

Отчёт по лабораторной работе №5

Паращенко Антонина Дмитриевна

Содержание

1	Цель работы	5
2	Ход лабораторной работы	6
2.1	Выполнение примеров из текста лабораторной работы	6
3	Вывод	15

Список иллюстраций

2.1	Копируем файл ~/abc1 в файл april и в файл may	6
2.2	Копируем файлы april и may в каталог monthly	6
2.3	Скопируем файл monthly/may в файл с именем june	7
2.4	Скопируем каталог monthly в каталог monthly.00	7
2.5	Изменяем название файла april на july в домашнем каталоге . . .	7
2.6	Перемещаем файл july в каталог monthly.00	8
2.7	Переименовать каталог monthly.00 в monthly.01	8
2.8	Переместить каталог monthly.01 в каталог reports	8
2.9	Переименовать каталог reports/monthly.01 в reports/monthly . . .	8
2.10	Права доступа файла may	8
2.11	Права доступа файла may	9
2.12	Права доступа каталога monthly	9
2.13	Права доступа файла abc1	9
2.14	Копирование файла	9
2.15	Создание директории, перемещение и переименование файла . .	10
2.16	Создание файла, его копирование и переименование	10
2.17	Создание каталога, перемещение файлов	10
2.18	Создание каталога, его перемещение и переименование	11
2.19	Переопределение прав доступа	11
2.20	Переопределение прав доступа	11
2.21	Переопределение прав доступа	11
2.22	Переопределение прав доступа	11
2.23	Файл password	12
2.24	Копирование и перемещение	12
2.25	Права доступа файла feathers	12
2.26	Файл password	13
2.27	Вызов команды man	13
2.28	mount	13
2.29	fsck	14
2.30	mkfs	14
2.31	kill	14

Список таблиц

1 Цель работы

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

2 Ход лабораторной работы

2.1 Выполнение примеров из текста лабораторной работы

1) Копируем файл в текущем каталоге (рис. 2.1)

```
adparathenko@dk3n38 ~ $ cd
adparathenko@dk3n38 ~ $ touch abc1
bash: touch: команда не найдена
adparathenko@dk3n38 ~ $ touch abc1
adparathenko@dk3n38 ~ $ cp abc1 april
adparathenko@dk3n38 ~ $ cd abc1 may
bash: cd: слишком много аргументов
adparathenko@dk3n38 ~ $ cp abc1 may
adparathenko@dk3n38 ~ $ mkdir monthly
adparathenko@dk3n38 ~ $ cp april may monthly
adparathenko@dk3n38 ~ $ cp monthly/may mounthly/june
cp: невозможно создать обычный файл 'mounthly/june': Нет такого файла или каталога
adparathenko@dk3n38 ~ $ cp monthly/may monthly/june
adparathenko@dk3n38 ~ $ ls monthly
april  june  may
adparathenko@dk3n38 ~ $
```

Рис. 2.1: Копируем файл ~/abc1 в файл april и в файл may

2) Копируем несколько файлов в каталог (рис. 2.2)

```
adparathenko@dk3n38 ~ $ cd
adparathenko@dk3n38 ~ $ touch abc1
bash: touch: команда не найдена
adparathenko@dk3n38 ~ $ touch abc1
adparathenko@dk3n38 ~ $ cp abc1 april
adparathenko@dk3n38 ~ $ cd abc1 may
bash: cd: слишком много аргументов
adparathenko@dk3n38 ~ $ cp abc1 may
adparathenko@dk3n38 ~ $ mkdir monthly
adparathenko@dk3n38 ~ $ cp april may monthly
adparathenko@dk3n38 ~ $ cp monthly/may mounthly/june
cp: невозможно создать обычный файл 'mounthly/june': Нет такого файла или каталога
adparathenko@dk3n38 ~ $ cp monthly/may monthly/june
adparathenko@dk3n38 ~ $ ls monthly
april  june  may
adparathenko@dk3n38 ~ $
```

Рис. 2.2: Копируем файлы april и may в каталог monthly

3) Копируем файлы в произвольном каталоге (рис. 2.3)

```
adparathenko@dk3n38 ~ $ cd
adparathenko@dk3n38 ~ $ touch abc1
bash: touch: команда не найдена
adparathenko@dk3n38 ~ $ touch abc1
adparathenko@dk3n38 ~ $ cp abc1 april
adparathenko@dk3n38 ~ $ cd abc1 may
bash: cd: слишком много аргументов
adparathenko@dk3n38 ~ $ cp abc1 may
adparathenko@dk3n38 ~ $ mkdir monthly
adparathenko@dk3n38 ~ $ cp april may monthly
adparathenko@dk3n38 ~ $ cp monthly/may mounthly/june
cp: невозможно создать обычный файл 'mounthly/june': Нет такого файла или каталога
adparathenko@dk3n38 ~ $ cp monthly/may monthly/june
adparathenko@dk3n38 ~ $ ls monthly
april  june  may
adparathenko@dk3n38 ~ $
```

Рис. 2.3: Скопируем файл monthly/may в файл с именем june

4) Копируем каталоги в текущем каталоге (рис. 2.4)

```
adparathenko@dk3n38 ~ $ mkdir monthly.00
adparathenko@dk3n38 ~ $ cp -r monthly monthly.00
adparathenko@dk3n38 ~ $ ls monthly.00
monthly
adparathenko@dk3n38 ~ $ cp -r monthly.00 /tmp
adparathenko@dk3n38 ~ $ ls /tmp
systemd-private-4e8160a995454536a87b23aae0afed91-colorld.service-r3jmye
systemd-private-4e8160a995454536a87b23aae0afed91-systemd-logind.service-ucye03
krb5cc_4262_SfZUEZ systemd-private-4e8160a995454536a87b23aae0afed91-systemd-resolved.service-uNrhFU
krb5cc_4262_xmHKB systemd-private-4e8160a995454536a87b23aae0afed91-systemd-timesyncd.service-1lNrdh
monthly.00 systemd-private-4e8160a995454536a87b23aae0afed91-upower.service-gEDAid
pulse-PKdhtXMmr18n Temp-c6657178-8f2c-4c9a-97b5-ef5c35d0b559
root tmux-0
screen
adparathenko@dk3n38 ~ $
```

Рис. 2.4: Скопируем каталог monthly в каталог monthly.00

5) Копируем каталоги в произвольном каталоге (рис. ??)

Скопируем каталог monthly.00 в каталог /tmp

6) Переименовываем файлы в текущем каталоге (рис. 2.5)

```
adparathenko@dk3n31 ~ $ mv april july
```

Рис. 2.5: Изменяем название файла april на july в домашнем каталоге

7) Перемещаем файлов в другой каталог (рис. 2.6)

```
adparathenko@dk3n31 ~ $ mv july monthly.00
adparathenko@dk3n31 ~ $ ls monthly.00
july  monthly
```

Рис. 2.6: Перемещаем файл july в каталог monthly.00

- 8) Переименовываем каталоги в текущем каталоге (рис. 2.7)

```
adparathenko@dk3n31 ~ $ mv monthly.00 monthly.01
```

Рис. 2.7: Переименовать каталог monthly.00 в monthly.01

- 9) Перемещаем каталог в другой каталог (рис. 2.8)

```
adparathenko@dk3n31 ~ $ mkdir reports
adparathenko@dk3n31 ~ $ mv monthly.01 reports
```

Рис. 2.8: Переместить каталог monthly.01 в каталог reports

- 10) Переименовываем каталог, не являющийся текущим (рис. 2.9)

```
adparathenko@dk3n31 ~ $ mv reports/monthly.01 reports/monthly
```

Рис. 2.9: Переименовать каталог reports/monthly.01 в reports/monthly

- 11) Создаём файл ~/may с правом выполнения для владельца (рис. 2.10)

```
adparathenko@dk3n31 ~/reports/monthly $ touch may
adparathenko@dk3n31 ~/reports/monthly $ ls -l may
-rw-r--r-- 1 adparathenko studsci 0 мая  4 17:19 may
adparathenko@dk3n31 ~/reports/monthly $ chmod u+x may
adparathenko@dk3n31 ~/reports/monthly $ ls -l may
-rwxr--r-- 1 adparathenko studsci 0 мая  4 17:19 may
```

Рис. 2.10: Права доступа файла may

- 12) Лишаем владельца файла may права выполнение (рис. 2.11)


```
adparathenko@dk3n31 ~/reports/monthly $ chmod u-x may
adparathenko@dk3n31 ~/reports/monthly $ ls -l may
-rwx-r--r-- 1 adparathenko studsci 0 мая  4 17:19 may
adparathenko@dk3n31 ~/reports/monthly $ mkdir monthly
```

Рис. 2.11: Права доступа файла may

- 13) Создаём каталог monthly с запретом на чтение для членов группы и всех остальных пользователей (рис. 2.12)

```
adparathenko@dk3n31 ~/reports/monthly $ chmod g-r monthly
adparathenko@dk3n31 ~/reports/monthly $ chmod o-r monthly
adparathenko@dk3n31 ~/reports/monthly $ ls -l monthly
```

Рис. 2.12: Права доступа каталога monthly

- 14) Создаём файл ~/abc1 с правом записи для членов группы. (рис. 2.13)

```
adparathenko@dk3n31 ~/reports/monthly $ touch abc1
adparathenko@dk3n31 ~/reports/monthly $ rm abc1
adparathenko@dk3n31 ~/reports/monthly $ cd
adparathenko@dk3n31 ~ $ touch abc1
adparathenko@dk3n31 ~ $ chmod g+w abc1
adparathenko@dk3n31 ~ $ ls -l abc1
-rwx-rw-r-- 1 adparathenko studsci 0 мая  4 17:28 abc1
```

Рис. 2.13: Права доступа файла abc1

- 15) Скопируем файл /usr/include/sys/io.h в домашний каталог и назовём его equipment (рис. 2.14)

```
adparathenko@dk3n31 ~ $ cp /usr/include/sys/io.h equipment
adparathenko@dk3n31 ~ $ mkdir ski_places
```

Рис. 2.14: Копирование файла

- 16) В домашнем каталоге создаём директорию ~/ski.places. Перемещаем файл equipment в каталог ~/ski.places и переименовываем файл ~/ski.places/equipment в ~/ski.places/equiplist (рис. 2.15)

```

adparathenko@dk3n31 ~ $ mkdir ski.places
adparathenko@dk3n31 ~ $ mv equipment ski.places
adparathenko@dk3n31 ~ $ ls ski.places
equipment
adparathenko@dk3n31 ~ $ mv ski.places/equipment ski.places/equiplist
adparathenko@dk3n31 ~ $ ls
abc1      ski.places      'Снимок экрана от 2022-04-28 18-01-35.png'
bin       temp            'Снимок экрана от 2022-04-28 18-06-35.png'
docs      tmp            'Снимок экрана от 2022-04-28 18-08-09.png'
GNUstep   work           'Снимок экрана от 2022-04-28 18-34-00.png'
image     Видео          'Снимок экрана от 2022-04-28 18-42-04.png'
lab1.zip  Документы     'Снимок экрана от 2022-04-28 18-44-08.png'
may       Загрузки      'Снимок экрана от 2022-04-28 18-45-20.png'
monthly   Изображения   'Снимок экрана от 2022-04-28 18-46-47.png'
public    Музыка        'Снимок экрана от 2022-04-28 18-48-27.png'
public_html  Общедоступные 'Снимок экрана от 2022-04-28 18-51-51.png'
reports   'Рабочий стол' 'Снимок экрана от 2022-04-28 18-58-11.png'
adparathenko@dk3n31 ~ $ ls ski.places
equiplist

```

Рис. 2.15: Создание директории, перемещение и переименование файла

- 17) Создаём в домашнем каталоге файл abc1 и копируем его в каталог ~/ski.places, называем его equiplist2. (рис. 2.16)

```

adparathenko@dk3n31 ~ $ cp abc1 ski.places
adparathenko@dk3n31 ~ $ mv ski.places/abc1 ski.places/equiplist2
adparathenko@dk3n31 ~ $ cd ski.places
adparathenko@dk3n31 ~/ski.places $ ls
equiplist equiplist2
adparathenko@dk3n31 ~/ski.places $ cd

```

Рис. 2.16: Создание файла, его копирование и переименование

- 18) Создаём каталог с именем equipment в каталоге ~/ski.places. Перемещаем файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment (рис. 2.17)

```

adparathenko@dk3n31 ~ $ mv ski.places/equiplist ski.places/equiplist2 ski.places/equipment
adparathenko@dk3n31 ~ $ ls equipment
ls: невозможно получить доступ к 'equipment': Нет такого файла или каталога
adparathenko@dk3n31 ~ $ ls ski.places/equipment
equiplist equiplist2

```

Рис. 2.17: Создание каталога, перемещение файлов

- 19) Создаём и перемещаем каталог ~/newdir в каталог ~/ski.places и называем его plans. (рис. 2.18)

```
adparathenko@dk3n31 ~ $ mkdir newdir
adparathenko@dk3n31 ~ $ mv newdir ski.places
adparathenko@dk3n31 ~ $ mv newdir plans
mv: не удалось выполнить stat для 'newdir': Нет такого файла или каталога
adparathenko@dk3n31 ~ $ mv ski.places/newdir plans
```

Рис. 2.18: Создание каталога, его перемещение и переименование

- 20) Определяем опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа. (рис. 2.19) - (рис. 2.22)

```
adparathenko@dk3n31 ~ $ chmod u+rwx, g+r, o+r australia
```

Рис. 2.19: Переопределение прав доступа

```
adparathenko@dk3n31 ~ $ chmod u+rwx, g+x, o+x play
```

Рис. 2.20: Переопределение прав доступа

```
adparathenko@dk3n31 ~ $ chmod u+rx, g+r, o+r my_os
```

Рис. 2.21: Переопределение прав доступа

```
adparathenko@dk3n31 ~ $ chmod u+rw, g+rw, o+r feathers
```

Рис. 2.22: Переопределение прав доступа

- 21) Просматриваем содержимое файла `/etc/passwd`. (рис. 2.13)

```
adparathenko@dk3n31 ~ $ cat /etc/passwd
root:x:0:0:System user; root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
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 program user:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucppublic:/bin/false
operator:x:11:0:operator:/root:/bin/bash
man:x:13:15:System user; man:/dev/null:/sbin/nologin
postmaster:x:14:12:Postmaster user:/var/spool/mail:/sbin/nologin
cron:x:16:16:A user for sys-process/cronbase:/var/spool/cron:/sbin/nologin
ftp:x:21:21:./home/ftp:/bin/false
sshd:x:22:22:User for ssh:/var/empty:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/bin/false
squid:x:31:31:Squid:/var/cache/squid:/bin/false
gdm:x:32:32:User for running GDM:/var/lib/gdm:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/bin/false
games:x:35:35:games:/usr/games:/bin/bash
named:x:40:40:bind:/var/bind:/bin/false
mysql:x:60:60:MySQL program user:/dev/null:/sbin/nologin
postgres:x:70:70:PostgreSQL program user:/var/lib/postgresql:/bin/sh
```

Рис. 2.23: Файл password

- 22) Скопируем файл ~/feathers в файл ~/file.old. Перемещаем файл ~/file.old в каталог ~/play. Скопируем каталог ~/play в каталог ~/fun. Перемещаем каталог ~/fun в каталог ~/play и называем его games (рис. 2.24)

```
adparathenko@dk3n31 ~ $ cp feathers file.old
adparathenko@dk3n31 ~ $ mv file.old /play
mv: невозможно создать обычный файл '/play': Отказано в доступе
adparathenko@dk3n31 ~ $ mv file.old play
adparathenko@dk3n31 ~ $ cp -r play fun
adparathenko@dk3n31 ~ $ mv fun games
adparathenko@dk3n31 ~ $ mv games play
```

Рис. 2.24: Копирование и перемещение

- 23) Лишаем владельца файла ~/feathers права на чтение и видим, что мы не имеем доступ, чтобы прочитать файл (рис. 2.25)

```
adparathenko@dk3n31 ~ $ chmod u-r feathers
adparathenko@dk3n31 ~ $ ls -l feathers
--w-r--r-- 1 adparathenko studsci 0 мая  4 18:00 feathers
adparathenko@dk3n31 ~ $ cdt feathers
bash: cdt: команда не найдена
adparathenko@dk3n31 ~ $ cat feathers
cat: feathers: Отказано в доступе
```

Рис. 2.25: Права доступа файла feathers

- 24) Лишаем владельца каталога ~/play права на выполнение и видим, что права доступа к каталогу изменились. (рис. 2.26)

```
adparathenko@dk3n31 ~ $ chmod u-x play
```

Рис. 2.26: Файл password

- 25) Прочитаем man по командам mount, fsck, mkfs, kill. (рис. 2.27) - (рис. 2.31)

```
adparathenko@dk3n31 ~ $ man mount
adparathenko@dk3n31 ~ $ man fsck
adparathenko@dk3n31 ~ $ man mkfs
adparathenko@dk3n31 ~ $ man kill
adparathenko@dk3n31 ~ $
```

Рис. 2.27: Вызов команды man

```
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.

If only the directory or the device is given, for example:

    mount /dir

then mount looks for a mountpoint (and if not found then for a device) in the /etc/fstab file. It's
possible to use the --target or --source options to avoid ambiguous interpretation of the given argument.
For example:

    mount --target /mountpoint

The same filesystem may be mounted more than once, and in some cases (e.g., network filesystems) the same
filesystem may be mounted on the same mountpoint multiple times. The mount command does not implement any
policy to control this behavior. All behavior is controlled by the kernel and it is usually specific to the
filesystem driver. The exception is --all, in this case already mounted filesystems are ignored (see --all
below for more details).
```

Рис. 2.28: mount


```

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=8868abfe-88c5-4a83-98b8-bfc2405777bd 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 bit-wise OR of the exit statuses 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 the PATH environment variable. If the PATH is
undefined then fallback to /sbin.

Please see the filesystem-specific checker manual pages for further details.

```

Рис. 2.29: fsck

```

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.

```

Рис. 2.30: mkfs

```

DESCRIPTION
The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful signals
include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: -9, -SIGKILL
or -KILL. Negative PID values may be used to choose whole process groups; see the PGID column in ps com-
mand output. A PID of -1 is special; it indicates all processes except the kill process itself and init.

```

Рис. 2.31: kill

3 Вывод

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.