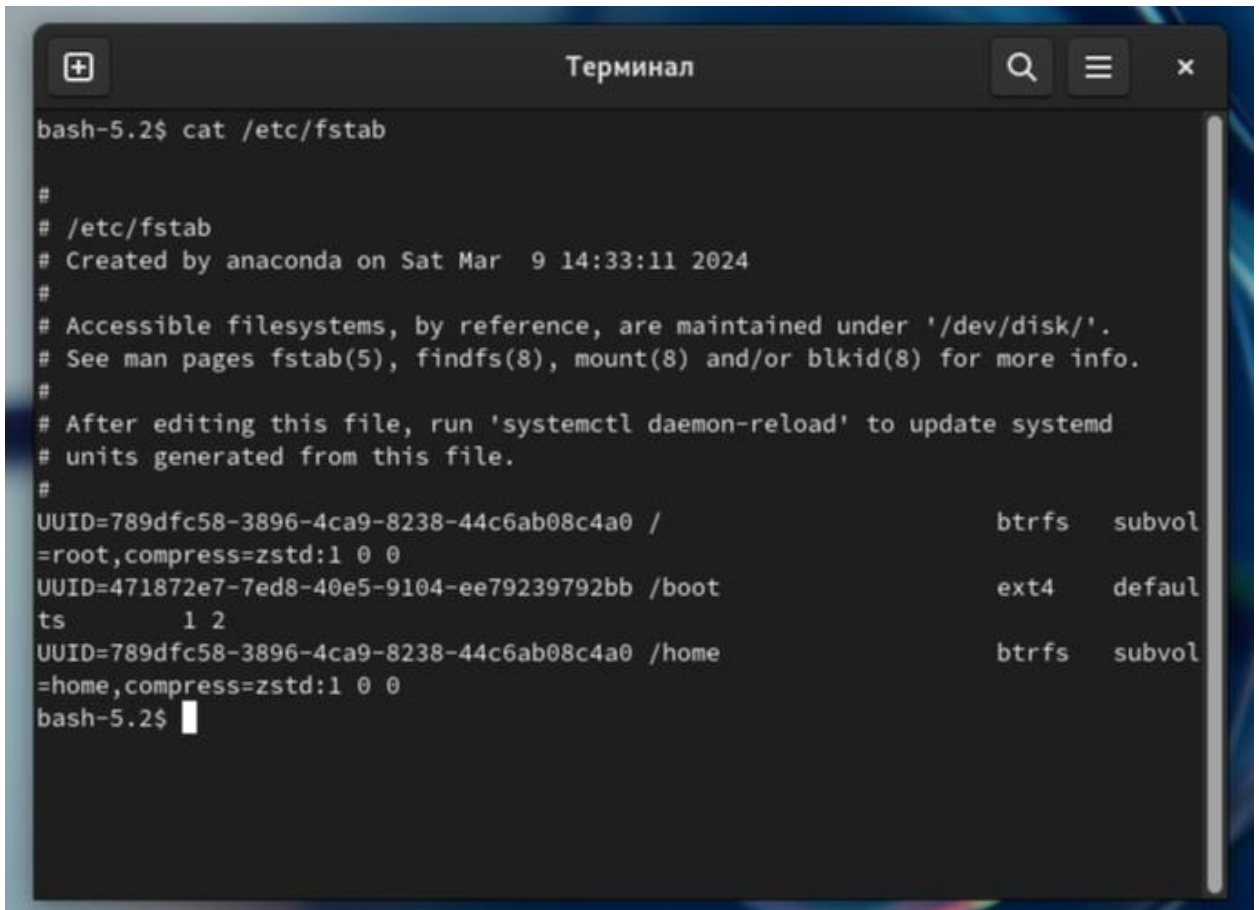
1.11. Используем команду mount для просмотра в операционной системе файловых систем (рис. 1.11)

A terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the title bar. The terminal shows the output of the 'mount' command, listing various mounted filesystems and their options. The output is as follows:

```
bash-5.2$ mount
/dev/sda3 on / type btrfs (rw,relatime,seclabel,compress=zstd:1,space_cache=v2,subvol=257,subvol=/root)
devtmpfs on /dev type devtmpfs (rw,nosuid,seclabel,size=4096k,nr_inodes=622155,mode=755,inode64)
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)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
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)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
configfs on /sys/kernel/config type configfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
tmpfs on /run type tmpfs (rw,nosuid,nodev,seclabel,size=1004072k,nr_inodes=81920,mode=755,inode64)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,nosuid,noexec,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=34,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=3912)
mqueue on /dev/mqueue type mqueue (rw,nosuid,nodev,noexec,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,nosuid,nodev,relatime,seclabel,pa
```

Рисунок 1.11. Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой `mount` без параметров

1.12. Рассмотрим другой способ просмотра файлов: `cat /etc/fstab` (рис. 1.12)

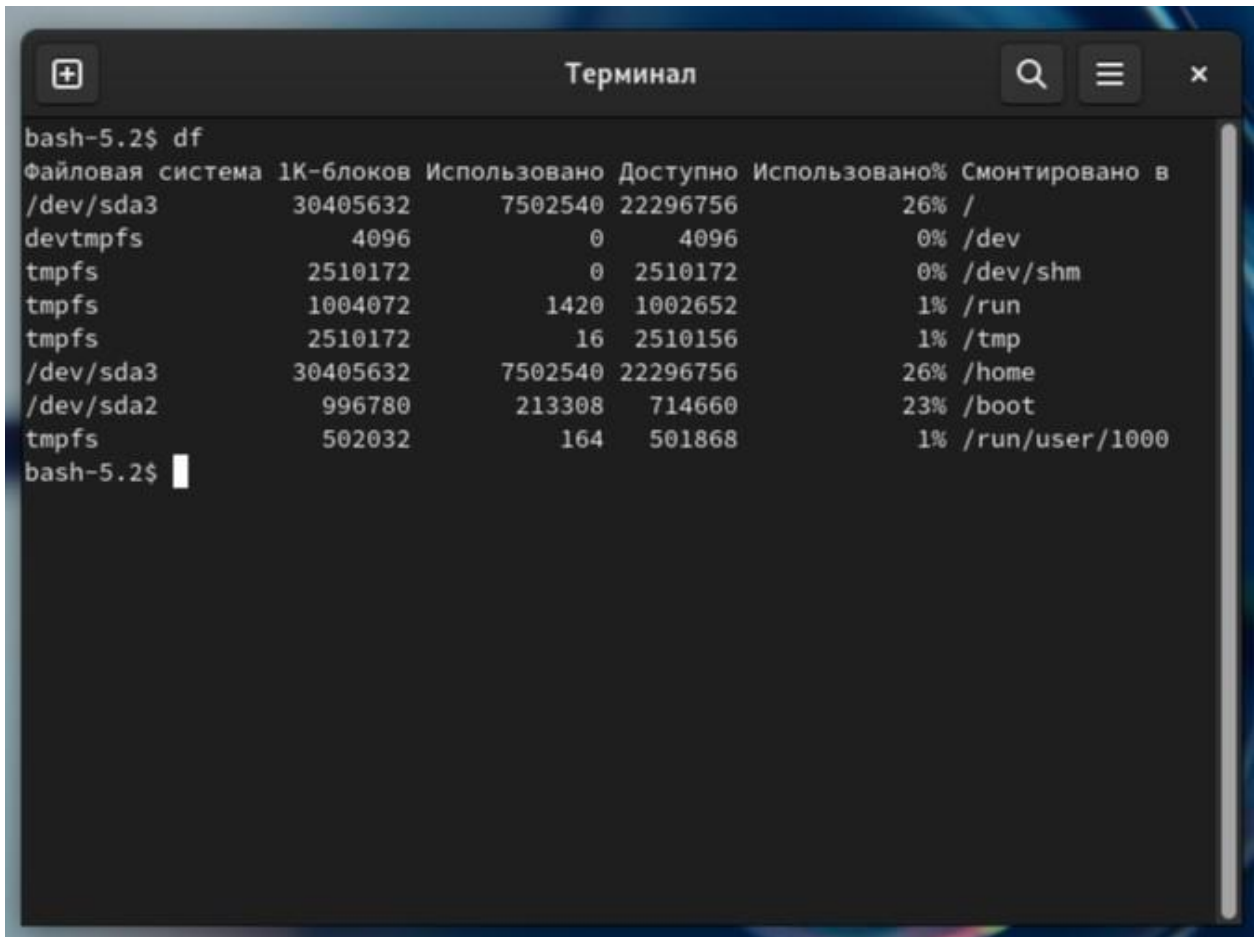
A screenshot of a terminal window titled "Терминал" (Terminal). The window has a dark background and a light-colored text. The terminal shows the command "cat /etc/fstab" being executed. The output displays the contents of the /etc/fstab file, which includes comments and three entries for filesystems: root (btrfs, subvol), boot (ext4, default), and home (btrfs, subvol). The terminal prompt is "bash-5.2\$".

```
bash-5.2$ cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sat Mar  9 14:33:11 2024
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=789dfc58-3896-4ca9-8238-44c6ab08c4a0 /                    btrfs  subvol
=root,compress=zstd:1 0 0
UUID=471872e7-7ed8-40e5-9104-ee79239792bb /boot                ext4    default
ts          1 2
UUID=789dfc58-3896-4ca9-8238-44c6ab08c4a0 /home                btrfs  subvol
=home,compress=zstd:1 0 0
bash-5.2$
```

Рисунок 1.12. Другой способ определения смонтированных в операционной системе файловых систем — просмотр файла /etc/fstab

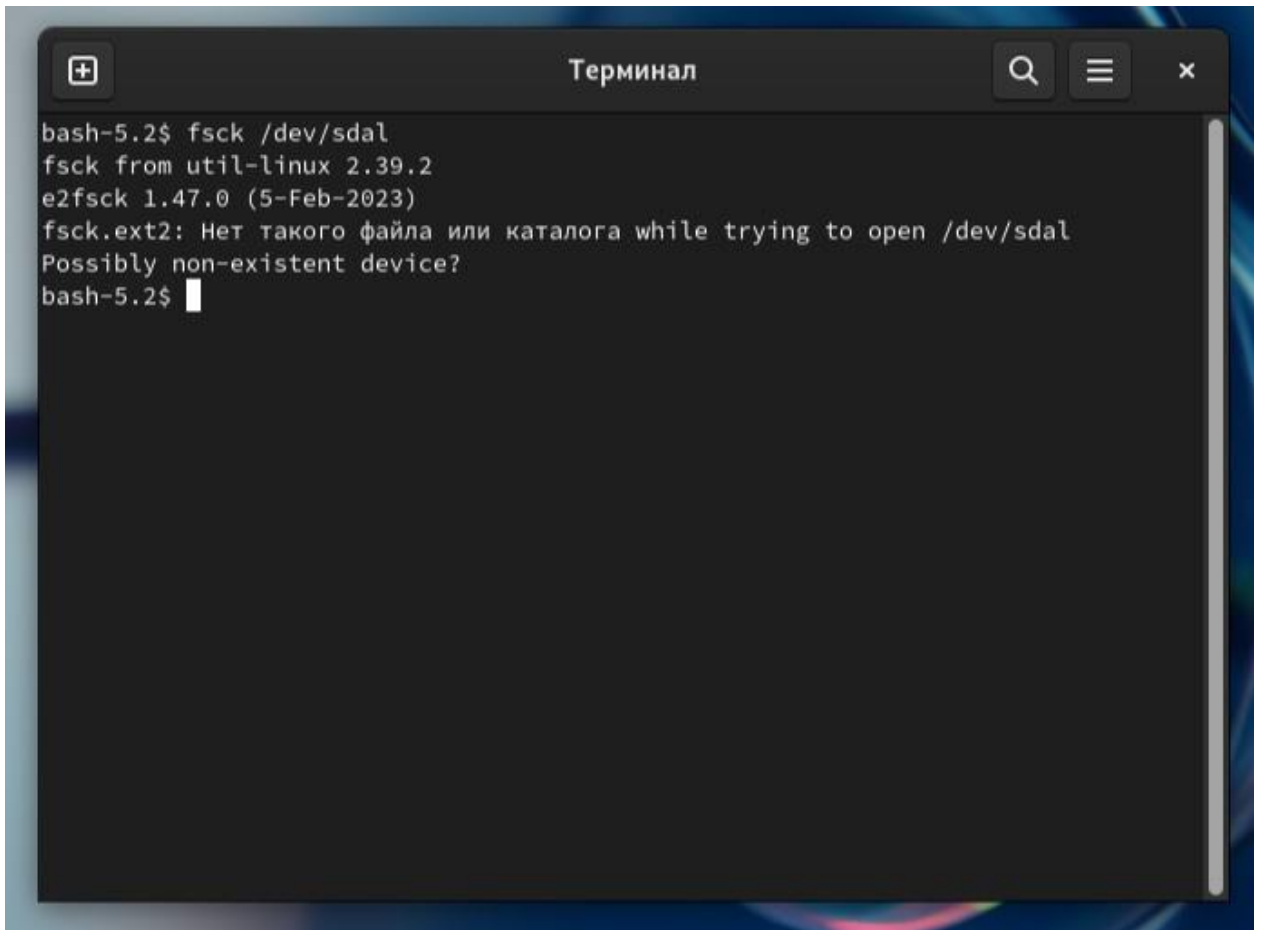
1.13. Воспользуемся командой `df` для определения объема свободного пространства на файловой системе (рис. 1.13)

A terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the top right corner. The terminal shows the command "bash-5.2\$ df" followed by its output. The output is a table with columns: "Файловая система" (File system), "1K-блоков" (1K blocks), "Использовано" (Used), "Доступно" (Available), "Использовано%" (Used %), and "Смонтировано в" (Mounted in). The data rows are: "/dev/sda3" (30405632, 7502540, 22296756, 26%, "/"), "devtmpfs" (4096, 0, 4096, 0%, "/dev"), "tmpfs" (2510172, 0, 2510172, 0%, "/dev/shm"), "tmpfs" (1004072, 1420, 1002652, 1%, "/run"), "tmpfs" (2510172, 16, 2510156, 1%, "/tmp"), "/dev/sda3" (30405632, 7502540, 22296756, 26%, "/home"), "/dev/sda2" (996780, 213308, 714660, 23%, "/boot"), and "tmpfs" (502032, 164, 501868, 1%, "/run/user/1000"). The prompt "bash-5.2\$" is shown at the bottom with a cursor.

```
bash-5.2$ df
Файловая система 1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
/dev/sda3         30405632      7502540  22296756         26% /
devtmpfs           4096           0      4096           0% /dev
tmpfs             2510172           0  2510172           0% /dev/shm
tmpfs             1004072        1420  1002652           1% /run
tmpfs             2510172          16  2510156           1% /tmp
/dev/sda3         30405632      7502540  22296756         26% /home
/dev/sda2         996780        213308   714660          23% /boot
tmpfs             502032          164   501868           1% /run/user/1000
bash-5.2$
```

Рисунок 1.13. Для определения объема 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

1.14. Используем команду `fsck` для проверки целостности файловой системы (рис. 1.14)



```
bash-5.2$ fsck /dev/sda1
fsck from util-linux 2.39.2
e2fsck 1.47.0 (5-Feb-2023)
fsck.ext2: Нет такого файла или каталога while trying to open /dev/sda1
Possibly non-existent device?
bash-5.2$
```

Рисунок 1.14. С помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы

2. Выполним следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
  - 2.1. Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовем его `equipment`. Если файла `io.h` нет, то используем любой другой файл в каталоге `/usr/include/sys/` вместо него (рис. 2.1):

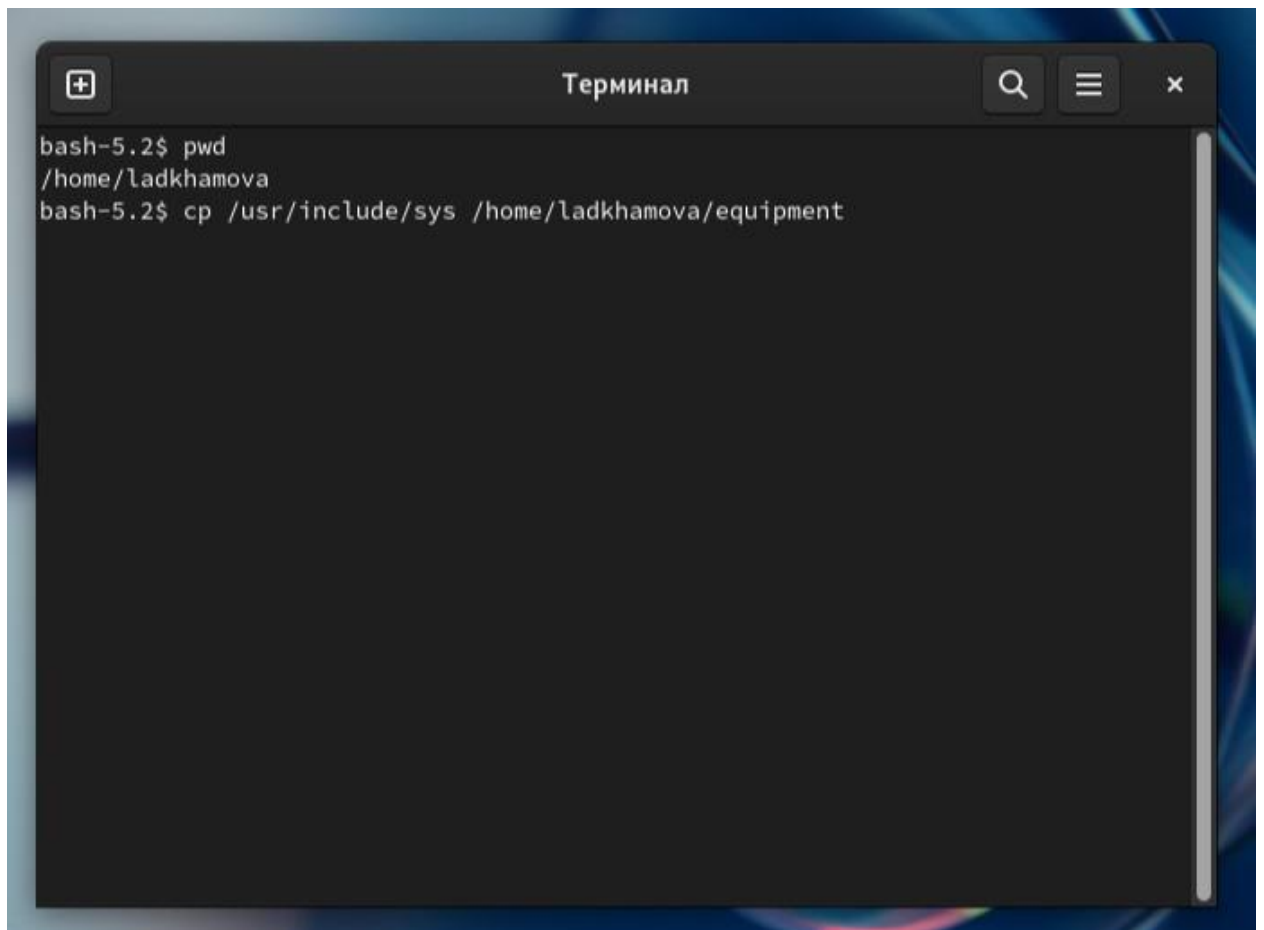


Рисунок 2.1. Копирование файла.

2.2. В домашнем каталоге создадим директорию ~/ski.places (рис. 2.2):



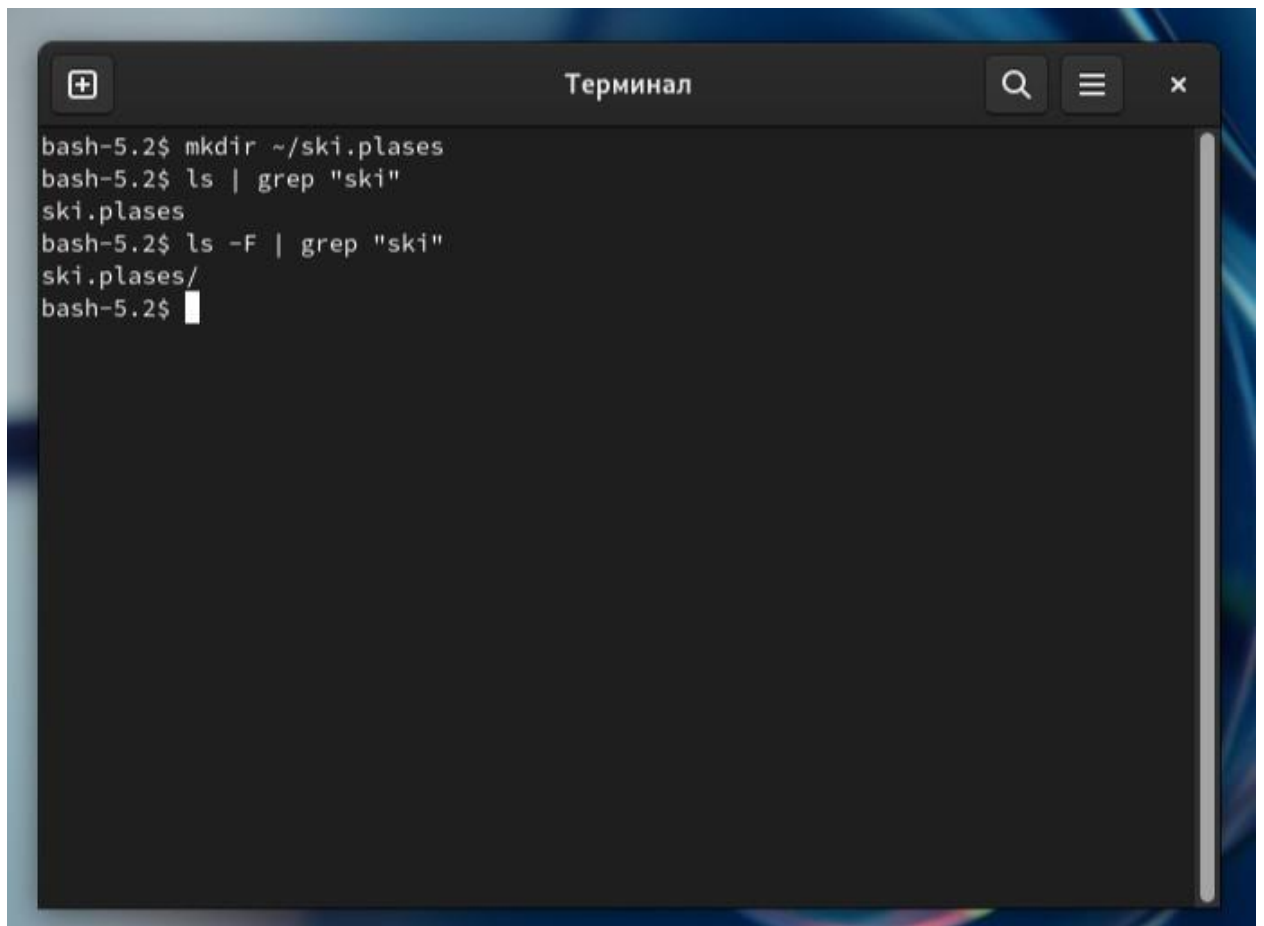


Рисунок 2.2. Создание каталога.

2.3. Переместим файл equipment в каталог ~/ski.places (рис. 2.3):

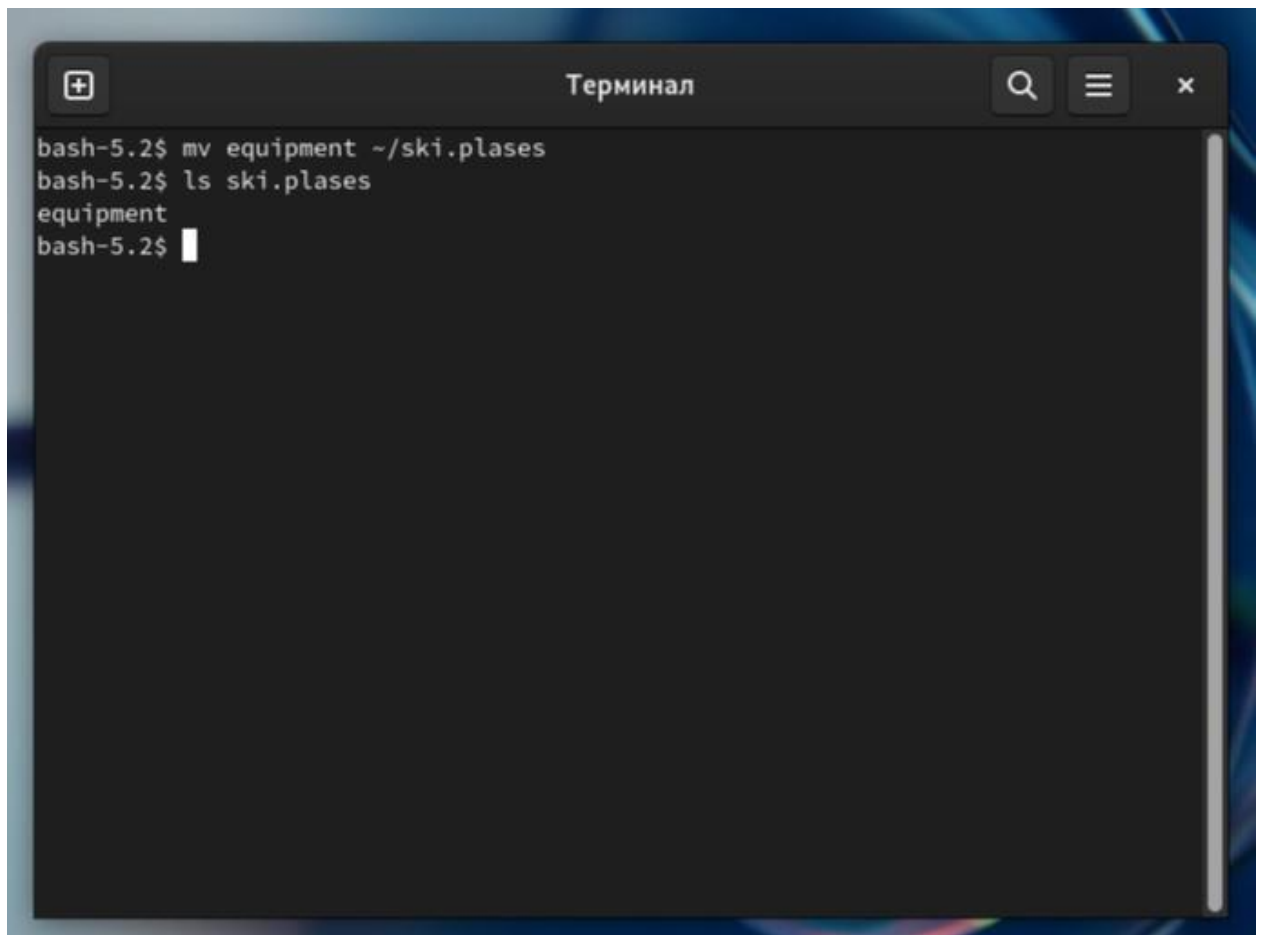


Рисунок 2.3. Перемещение файла.

2.4. Переименуем файл `~/ski.plases/equipment` в `~/ski.plases/equiplist` (рис. 2.4):

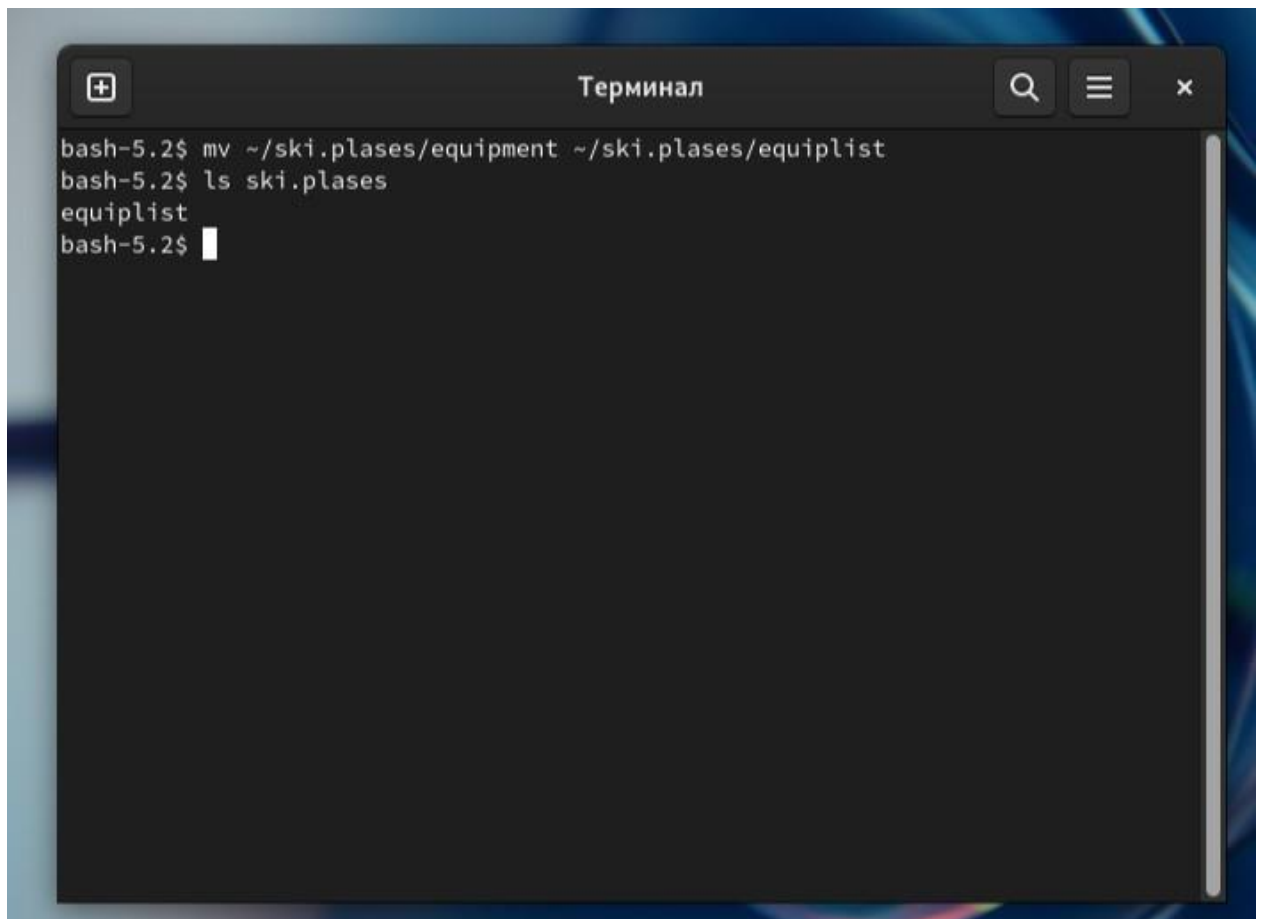
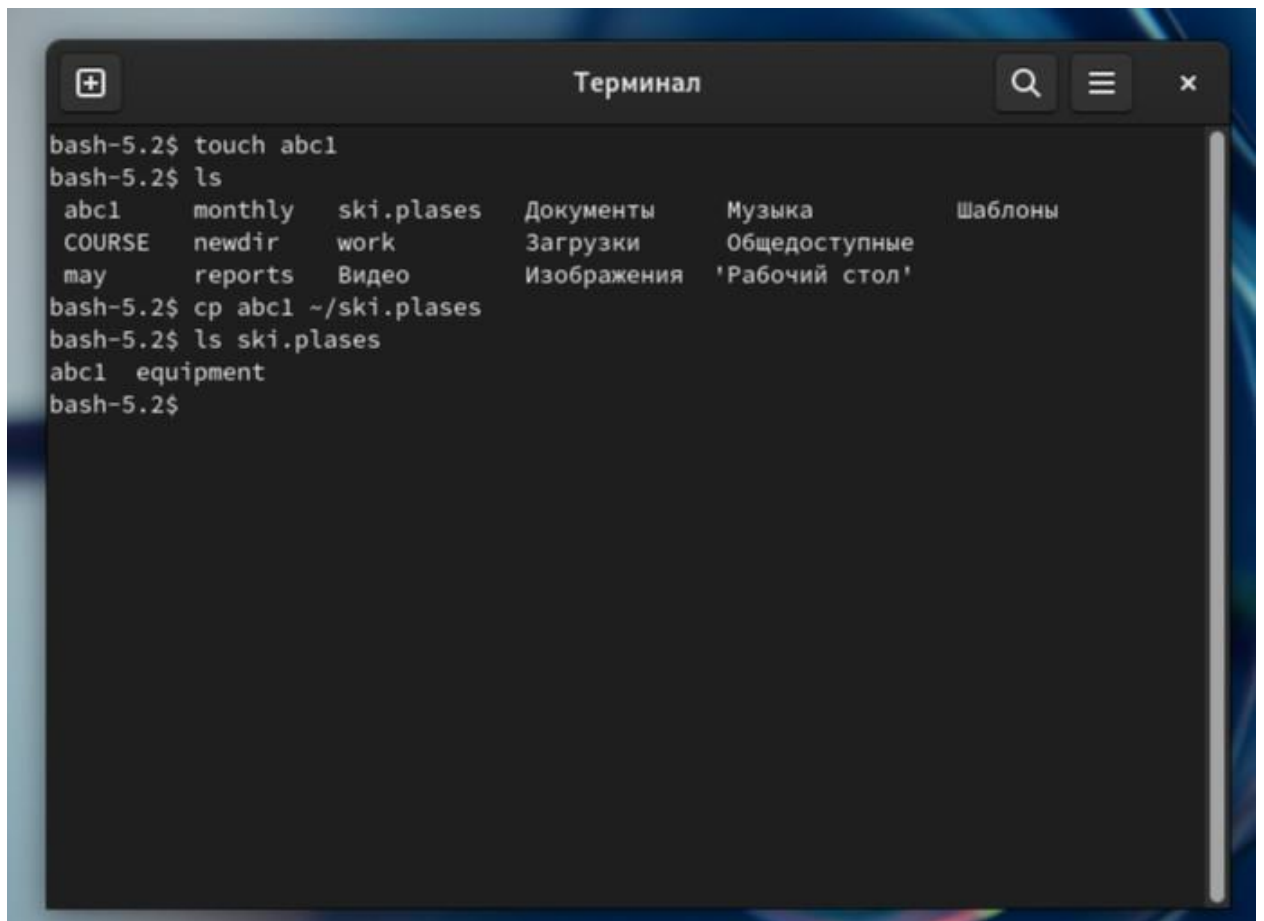


Рисунок 2.4. Переименование файла.

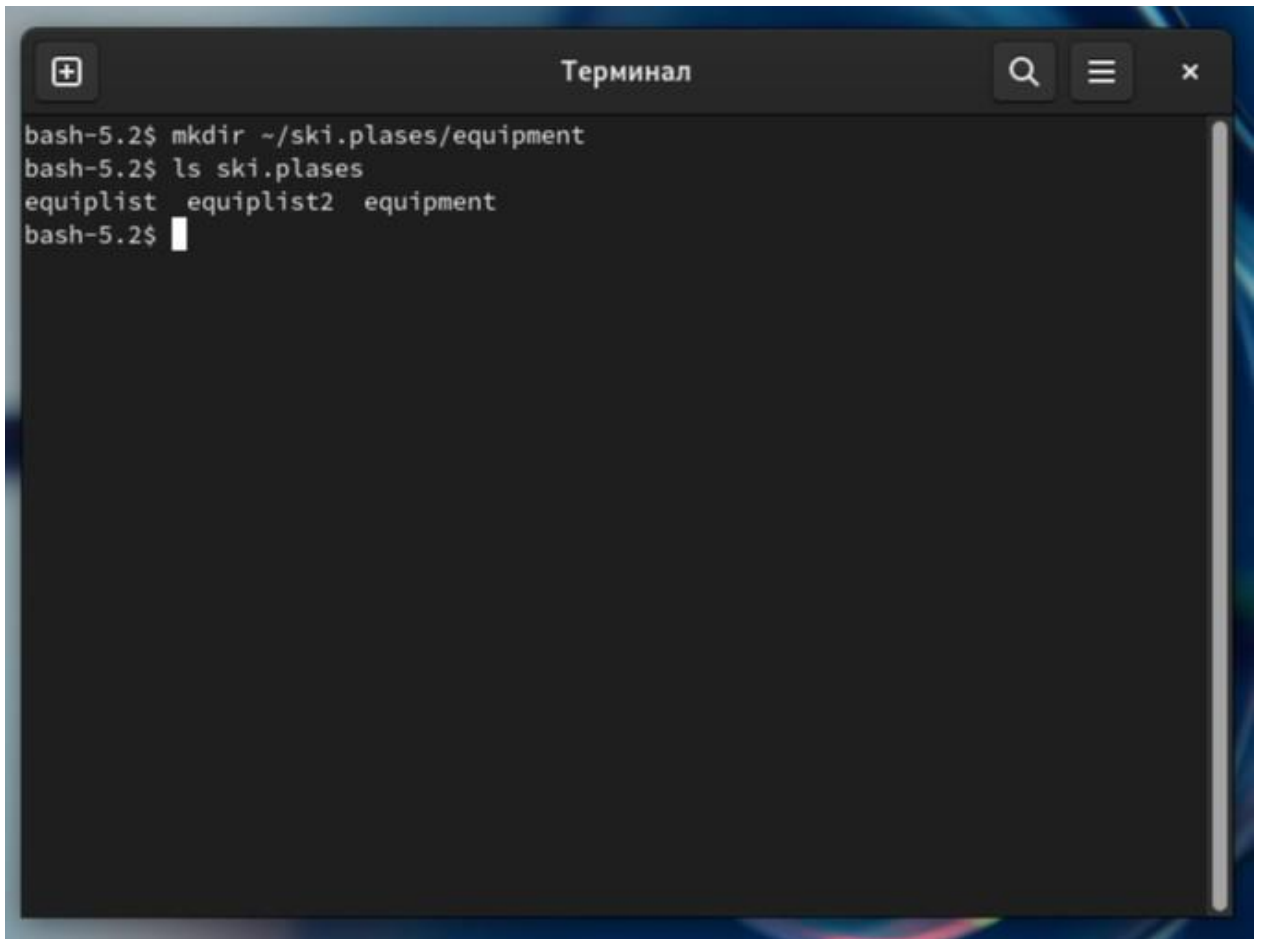
2.5. Создадим в домашнем каталоге файл `abc1` и скопируем его в каталог `~/ski.places`, назовем его `equiplist2` (рис 2.5):



```
bash-5.2$ touch abc1
bash-5.2$ ls
abc1      monthly  ski.plases  Документы  Музыка      Шаблоны
COURSE    newdir   work        Загрузки   Общедоступные
may       reports  Видео       Изображения 'Рабочий стол'
bash-5.2$ cp abc1 ~/ski.plases
bash-5.2$ ls ski.plases
abc1 equipment
bash-5.2$
```

Рисунок 2.5. Создание каталога abc1 и копирование его в ~/ski.plases.

2.6. Создадим каталог с именем equipment в каталоге ~/ski.plases (рис. 2.6):

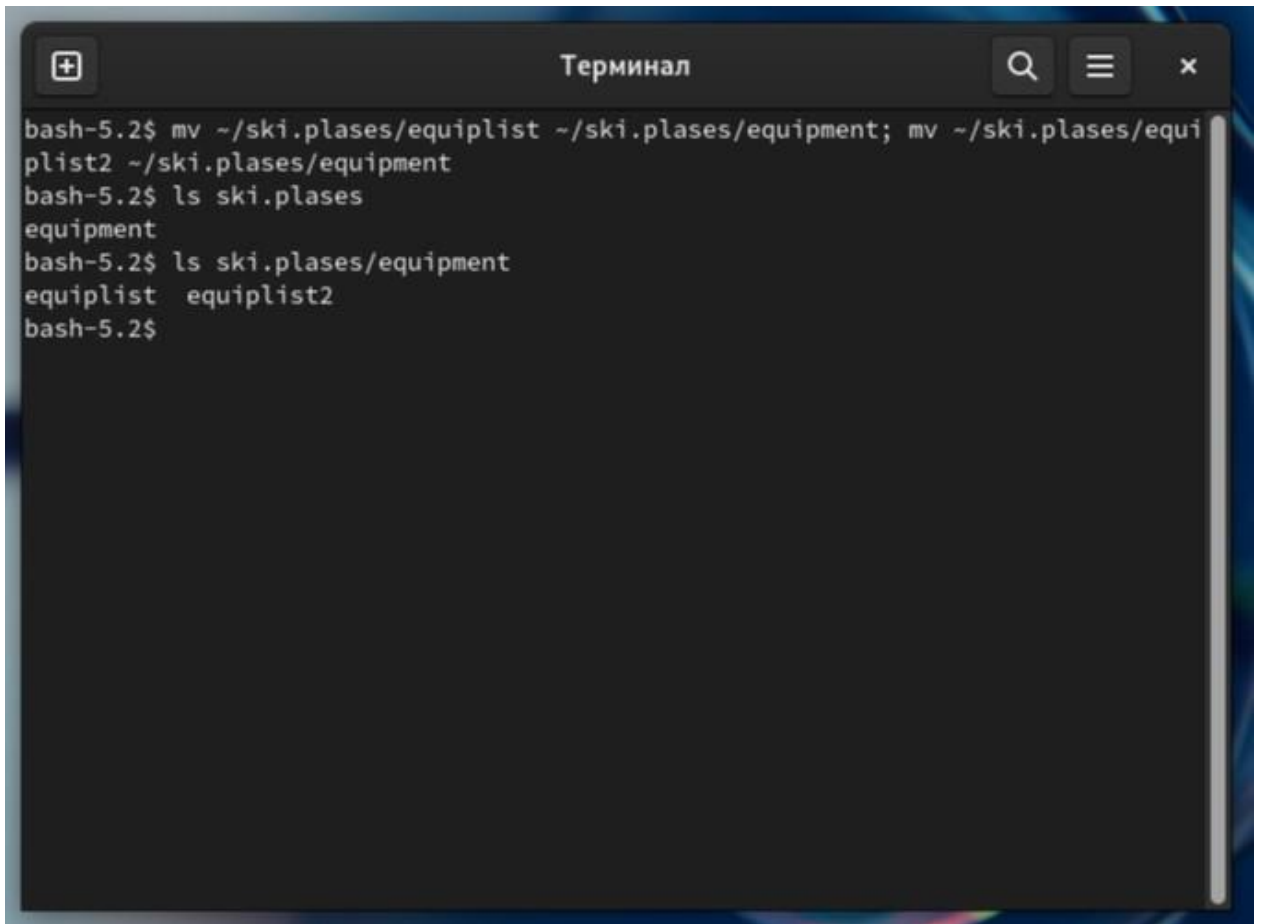
A screenshot of a macOS Terminal window titled "Терминал". The window has a dark background and a light gray title bar with standard window controls (a plus icon, a magnifying glass, a hamburger menu, and a close icon). The terminal shows the following commands and output:

```
bash-5.2$ mkdir ~/ski.plases/equipment
bash-5.2$ ls ski.plases
equiplist  equiplist2  equipment
bash-5.2$
```

The cursor is at the end of the last line, ready for the next command.

Рисунок 2.6. Создание каталога equipment.

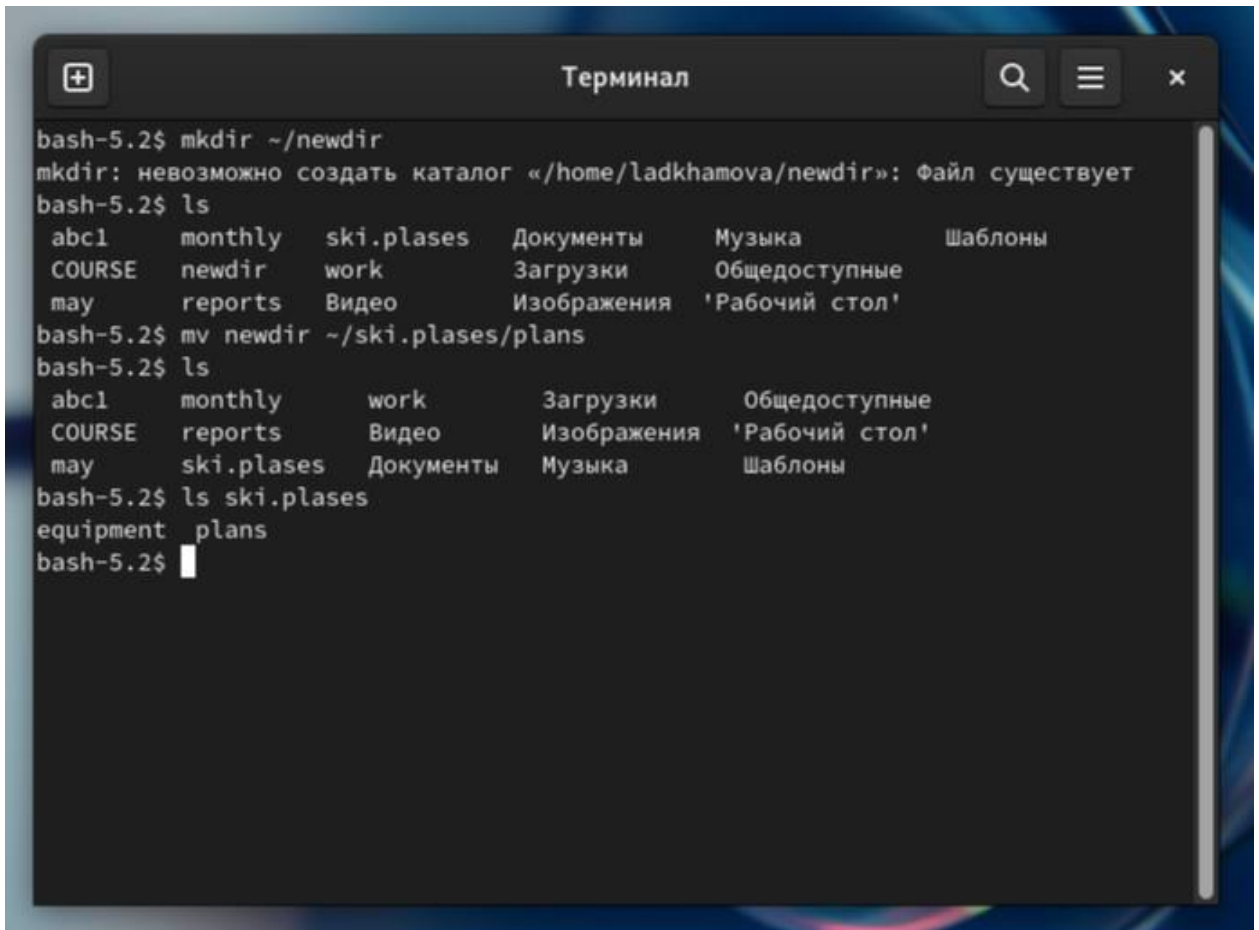
2.7. Переместим файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment` (рис. 2.7.):



```
bash-5.2$ mv ~/ski.plases/equiplist ~/ski.plases/equipment; mv ~/ski.plases/equiplist2 ~/ski.plases/equipment
bash-5.2$ ls ski.plases
equipment
bash-5.2$ ls ski.plases/equipment
equiplist  equiplist2
bash-5.2$
```

Рисунок 2.7. ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment

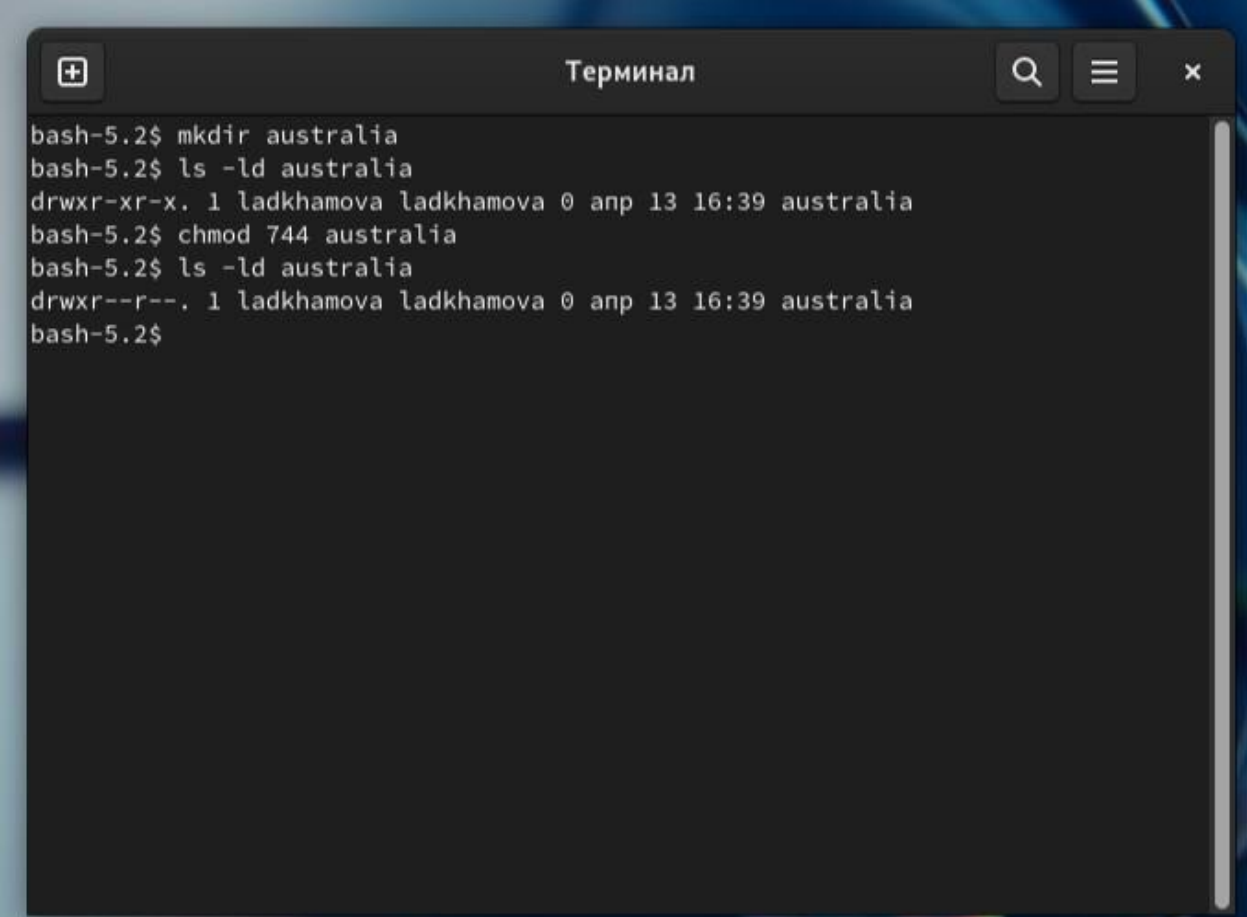
2.8. Создадим и переместим каталог ~/newdir в каталог ~/ski.plases и назовем его plans (рис. 2.8):

A terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the title bar. The terminal shows a series of commands and their outputs. The first command is `mkdir ~/newdir`, which fails with the message "mkdir: невозможно создать каталог «/home/ladkhamova/newdir»: Файл существует". The second command is `ls`, which lists the contents of the current directory: `abc1`, `monthly`, `ski.plases`, `Документы`, `Музыка`, and `Шаблоны`. The third command is `mv newdir ~/ski.plases/plans`, which moves the `newdir` directory to `~/ski.plases/plans`. The fourth command is `ls`, which lists the contents of the current directory again, showing that `newdir` is no longer present. The fifth command is `ls ski.plases`, which lists the contents of the `ski.plases` directory, showing `equipment` and `plans`. The terminal prompt is `bash-5.2$`.

```
bash-5.2$ mkdir ~/newdir
mkdir: невозможно создать каталог «/home/ladkhamova/newdir»: Файл существует
bash-5.2$ ls
abc1    monthly  ski.plases  Документы  Музыка      Шаблоны
COURSE  newdir   work        Загрузки   Общедоступные
may     reports  Видео       Изображения 'Рабочий стол'
bash-5.2$ mv newdir ~/ski.plases/plans
bash-5.2$ ls
abc1    monthly  work        Загрузки   Общедоступные
COURSE  reports  Видео       Изображения 'Рабочий стол'
may     ski.plases  Документы  Музыка      Шаблоны
bash-5.2$ ls ski.plases
equipment  plans
bash-5.2$
```

Рисунок 2.8. Создание и перемещение.

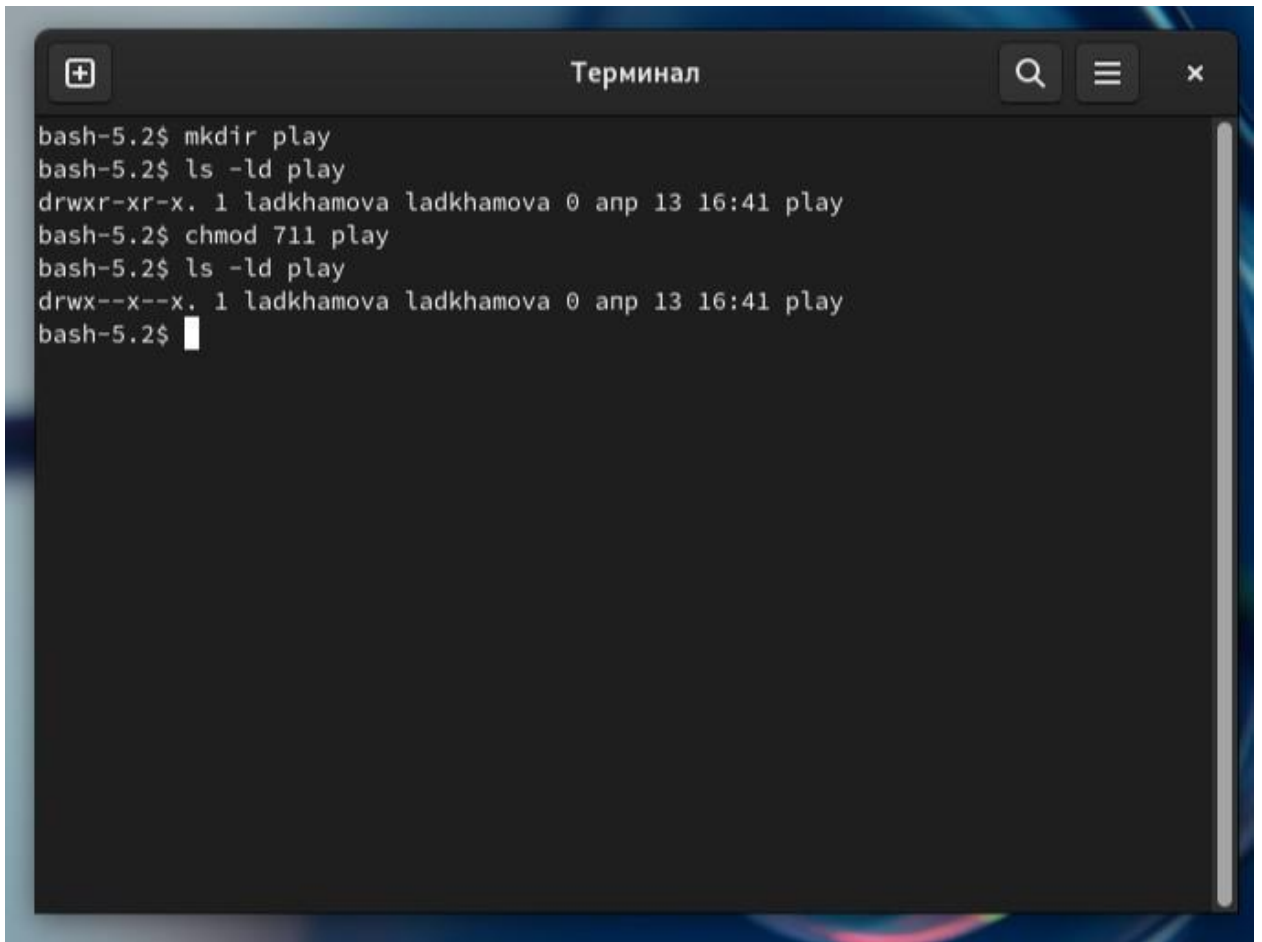
3. Определим опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:



```
bash-5.2$ mkdir australia
bash-5.2$ ls -ld australia
drwxr-xr-x. 1 ladkhamova ladkhamova 0 апр 13 16:39 australia
bash-5.2$ chmod 744 australia
bash-5.2$ ls -ld australia
drwxr--r--. 1 ladkhamova ladkhamova 0 апр 13 16:39 australia
bash-5.2$
```

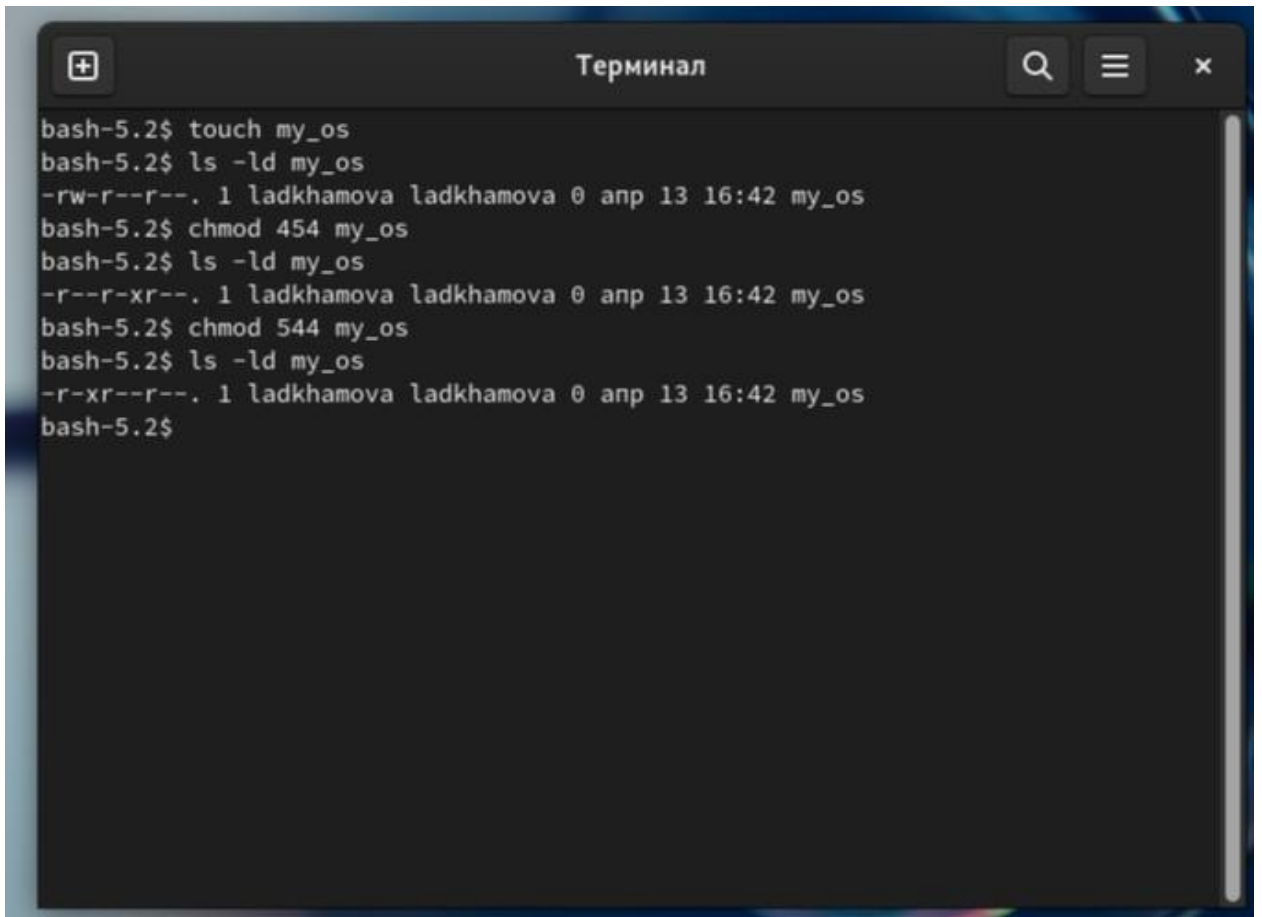
Рисунок 3.1. drwxr--r-- ... australia.



A screenshot of a macOS Terminal window titled "Терминал". The window has a dark background and a light gray title bar with standard macOS window controls (a plus icon on the left, and search, menu, and close icons on the right). The terminal shows a series of commands and their outputs. The user creates a directory named "play", lists its permissions, changes them to 711, lists them again, and then the prompt returns. The text is as follows:

```
bash-5.2$ mkdir play
bash-5.2$ ls -ld play
drwxr-xr-x. 1 ladkhamova ladkhamova 0 апр 13 16:41 play
bash-5.2$ chmod 711 play
bash-5.2$ ls -ld play
drwx--x--x. 1 ladkhamova ladkhamova 0 апр 13 16:41 play
bash-5.2$
```

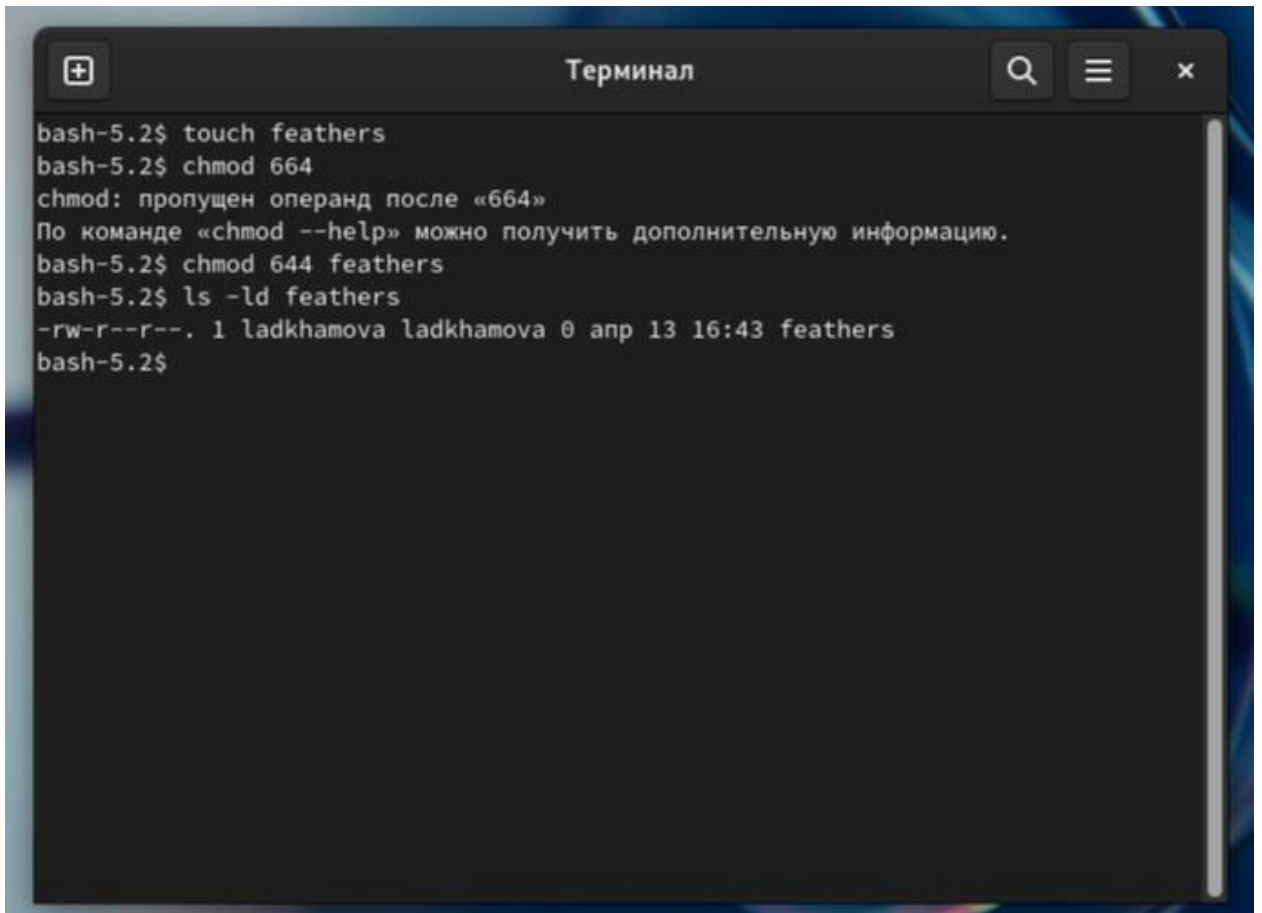
Рисунок 3.2. drwx--x--x ... play.

A terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the title bar. The terminal shows a series of commands and their outputs. The commands are: `touch my_os`, `ls -ld my_os`, `chmod 454 my_os`, `ls -ld my_os`, `chmod 544 my_os`, and `ls -ld my_os`. The outputs show the file `my_os` being created with permissions `-rw-r--r--`, then `-r--r-xr--`, and finally `-r-xr--r--`. The user is `ladkhamova` and the group is `ladkhamova`. The terminal is running on a `bash-5.2` shell.

```
bash-5.2$ touch my_os
bash-5.2$ ls -ld my_os
-rw-r--r--. 1 ladkhamova ladkhamova 0 anp 13 16:42 my_os
bash-5.2$ chmod 454 my_os
bash-5.2$ ls -ld my_os
-r--r-xr--. 1 ladkhamova ladkhamova 0 anp 13 16:42 my_os
bash-5.2$ chmod 544 my_os
bash-5.2$ ls -ld my_os
-r-xr--r--. 1 ladkhamova ladkhamova 0 anp 13 16:42 my_os
bash-5.2$
```

Рисунок 3.3. -r-xr--r-- ... my\_os.

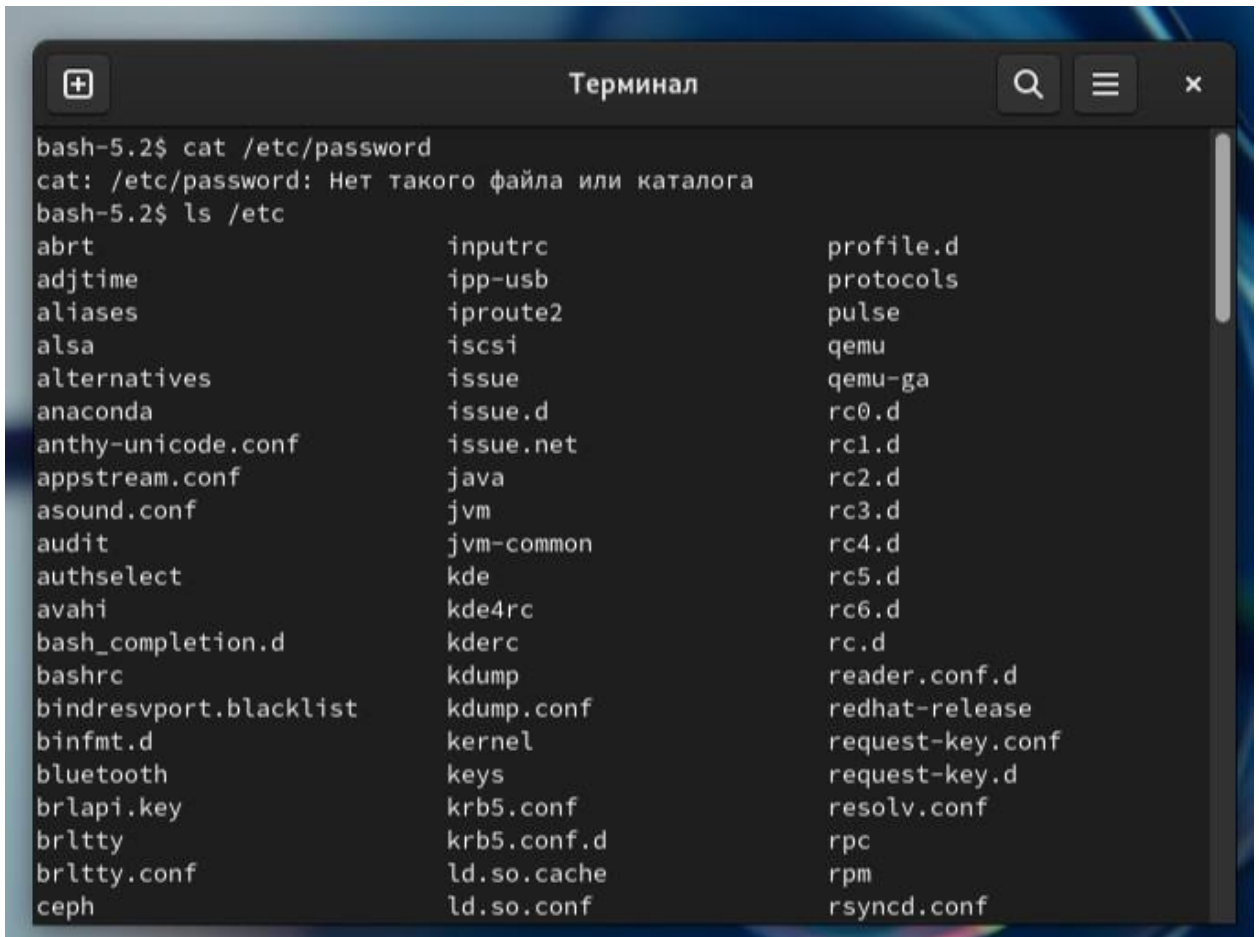
3.1. При необходимости создадим нужные файлы (рис. 3.4):

A terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the title bar. The terminal shows the following commands and output:

```
bash-5.2$ touch feathers
bash-5.2$ chmod 664
chmod: пропущен операнд после «664»
По команде «chmod --help» можно получить дополнительную информацию.
bash-5.2$ chmod 644 feathers
bash-5.2$ ls -ld feathers
-rw-r--r--. 1 ladkhamova ladkhamova 0 апр 13 16:43 feathers
bash-5.2$
```

Рисунок 3.4. -rw-rw-r-- ... feathers.

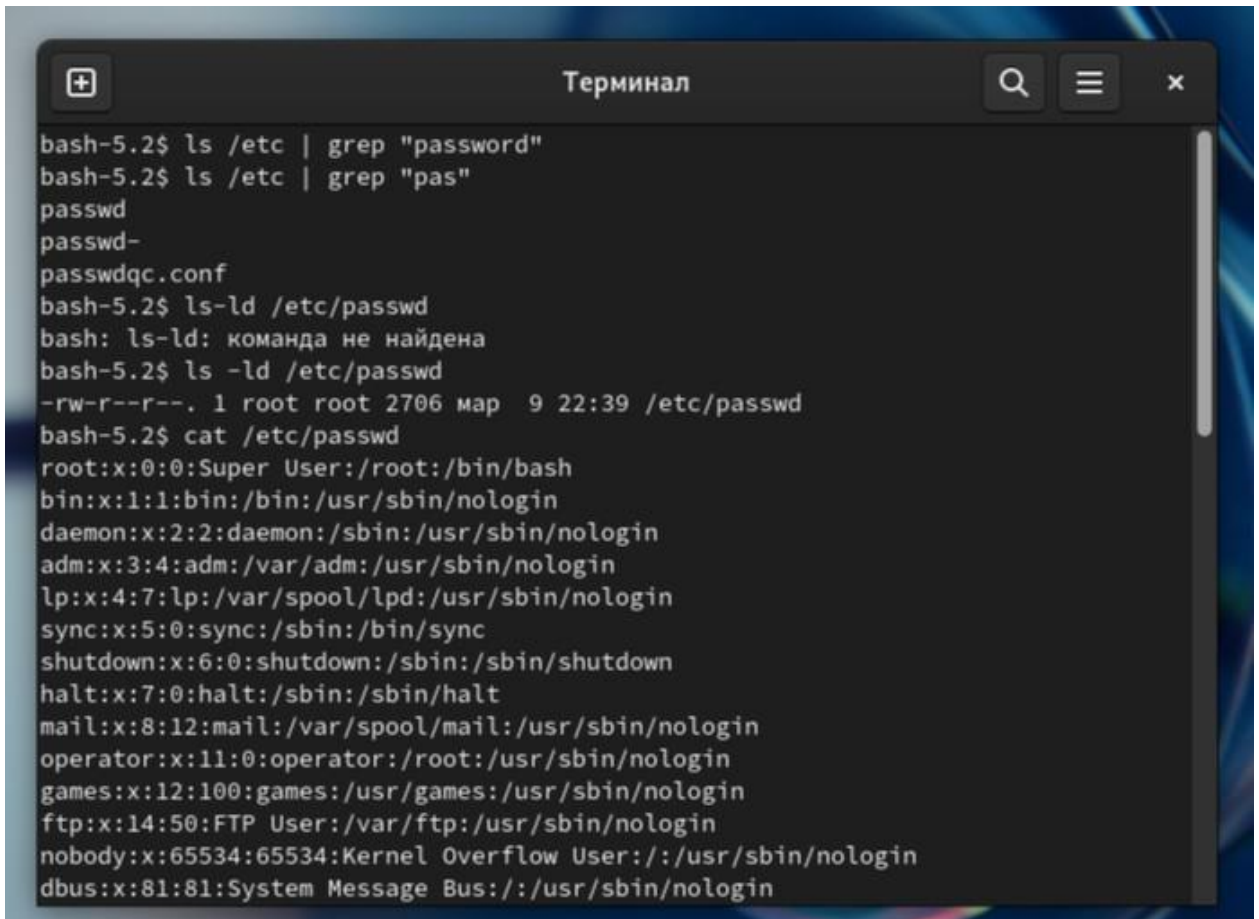
4. Прделаем приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
  - 4.1. Просмотрим содержимое файла /etc/password (рис. 4.1, 4.2):



The image shows a terminal window with a dark background and a light gray title bar. The title bar contains a plus icon on the left, the word "Терминал" in the center, and search, menu, and close icons on the right. The terminal content shows a user at a bash prompt attempting to view the contents of /etc/password, which fails. Subsequently, the user lists the contents of the /etc directory, resulting in a three-column output of files and directories.

```
bash-5.2$ cat /etc/password
cat: /etc/password: Нет такого файла или каталога
bash-5.2$ ls /etc
abrt                                inputrc                            profile.d
adjtime                            ipp-usb                           protocols
aliases                            iproute2                          pulse
alsa                               iscsi                             qemu
alternatives                       issue                             qemu-ga
anaconda                           issue.d                           rc0.d
anthy-unicode.conf                 issue.net                         rc1.d
appstream.conf                     java                              rc2.d
asound.conf                        jvm                               rc3.d
audit                              jvm-common                       rc4.d
authselect                         kde                               rc5.d
avahi                              kde4rc                           rc6.d
bash_completion.d                 kderc                            rc.d
bashrc                             kdump                            reader.conf.d
bindresvport.blacklist             kdump.conf                      redhat-release
binfmt.d                           kernel                           request-key.conf
bluetooth                          keys                             request-key.d
brlapi.key                         krb5.conf                       resolv.conf
brltty                             krb5.conf.d                     rpc
brltty.conf                       ld.so.cache                     rpm
ceph                               ld.so.conf                      rsyncd.conf
```

Рисунок 4.1. Выполнение команды `/etc/password`.

A terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the title bar. The terminal shows the following commands and output:

```
bash-5.2$ ls /etc | grep "password"
bash-5.2$ ls /etc | grep "pas"
passwd
passwd-
passwdqc.conf
bash-5.2$ ls -ld /etc/passwd
bash: ls-ld: команда не найдена
bash-5.2$ ls -ld /etc/passwd
-rw-r--r--. 1 root root 2706 map  9 22:39 /etc/passwd
bash-5.2$ cat /etc/passwd
root:x:0:0:Super User:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/sbin:/usr/sbin/nologin
adm:x:3:4:adm:/var/adm:/usr/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/usr/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/usr/sbin/nologin
operator:x:11:0:operator:/root:/usr/sbin/nologin
games:x:12:100:games:/usr/games:/usr/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/usr/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/usr/sbin/nologin
dbus:x:81:81:System Message Bus:/:/usr/sbin/nologin
```

Рисунок 4.2. Продолжение.

4.2. Скопируем файл ~/feathers в файл ~/file.old (рис. 4.3, 4.4):

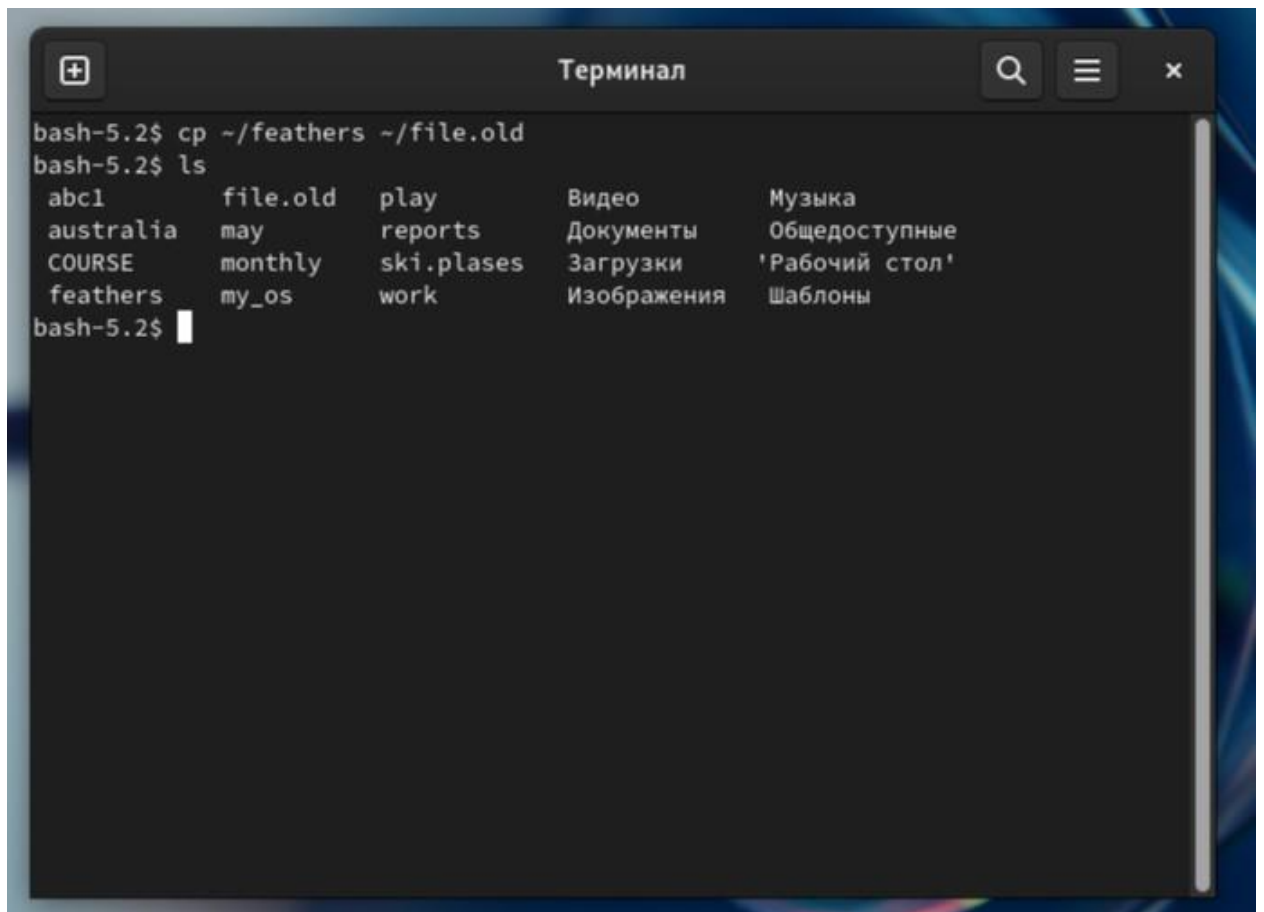


Рисунок 4.3. ~/feathers в файл ~/file.old

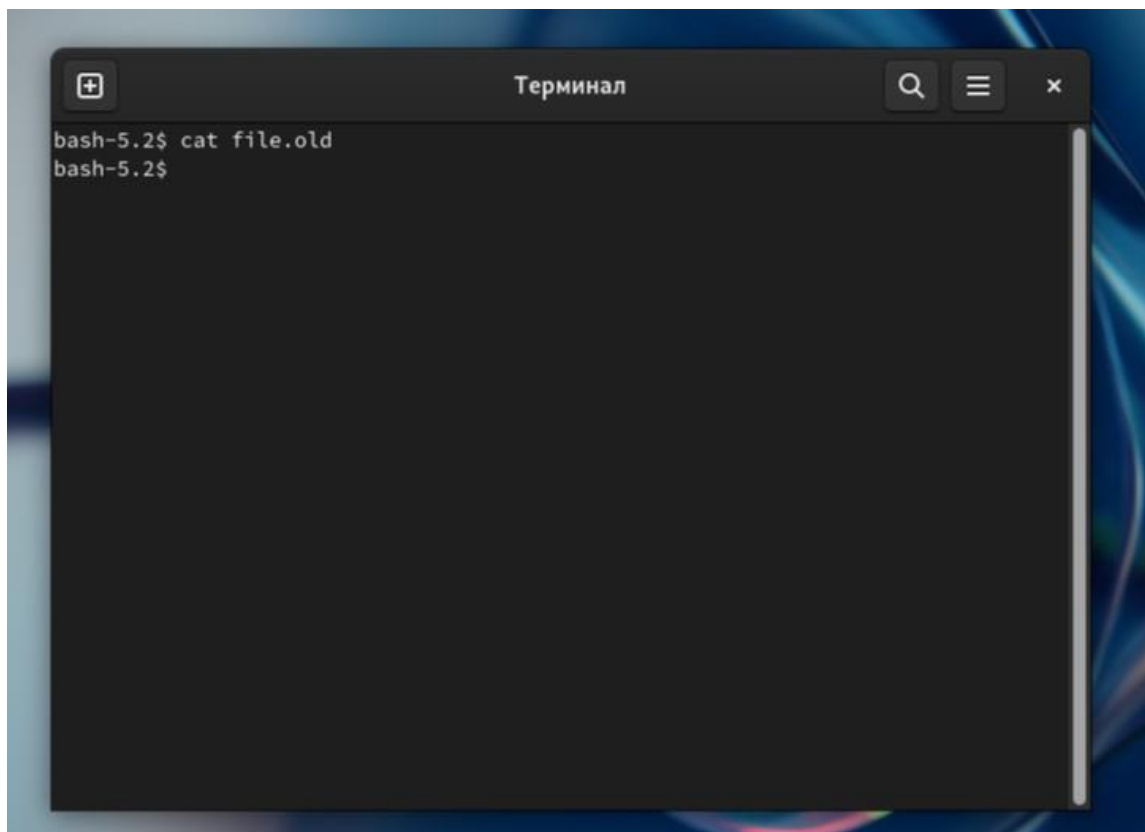
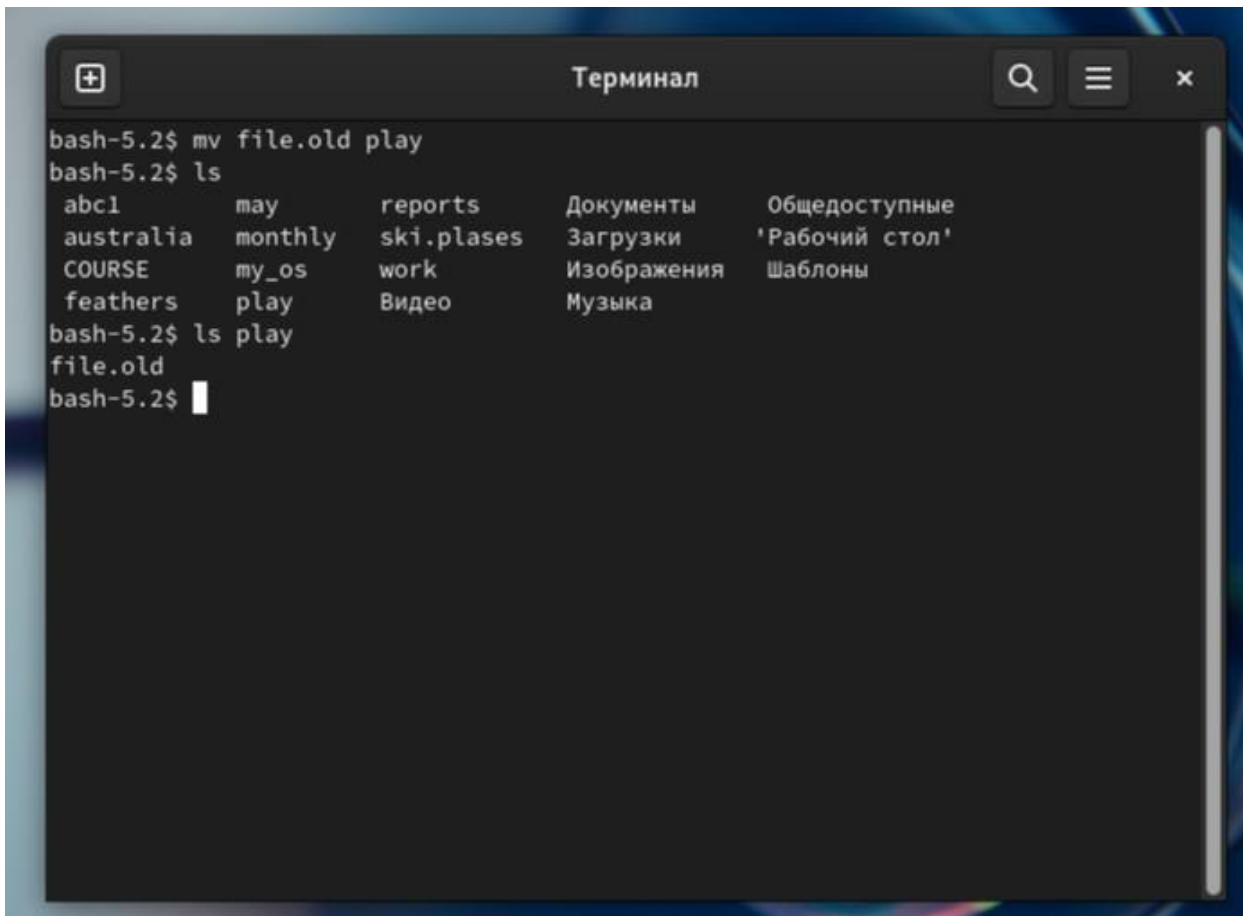


Рисунок 4.4. Продолжение.

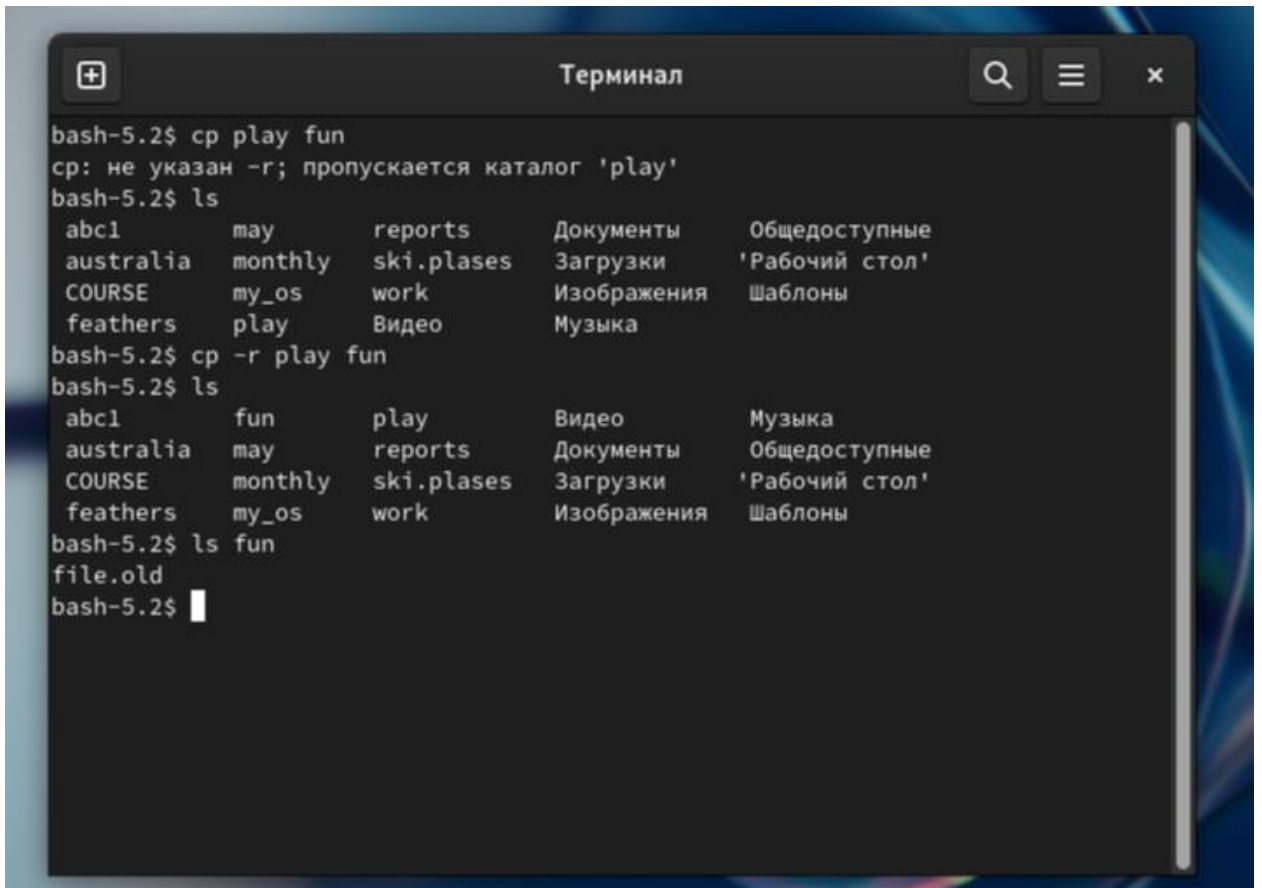
4.3. Переместим файл ~/file.old в каталог ~/play (рис. 4.5):

A screenshot of a macOS Terminal window titled "Терминал". The window shows a series of commands and their outputs. The first command is `mv file.old play`, which moves the file `file.old` to the `play` directory. The second command is `ls`, which lists the contents of the current directory. The output shows a list of files and directories: `abc1`, `australia`, `COURSE`, `feathers`, `may`, `monthly`, `my_os`, `play`, `reports`, `ski.plases`, `work`, `Видео`, `Документы`, `Загрузки`, `Изображения`, `Музыка`, `Общедоступные`, and `'Рабочий стол'`. The third command is `ls play`, which lists the contents of the `play` directory. The output shows the file `file.old`. The prompt `bash-5.2$` is visible at the end of each line.

```
bash-5.2$ mv file.old play
bash-5.2$ ls
abc1      may      reports  Документы  Общедоступные
australia monthly  ski.plases  Загрузки    'Рабочий стол'
COURSE    my_os    work      Изображения  Шаблоны
feathers   play     Видео     Музыка
bash-5.2$ ls play
file.old
bash-5.2$
```

Рисунок 4.5. Перемещение файла.

4.4. Скопируем каталог ~/play в каталог ~/fun (рис. 4.6):



```
bash-5.2$ cp play fun
cp: не указан -r; пропускается каталог 'play'
bash-5.2$ ls
abc1      may      reports  Документы  Общедоступные
australia monthly  ski.places Загрузки   'Рабочий стол'
COURSE    my_os    work     Изображения Шаблоны
feathers   play     Видео    Музыка
bash-5.2$ cp -r play fun
bash-5.2$ ls
abc1      fun      play     Видео      Музыка
australia may      reports  Документы  Общедоступные
COURSE    monthly  ski.places Загрузки   'Рабочий стол'
feathers   my_os    work     Изображения Шаблоны
bash-5.2$ ls fun
file.old
bash-5.2$
```

Рисунок 4.6. Копирование.

4.5. Переместим каталог ~/fun в каталог ~/play и назовем его games (рис. 4.7):



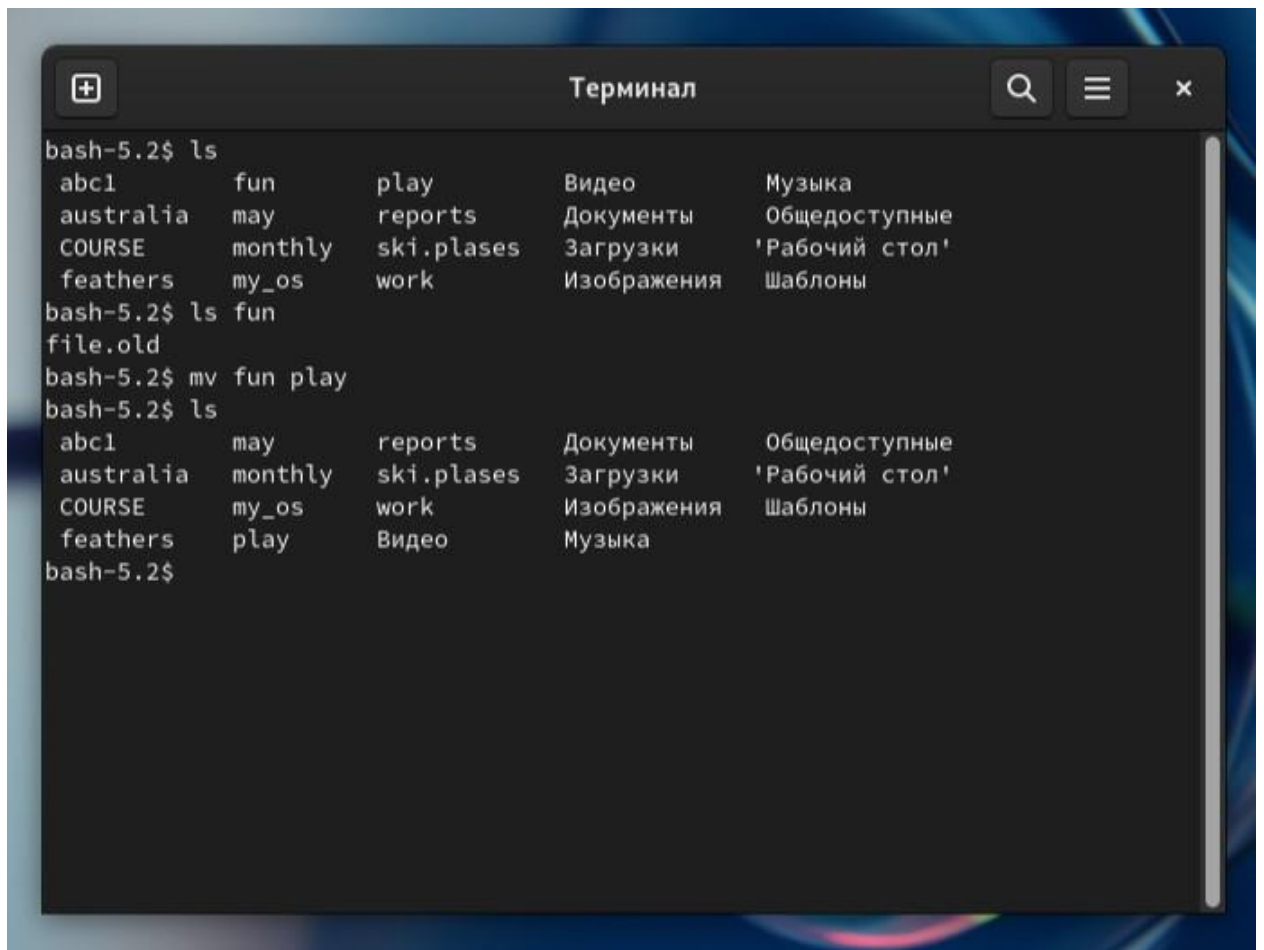


Рисунок 4.7. ~/fun в каталог ~/play.

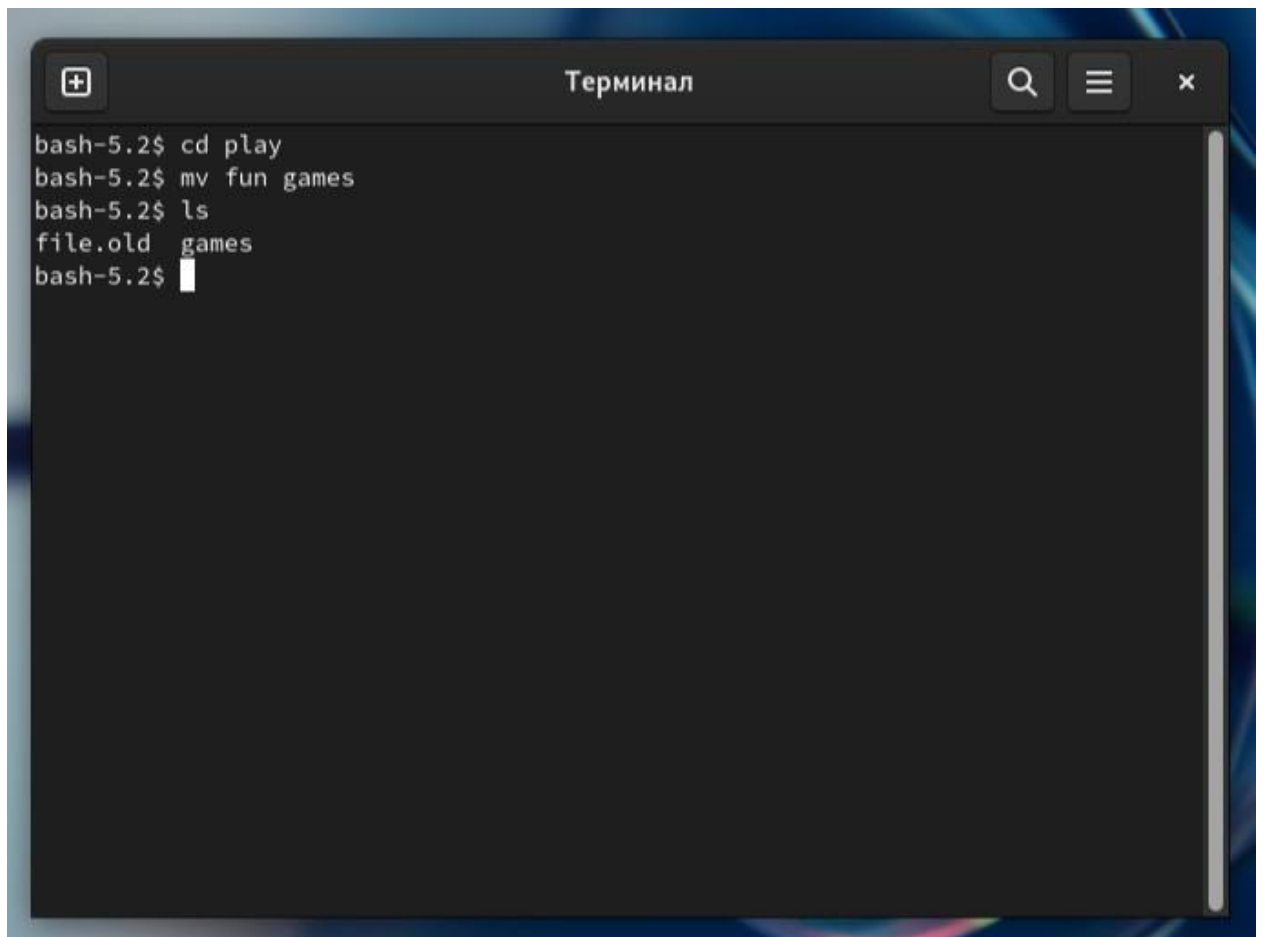
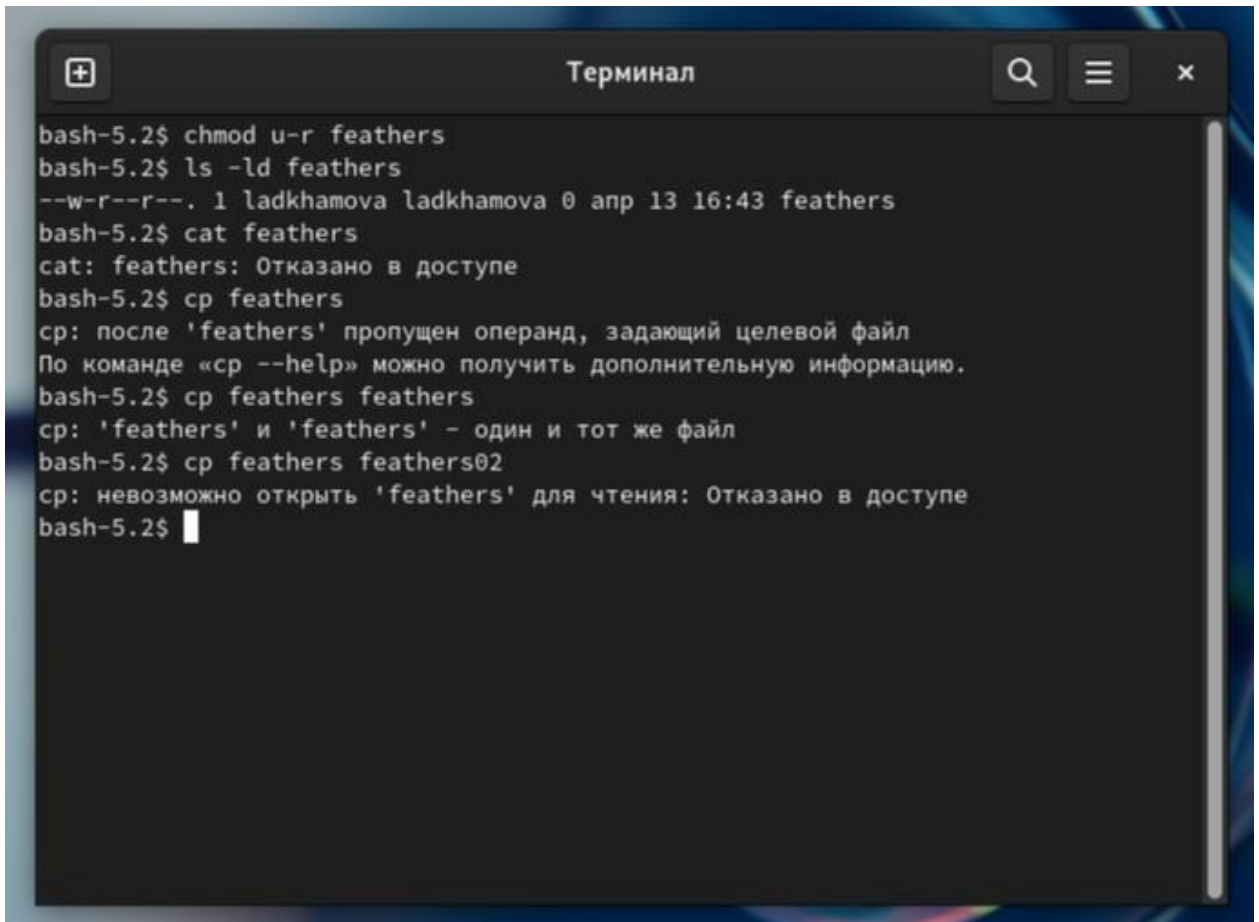


Рисунок 4.8. Называем файл games.

4.6. Лишим владельца файла ~/feathers права на чтение (рис. 4.9):



```
bash-5.2$ chmod u-r feathers
bash-5.2$ ls -ld feathers
--w-r--r--. 1 ladkhamova ladkhamova 0 апр 13 16:43 feathers
bash-5.2$ cat feathers
cat: feathers: Отказано в доступе
bash-5.2$ cp feathers
cp: после 'feathers' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
bash-5.2$ cp feathers feathers
cp: 'feathers' и 'feathers' - один и тот же файл
bash-5.2$ cp feathers feathers02
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
bash-5.2$
```

Рисунок 4.9. Лишение права на чтение.

- 4.7. Что произойдёт, если мы попытаемся просмотреть файл ~/feathers командой cat? (рис. 4.9):
- 4.8. Что произойдёт, если мы попытаемся скопировать файл ~/feathers? (рис. 4.9):
- 4.9. Дадим владельцу файла ~/feathers право на чтение (рис. 4.10):

```
bash-5.2$ mkdir tmp05_lab
bash-5.2$ cp feathers ~/ tmp05_lab
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
cp: не указан -r; пропускается каталог '/home/ladkhamova/'
bash-5.2$ cp feathers ~/tmp05_lab
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
bash-5.2$ chmod u+r feathers
bash-5.2$ ls -ld feathers
-rw-r--r--. 1 ladkhamova ladkhamova 0 апр 13 16:43 feathers
bash-5.2$
```

Рисунок 4.10. Дадим право владельцу на чтение.

4.10. Лишим владельца каталога ~/play права на выполнение (рис. 4.11):

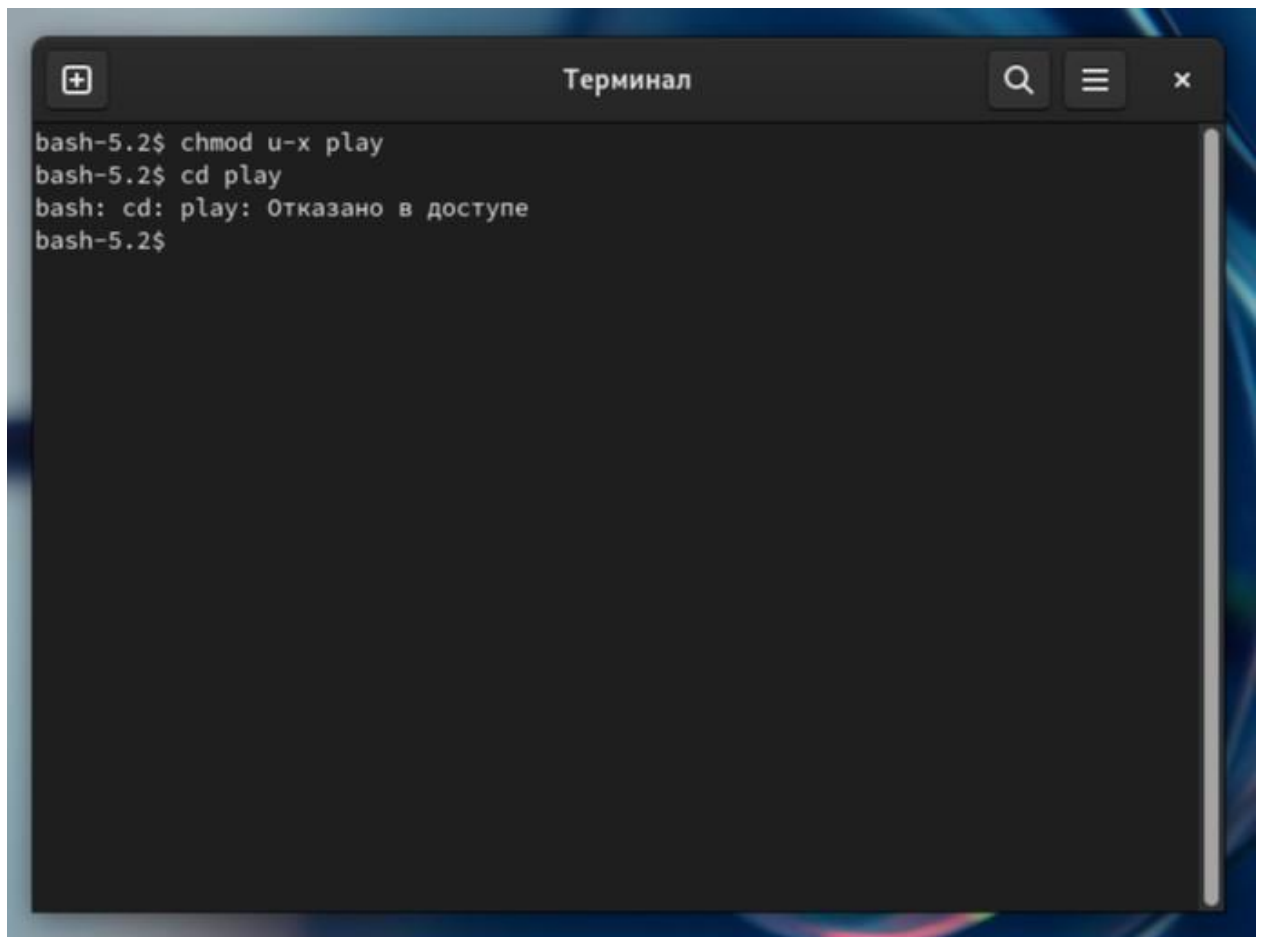


Рисунок 4.11. Лишение владельца права в каталог play.

4.11. Перейдем в каталог ~/play (рис. 4.11):

4.12. Дадим владельцу каталога ~/play право на выполнение (рис. 4.12):

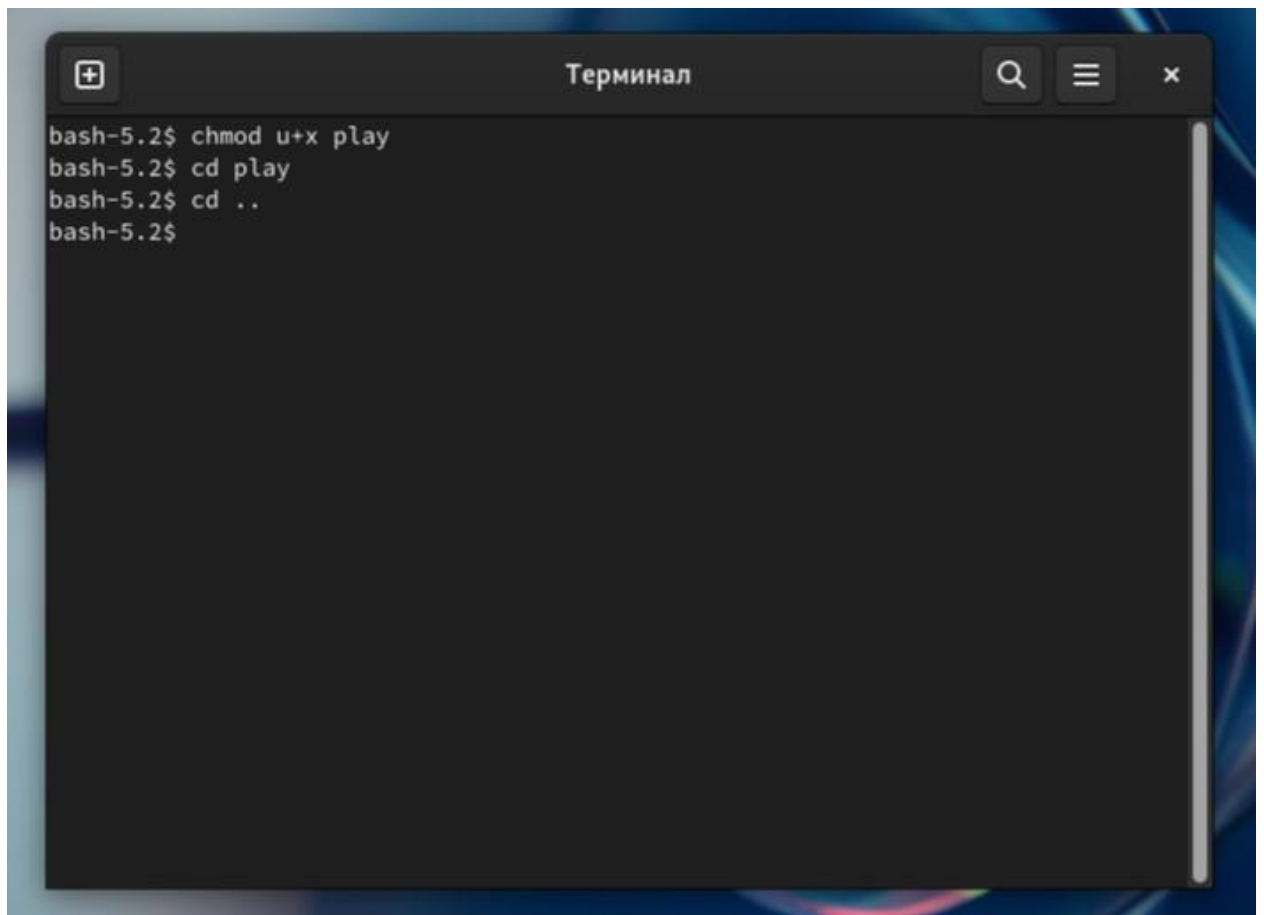
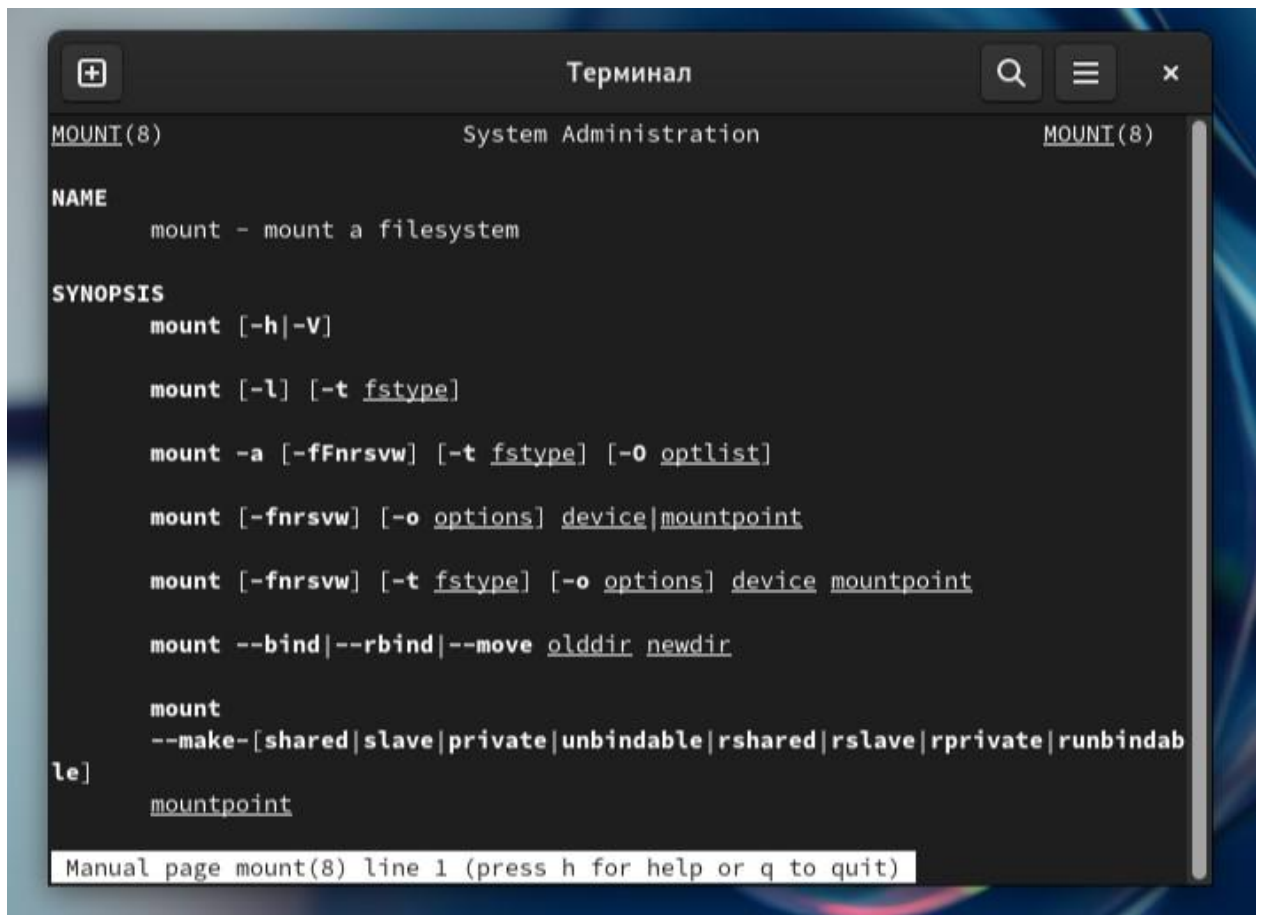


Рисунок 4.12. Дадим владельцу каталога ~/play право на выполнение.

5. Прочитаем man по командам mount, fsck, mkfs, kill и кратко их охарактеризуем, приведя примеры (рис. 5.1, 5.2, 5.3, 5.4):



The image shows a terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the top right corner. The window displays the manual page for the `mount` command. The title bar also shows "MOUNT(8)" on the left and right. The content includes the command name, a synopsis with various options, and a description. The terminal text is as follows:

```
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

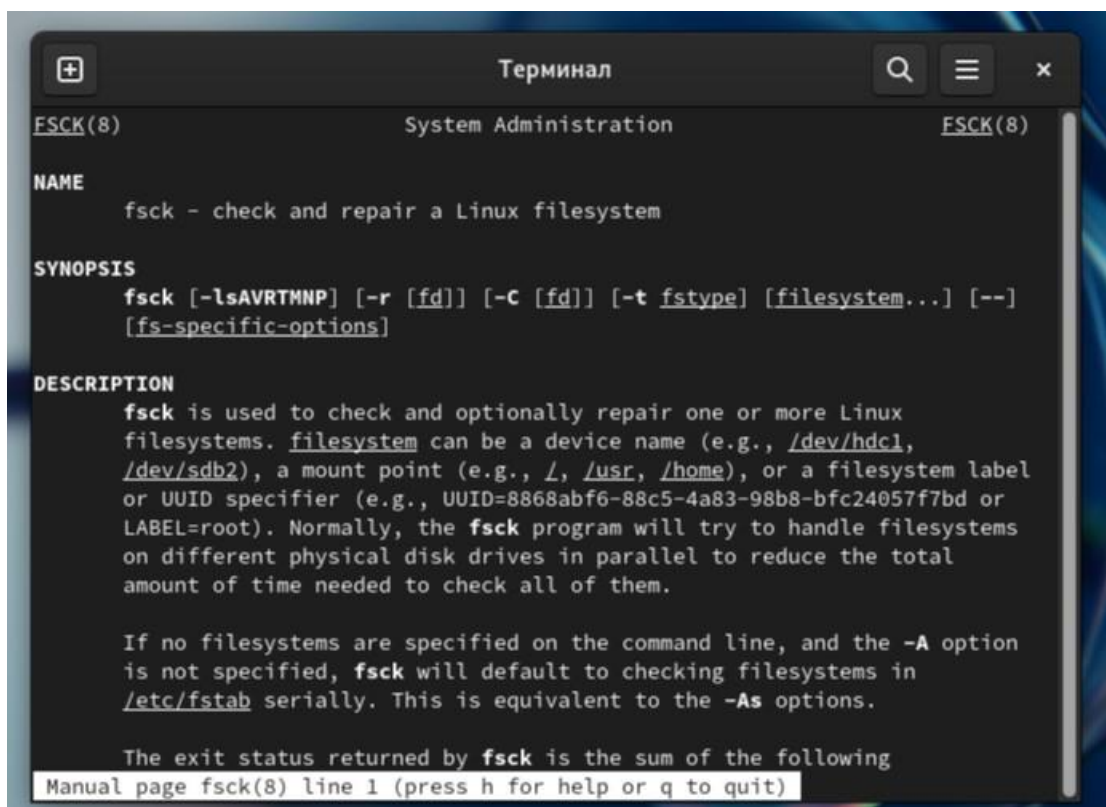
    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

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

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

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

Рисунок 5.1. Выполнение с помощью команды `mount`.



The image shows a terminal window titled "Терминал" (Terminal) with a search icon, a menu icon, and a close icon in the top right corner. The window displays the manual page for the `fsck` command. The title bar also shows "FSCK(8)" on the left and right. The content includes the command name, a synopsis, and a detailed description. The terminal text is as follows:

```
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 a 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

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

Рисунок 5.2. Выполнение с помощью команды `fsck`.

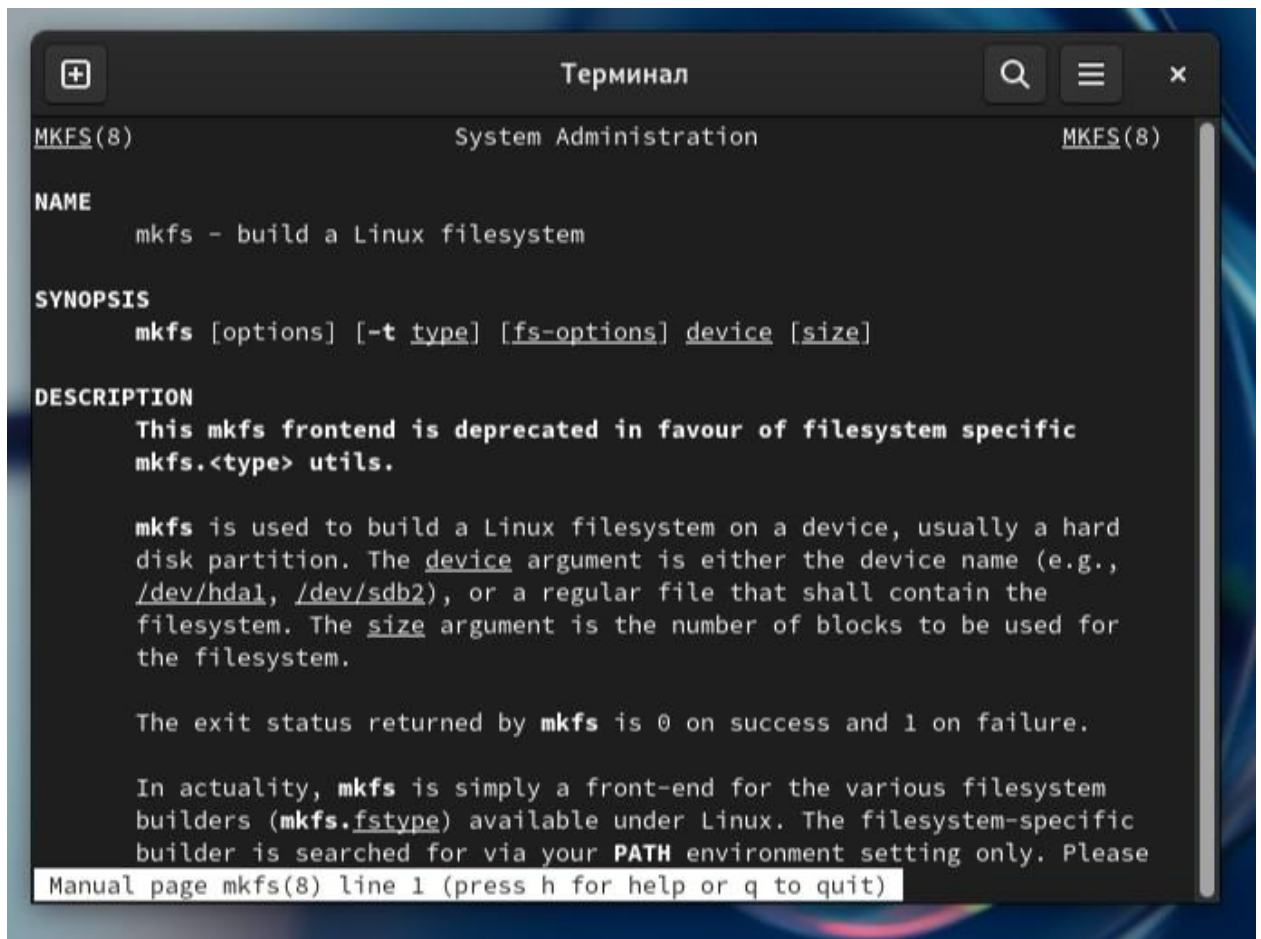
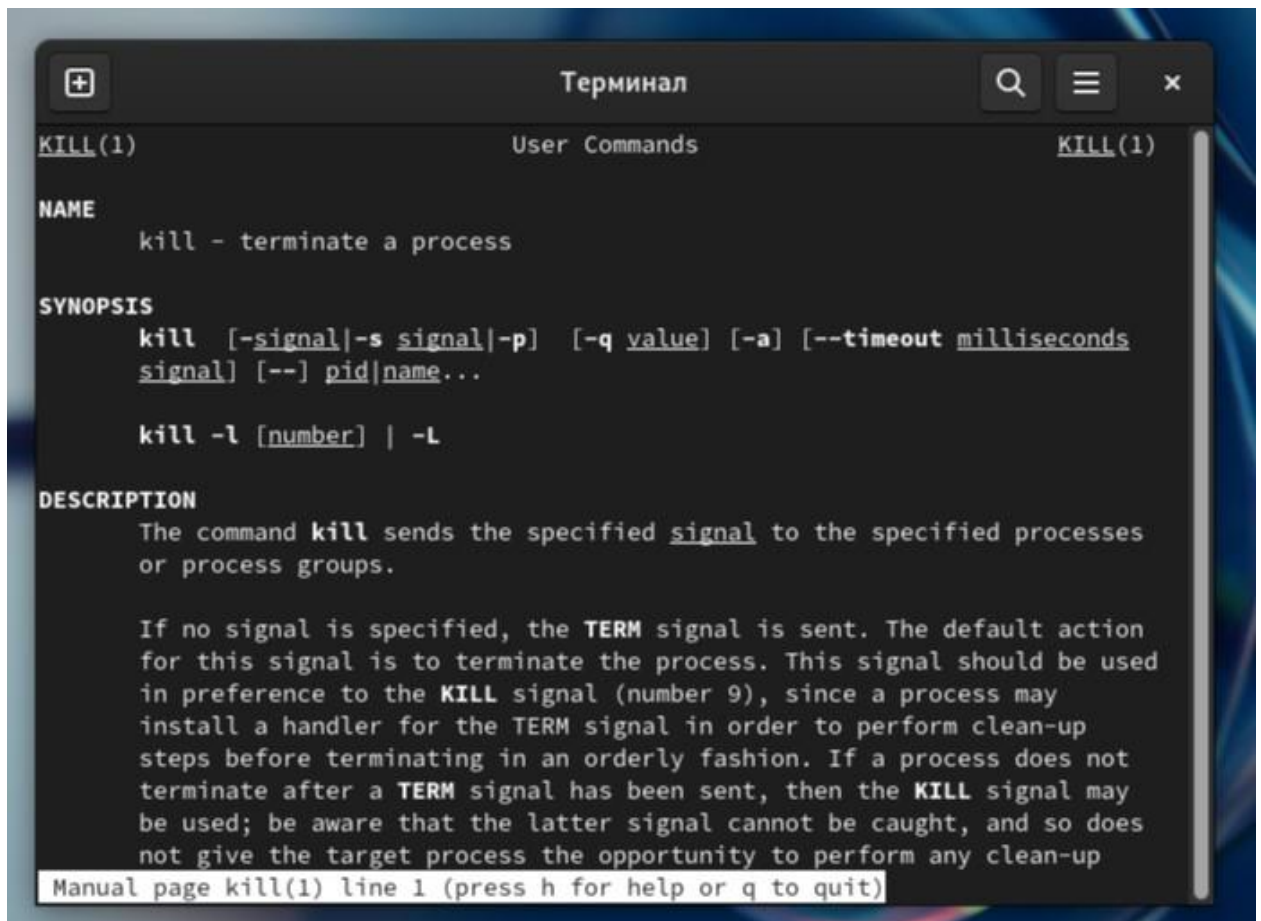


Рисунок 5.3. Выполнение с помощью команды `mkfs`.





```
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

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

Рисунок 5.3. Выполнение с помощью команды kill.

## Вывод:

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

## Ответы на контрольные вопросы:

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

TARGET	SOURCE	FSTYPE	OPTIONS
/proc	proc	proc	rw,nosuid,nodev,noexec,relatime
/sys	sysfs	sysfs	rw,nosuid,nodev,noexec,relatime,se
/dev	devtmpfs	devtmpfs	rw,nosuid,seclabel,size=4096k,nr_i
/sys/kernel/security	securityfs	securit	rw,nosuid,nodev,noexec,relatime
/dev/shm	tmpfs	tmpfs	rw,nosuid,nodev,seclabel,inode64
/dev/pts	devpts	devpts	rw,nosuid,noexec,relatime,seclabel
/run	tmpfs	tmpfs	rw,nosuid,nodev,seclabel,size=7852
/sys/fs/cgroup	cgroup2	cgroup2	rw,nosuid,nodev,noexec,relatime,se
/sys/fs/pstore	pstore	pstore	rw,nosuid,nodev,noexec,relatime,se
/sys/firmware/efi/efivars	efivarfs	efivarf	rw,nosuid,nodev,noexec,relatime
/sys/fs/bpf	bpf	bpf	rw,nosuid,nodev,noexec,relatime,mo
/	/dev/sda3[/root]	btrfs	rw,relatime,seclabel,compress=zstd
/sys/fs/selinux	selinuxfs	selinux	rw,nosuid,noexec,relatime
/proc/sys/fs/binfmt_misc	systemd-1	autofs	rw,relatime,fd=35,pgrp=1,timeout=0
/sys/kernel/tracing	tracefs	tracefs	rw,nosuid,nodev,noexec,relatime,se
/sys/kernel/debug	debugfs	debugfs	rw,nosuid,nodev,noexec,relatime,se
/dev/hugepages	hugetlbfs	hugetlb	rw,relatime,seclabel,pagesize=2M
/dev/mqueue	mqueue	mqueue	rw,nosuid,nodev,noexec,relatime,se
/sys/fs/fuse/connections	fusectl	fusectl	rw,nosuid,nodev,noexec,relatime
/sys/kernel/config	configfs	configf	rw,nosuid,nodev,noexec,relatime
/tmp	tmpfs	tmpfs	rw,nosuid,nodev,seclabel,size=1963
/home	/dev/sda3[/home]	btrfs	rw,relatime,seclabel,compress=zstd
/boot	/dev/sda2	ext4	rw,relatime,seclabel
/boot/efi	/dev/sda1	vfat	rw,relatime,fmask=0077,dmask=0077,
/proc/sys/fs/binfmt_misc	binfmt_misc	binfmt_	rw,nosuid,nodev,noexec,relatime
/var/lib/nfs/rpc_pipefs	sunrpc	rpc_pip	rw,relatime
/run/user/1000	tmpfs	tmpfs	rw,nosuid,nodev,relatime,seclabel,
/run/user/1000/gvfs	gvfsd-fuse	fuse.gv	rw,nosuid,nodev,relatime,user_id=1
/run/user/1000/doc	portal	fuse.po	rw,nosuid,nodev,relatime,user_id=1
/tmp/.mount_jetbra1uUEwF	jetbrains-toolbox	fuse.je	ro,nosuid,nodev,relatime,user_id=1

- **proc.** Файловая система **proc** является важным источником информации о вашей Linux-системе, который попросту нельзя игнорировать. Вообще, **proc** является псевдо- или виртуальной файловой системой, которая предоставляет пользователям доступ к внутренним структурам ядра Linux. Другими словами, **proc** не является реальной файловой системой в обычном смысле; она располагается исключительно в оперативной памяти, а не на диске. При этом она автоматически монтируется системой.
- **Sysfs** - отправляет данные в пространство пользователя с помощью виртуальных файлов. Эти данные содержат данные о различных подсистемах ядра, аппаратных устройствах и связанных с ними драйверах устройств.
- **tmpfs** и **devtmpfs** - они относятся к энергозависимой памяти.
- **devpts** - обеспечивает доступ к терминалам **pseudo (PTY)**.
- **cgroup2** - Неверно ведущий себя процесс может создавать тонны процессов через ветвления, запуская некую бомбу ветвлений и сокращая своё ядро. Это означает, что нам требуется ввести некий способ контроля ресурсов для процессов в пределах заданного пространства имён. Это достигается через механизм, носящий название групп контроля (**control groups**), обычно именуемых **cgroups**. **cgroups** работают под понятием контроллеров **cgroup** и представляются в файловой системе с названием **cgroupfs** в самом ядре Linux. В настоящее время применяется **cgroup v2** версия **cgroups**.

- `pstore` - был введен в Linux для записи информации (например, `dmesg tail`) при выключении. `Pstore` не зависит от `kdump` и может запускаться до него. В определенных сценариях (например, хосты/гости с корневыми файловыми системами на NFS/iSCSI, где произошел сбой сетевого программного и/или аппаратного обеспечения), `pstore` может содержать информацию, доступную для посмертной отладки, которая не может быть получена иным образом.
- `brf` - это псевдо-файловая система, существующая только в памяти, которая позволяет создавать файлы, ссылающиеся на объекты BPF.
- `btrfs` - файловая система для Linux, основанная на структурах B-деревьев и работающая по принципу «копирование при записи» (`copy-on-write`). Опубликована корпорацией Oracle в 2007 году под лицензией GNU General Public License.
- `selinux` - Как и файловая система `/proc`, `/selinux` является псевдофайловой системой. Новая реализация SE Linux использует расширенные атрибуты для хранения контекста безопасности.
- `tracefs` - файловая система для задач трассировки Linux
- `debugfs` - `DebugFS` является самой известной утилитой, предназначенной для работы с файловыми системами `EXT2FS` и `EXT3FS`.
- `hugetlb` - использует страницы большого размера, что позволяет кэшировать больше адресов за раз.
- `mqueue` - обеспечивает необходимую поддержку ядра для библиотеки пользовательского пространства, которая реализует интерфейсы очереди сообщений POSIX.
- `fusectl` - это простой интерфейс для программ пользовательского пространства для экспорта виртуальной файловой системы в ядро Linux.
- `configf` - Виртуальная файловая система, представляющая состояние ядра операционной системы и запущенных процессов в виде файлов.
- `ext4` - журналируемая файловая система, используемая преимущественно в операционных системах с ядром Linux, созданная на базе `ext3` в 2006 году.
- `fuse.gv`, `fuse.po`, `fuse.je` - FUSE (файловая система в пользовательском пространстве) — это интерфейс для программ пользовательского пространства для экспорта файловой системы в ядро Linux.

2. Приведите общую структуру файловой системы и дайте характеристику каждой ди- ректории первого уровня этой структуры.

`/` — root каталог. Содержит в себе всю иерархию системы;

`/bin` — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: `pwd`, `ls`, `cat`, `ps`);

`/boot` — тут расположены файлы, используемые для загрузки системы (образ `initrd`, ядро `vmlinuz`);

`/dev` — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск,

можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;

/etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;

/home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;

/lib — содержит системные библиотеки, с которыми работают программы и модули ядра;

/lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;

/media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;

/mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;

/opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);

/proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;

/root — директория, которая содержит файлы и личные настройки суперпользователя;

/run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;

/sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;

/srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);

/sys — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;

/tmp — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;

/usr — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой;

/var — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.

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

Монтирование тома.

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

Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:

1. Один блок адресуется несколькими mode (принадлежит нескольким файлам).
  2. Блок помечен как свободный, но в то же время занят (на него ссылается onode).
  3. Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).
  4. Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).
  5. Несовпадение между размером файла и суммарным размером адресуемых inode блоком.
  6. Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
  7. "Потерянные" файлы (правильные inode, на которые не ссылаются записи каталогов).
  8. Недопустимые или неразмещенные номера inode в записях каталогов.
5. Как создаётся файловая система?
    - mkfs - позволяет создать файловую систему Linux.

Создать файловую систему linux, семейства ext, на устройстве можно с помощью команды mkfs. Ее синтаксис выглядит следующим образом:

```
sudo mkfs -t тип устройства
```

Доступны дополнительные параметры:

- c - проверить устройство на наличие битых секторов
  - b - размер блока файловой системы
  - j - использовать журналирование для ext3
  - L - задать метку раздела
  - v - показать подробную информацию о процессе работы
  - V - версия программы
6. Дайте характеристику командам для просмотра текстовых файлов.
    - cat <имя\_файла> - Это самая простая и, пожалуй, самая популярная команда для просмотра файла в Linux. Cat просто печатает содержимое файла на стандартном экране, т.е. на экране. В основном используется для небольших файлов.
    - Less <имя\_файла>. Команда Less просматривает файл по одной странице за раз.
    - Head <имя\_файла>. Команда Head — это еще один способ просмотра текстового файла, но с небольшой разницей. Команда head отображает первые 10 строк текстового файла по умолчанию. Вы можете изменить это поведение, используя опции с командой head, но основной принцип остается тем же: команда head начинает работать с заголовка (начала) файла.

- `Tail <имя_файла>`. Команда `Tail` в Linux аналогична и все же противоположна команде `head`. В то время как команда `head` отображает файл с начала, команда `tail` отображает файл с конца. По умолчанию команда `tail` отображает последние 10 строк файла. Команды `Head` и `Tail` могут быть объединены для отображения выбранных строк из файла. Вы также можете использовать команду `tail` для просмотра изменений, внесенных в файл в режиме реального времени.

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

Это сокращение от `copy`, и она делает именно то, что предполагает ее название: она копирует. `cp` используется для копирования файлов из одного местоположения в другое. `cp` также можно использовать для копирования всех каталогов в новое место. Можно использовать эту команду для копирования нескольких файлов и каталогов.

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

Команда `mv` используется для перемещения файлов из одного каталога в другой. Также команда `mv` используется для переименования файла в системах Linux.

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

Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами.

Каждый файл можно изменять по трём параметрам доступа. Вот они:

- Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем;
- Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги;
- Выполнение - вы не можете выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу.

Чтобы получить доступ к файлам в Linux, используются разрешения. Эти разрешения назначаются трем объектам: файлу, группе и другому объекту (то есть всем остальным). Изменить права доступа можно при помощи команды **`chmod`**:

`chmod <параметры изменения> <имя_файла/каталога>`