

Отчет по лабораторной работе №6

Дисциплина: Операционные системы

Пермякова Елизавета Евгеньевна

Содержание

1	Цель работы	4
2	Задачи	5
3	Выполнение лабораторной работы	6
4	Контрольные вопросы	19
5	Выводы	28

List of Figures

3.1	Копирование файлов и каталогов	7
3.2	Перемещение и переименование файлов и каталогов	8
3.3	Изменение прав доступа	9
3.4	Создаем, перемещаем и переименовываем файлы и каталоги . .	10
3.5	Создание необходимых файлов	10
3.6	Опции команды chmod	11
3.7	Просмотрим содержимое файла /etc/password.	13
3.8	Лишаем владельца файла или каталога прав и даем их ему	13
3.9	Команда man mount	15
3.10	Команда man fsck	16
3.11	Команда man mkfs	17
3.12	Команда man kill	18
4.1	Команда df -Th	19

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 Задачи

1. Изучить команды для работы с файлами и каталогами.
2. Изучить команды для копирования файлов и каталогов.
3. Изучить команды для перемещения и переименования файлов и каталогов.
4. Изучить права доступа файлов или каталогов.
5. В ходе работы использовать эти команды и интерпретировать их вывод.
6. Выполнить отчет.

3 Выполнение лабораторной работы

- 1) Для начала выполним примеры, описанные в первой части описания лабораторной работы.

Рисунок 1(рис. 3.1):

1. Скопируем файл ~/abc1 в файл april и в файл may. Для этого создадим файл abc1, используя команду «touch abc1», далее осуществим копирование с помощью команд «cp abc1 april» и «cp abc1 may».
2. Скопируем файлы april и may в каталог monthly, используя команды «mkdir monthly» – для создания каталога monthly и «cp april may monthly» – для копирования.
3. Скопируем файл monthly/may в файл с именем june. Выполним команды «cp monthly/may monthly/june» и «ls monthly» (для просмотра содержимого каталога).
4. Скопируем каталог monthly в каталог monthly.00. Для этого создадим каталог monthly.00 командой «mkdir monthly.00» и осуществим копирование, используя команду «cp -r monthly monthly.00» (команда cp с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами).
5. Скопируем каталог monthly.00 в каталог /tmp, используя команду «cp -r monthly.00 /tmp».

```
~ : bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
eepерmyakova@dk3n59 ~ $ cd
eepерmyakova@dk3n59 ~ $ touch abcl
eepерmyakova@dk3n59 ~ $ cp abcl april
eepерmyakova@dk3n59 ~ $ cp abcl may
eepерmyakova@dk3n59 ~ $ mkdir monthly
eepерmyakova@dk3n59 ~ $ cp april may monthly
eepерmyakova@dk3n59 ~ $ cp monthly/may monthly/june
eepерmyakova@dk3n59 ~ $ ls monthly
april  june  may
eepерmyakova@dk3n59 ~ $ mkdir monthly.00
eepерmyakova@dk3n59 ~ $ cp -r monthly monthly.00
eepерmyakova@dk3n59 ~ $ cp -r monthly.00 /tmp
```

Figure 3.1: Копирование файлов и каталогов

Рисунок 2(рис. 3.2):

1. Изменим название файла `april` на `july` в домашнем каталоге, используя команду «`mv april july`».
2. Переместим файл `july` в каталог `monthly.00` с помощью команды «`mv july monthly.00`». Проверим результат командой «`ls monthly.00`».
3. Переименуем каталог `monthly.00` в `monthly.01`, используя команду «`mv monthly.00 monthly.01`».
4. Переместим каталог `monthly.01` в каталог `reports`. Для этого создадим каталог `reports` с помощью команды «`mkdir reports`» и выполним перемещение командой «`mv monthly.01 reports`».
5. Переименуем каталог `reports/monthly.01` в `reports/monthly` командой «`mv reports/monthly.01 reports/monthly`».

```

eepermyakova@dk3n59 ~ $ cd
eepermyakova@dk3n59 ~ $ mv april july
eepermyakova@dk3n59 ~ $ mv july monthly.00
eepermyakova@dk3n59 ~ $ ls monthly.00
july  monthly
eepermyakova@dk3n59 ~ $ mv monthly.00 monthly.01
eepermyakova@dk3n59 ~ $ mkdir reports
eepermyakova@dk3n59 ~ $ mv monthly.01 reports
eepermyakova@dk3n59 ~ $ mv reports/monthly.01 reports/monthly
eepermyakova@dk3n59 ~ $

```

Figure 3.2: Перемещение и переименование файлов и каталогов

Рисунок 3(рис. 3.3):

1. Создадим файл ~/may с правом выполнения для владельца. Для этого выполним следующие команды: «touch may» (создание файла), «ls -l may» (просмотр сведений о файле), «chmod u+x may» (изменение прав), «ls -l may».
2. Лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may» (изменение прав), «ls -l may» (просмотр сведений о файле).
3. Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly» (создание каталога), «chmod go-r monthly» (изменение прав).
4. Создаем файл ~/abcl с правом записи для членов группы, используя команды: «touch abcl» (создание файла), «chmod g+w abcl» (изменение прав).


```

eepermyakova@dk3n59 ~ $ cd
eepermyakova@dk3n59 ~ $ touch may
eepermyakova@dk3n59 ~ $ ls -l may
-rw-r--r-- 1 eepermyakova studsci 0 мая 12 09:49 may
eepermyakova@dk3n59 ~ $ chmod u+x may
eepermyakova@dk3n59 ~ $ ls -l may
-rwxr--r-- 1 eepermyakova studsci 0 мая 12 09:49 may
eepermyakova@dk3n59 ~ $ chmod u-x may
eepermyakova@dk3n59 ~ $ ls -l may
-rw-r--r-- 1 eepermyakova studsci 0 мая 12 09:49 may
eepermyakova@dk3n59 ~ $ cd
eepermyakova@dk3n59 ~ $ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
eepermyakova@dk3n59 ~ $ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
eepermyakova@dk3n59 ~ $ chmod go-r monthly
eepermyakova@dk3n59 ~ $ cd
eepermyakova@dk3n59 ~ $ touch abcl
eepermyakova@dk3n59 ~ $ chmod g+w abcl
eepermyakova@dk3n59 ~ $

```

Figure 3.3: Изменение прав доступа

2) Выполняем следующие действия (рис. 3.4):

1. Копируем файл `/usr/include/sys/io.h` в домашний каталог (команда «`cp /usr/include/sys/io.h ~`») и называемого `equipment` (команда «`mv io.h equipment`»).
2. В домашнем каталоге создаем директорию `~/ski.places` (команда «`mkdir ski.places`»).
3. Перемещаем файл `equipment` в каталог `~/ski.places` (команда «`mv equipment ski.places`»).
4. Переименовываем файл `~/ski.places/equipment` в `~/ski.places/equiplist` (команда «`mv ski.places/equipment ski.places/equiplist`»).
5. Создаем в домашнем каталоге файл `abcl` (команда «`touch abcl`») и копируем его в каталог `~/ski.places` (команда «`cp abcl ski.places`»), называемого `equiplist2` (команда «`mv ski.places/abcl ski.places/equiplist2`»).
6. Создаем каталог с именем `equipment` в каталоге `~/ski.places` (команда «`mkdir ski.places/equipment`»).

7. Перемещаем файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment (команда «mv ski.places/equiolist ski.places/equiplist2 ski.places/equipment»).
8. Создаем (команда «mkdir newdir») и перемещаем каталог ~/newdir в каталог ~/ski.places (команда «mv newdir ski.places») и называем его plans (команда «mv ski.places/newdir ski.places/plans»)

```

eepermyakova@dk3n59 ~ $ cp /usr/include/sys/io.h ~
eepermyakova@dk3n59 ~ $ mv io.h equipment
eepermyakova@dk3n59 ~ $ mkdir ski.places
eepermyakova@dk3n59 ~ $ mv equipment ski.places
eepermyakova@dk3n59 ~ $ mv ski.places/equipment ski.places/equiplist
eepermyakova@dk3n59 ~ $ touch abcl
eepermyakova@dk3n59 ~ $ cp abcl ski.places
eepermyakova@dk3n59 ~ $ mv ski.places/abcl ski.places/equiplist2
eepermyakova@dk3n59 ~ $ mkdir ski.places/equipment
eepermyakova@dk3n59 ~ $ mv ski.places/equiplist ski.places/equiplist2 ski.places/equipment
eepermyakova@dk3n59 ~ $ mkdir newdir
mkdir: невозможно создать каталог «newdir»: Файл существует
eepermyakova@dk3n59 ~ $ mv newdir ski.places
eepermyakova@dk3n59 ~ $ mv ski.places/newdir ski.places/plans

```

Figure 3.4: Создаем, перемещаем и переименовываем файлы и каталоги

- 3) Определяем опции команды chmod, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем необходимые файлы, используя команды: «mkdir australia», «mkdir play», «touch my_os», «touch feathers». (рис. 3.5)

```

eepermyakova@dk3n59 ~ $ mkdir australia
eepermyakova@dk3n59 ~ $ mkdir play
eepermyakova@dk3n59 ~ $ touch my_os
eepermyakova@dk3n59 ~ $ touch feathers

```

Figure 3.5: Создание необходимых файлов

1. drwxr-r-... australia: команда «chmod 744 australia» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение).
2. drwx-x-x ... play: команда «chmod 711 play» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение).

3. -r-xr-r-... my_os: команды «chmod 544 my_os» (это файл, владелец имеет право на чтение и выполнение, группа владельца и остальные – только чтение).
4. -rw-rw-r-... feathers: команды «chmod 664 feathers» (это файл, владелец и группа владельца имеют право на чтение и запись, остальные – только чтение)

Командой «ls-l» проверяем правильность выполненных действий. (рис. 3.6)

```
eepermyakova@dk3n59 ~$ chmod 744 australia
eepermyakova@dk3n59 ~$ chmod 711 play
eepermyakova@dk3n59 ~$ chmod 544 my_os
eepermyakova@dk3n59 ~$ chmod 664 feathers
eepermyakova@dk3n59 ~$ ls -l
итого 1828
-rw-r--r-- 1 eepermyakova studsci 1726649 апр 29 11:56 2_Laboratornaya_rabota.pptx
-rw-rw-r-- 1 eepermyakova studsci 0 мая 12 10:04 abcl
drwxr-xr-x 4 eepermyakova studsci 2048 апр 30 13:23 academic-laboratory-report-template
drwxr-xr-x 4 eepermyakova studsci 2048 апр 30 13:01 academic-presentation-markdown-template
-rwxr-xr-x 1 eepermyakova studsci 1120 сен 24 2020 asdfg
-rw-r--r-- 1 eepermyakova studsci 2650 сен 24 2020 asdfg.asm
-rwxr-xr-x 1 eepermyakova studsci 1112 сен 24 2020 asfdg
-rw-r--r-- 1 eepermyakova studsci 2403 сен 24 2020 asfdg.asm
drwxr--r-- 2 eepermyakova studsci 2048 мая 12 10:10 australia
-rw-rw-r-- 1 eepermyakova studsci 0 мая 12 10:10 feathers
drwxr-xr-x 2 eepermyakova studsci 2048 апр 29 14:39 image3
drwxr-xr-x 2 eepermyakova studsci 2048 апр 29 19:26 lab02
drwxr-xr-x 2 eepermyakova studsci 2048 апр 29 12:55 lab03a
drwxr-xr-x 2 eepermyakova studsci 2048 фев 3 12:39 lab03b
-rwxr-xr-x 1 eepermyakova studsci 1328 окт 9 2020 lab05
-rw-r--r-- 1 eepermyakova studsci 3784 окт 9 2020 lab05.asm
-rwxr-xr-x 1 eepermyakova studsci 1288 окт 13 2020 lab06
-rw-r--r-- 1 eepermyakova studsci 3278 окт 13 2020 lab06.asm
-rw-r--r-- 1 eepermyakova studsci 1488 окт 16 2020 lab07
-rwxr-xr-x 1 eepermyakova studsci 4470 окт 16 2020 lab07.asm
-rwxr-xr-x 1 eepermyakova studsci 936 сен 17 2020 lab2
-rw-r--r-- 1 eepermyakova studsci 1030 сен 17 2020 lab2.asm
drwxr-xr-x 5 eepermyakova studsci 2048 апр 29 16:25 lab3
drwxr-xr-x 4 eepermyakova studsci 2048 апр 30 10:49 laboratory
drwxr-xr-x 2 eepermyakova studsci 2048 мая 11 09:26 lessfun
-rw-r--r-- 1 eepermyakova studsci 0 мая 12 09:49 may
drwx--x--x 2 eepermyakova studsci 2048 мая 12 09:41 monthly
-r-xr--r-- 1 eepermyakova studsci 0 мая 12 10:10 my_os
drwx--x--x 2 eepermyakova studsci 2048 мая 12 10:10 play
```

Figure 3.6: Опции команды chmod

- 4) Выполняем следующие действия (рис. 3.7) (рис. 3.8):
 1. Просмотрим содержимое файла /etc/passwd (команда «cat /etc/passwd»).
 2. Копируем файл ~/feathers в файл ~/file.old (команда «cp feathers file.old»).
 3. Переместим файл ~/file.old в каталог ~/play (команда «mv file.old play»).
 4. Скопируем каталог ~/play в каталог ~/fun(команда «cp -r play fun»).

5. Переместим каталог ~/fun в каталог ~/play (команда «mv fun play»)и назовем его games(команда «mv play/fun play/games»).
6. Лишим владельца файла ~/feathers права на чтение(команда «chmod u-rfeathers»).
7. Если мы попытаемся просмотреть файл ~/feathers командой cat, то получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на чтение данного файла.
8. Если мы попытаемся скопировать файл ~/feathers, например, в каталог monthly, то получим отказ в доступе, по причине, описанной в предыдущем пункте.
9. Дадим владельцу файла ~/feathers право на чтение (команда «chmod u+rfeathers»).
10. Лишим владельца каталога ~/play права на выполнение (команда «chmod u-x play»).
11. Перейдем в каталог ~/play (команда «cdplay»). Мы не получили отказ в доступе, несмотря на то, что в предыдущем пункте лишили владельца права на выполнение данного каталога.
12. Дадим владельцу каталога ~/play право на выполнение(команда «chmod u+x play»).

```

eepermyakova@dk3n59 ~ $ cat /etc/passwd
cat: /etc/passwd: Нет такого файла или каталога
eepermyakova@dk3n59 ~ $ cat /etc/passwd
root:x:0:0: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:cron:/var/spool/cron:/bin/false
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:GDM:/var/lib/gdm:/bin/false
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
nut:x:84:84:nut:/var/state/nut:/bin/false
cyrus:x:85:12:./usr/cyrus:/bin/false
vpopmail:x:89:89:./var/vpopmail:/bin/false
alias:x:200:200:./var/qmail/alias:/bin/false
qmaild:x:201:200:./var/qmail:/bin/false
qmail1:x:202:200:./var/qmail:/bin/false
qmailp:x:203:200:./var/qmail:/bin/false
qmailq:x:204:201:./var/qmail:/bin/false
qmailr:x:205:201:./var/qmail:/bin/false
qmails:x:206:201:./var/qmail:/bin/false
postfix:x:207:207:postfix:/var/spool/postfix:/bin/false
smmsp:x:209:209:smmsp:/var/spool/mqueue:/bin/false
portage:x:250:250:System user; portage:/var/tmp/portage:/sbin/nologin
nobody:x:65534:65534:nobody:/bin/false
sockd:x:101:2:added by portage for dante:/etc/socks:/bin/false
rpc:x:111:111:added by portage for portmap:/dev/null:/bin/false
ntp:x:123:123:user for ntp daemon:/dev/null:/sbin/nologin
ldap:x:439:439:added by portage for openldap:/usr/lib/openldap:/bin/false

```

Figure 3.7: Просмотрим содержимое файла /etc/passwd.

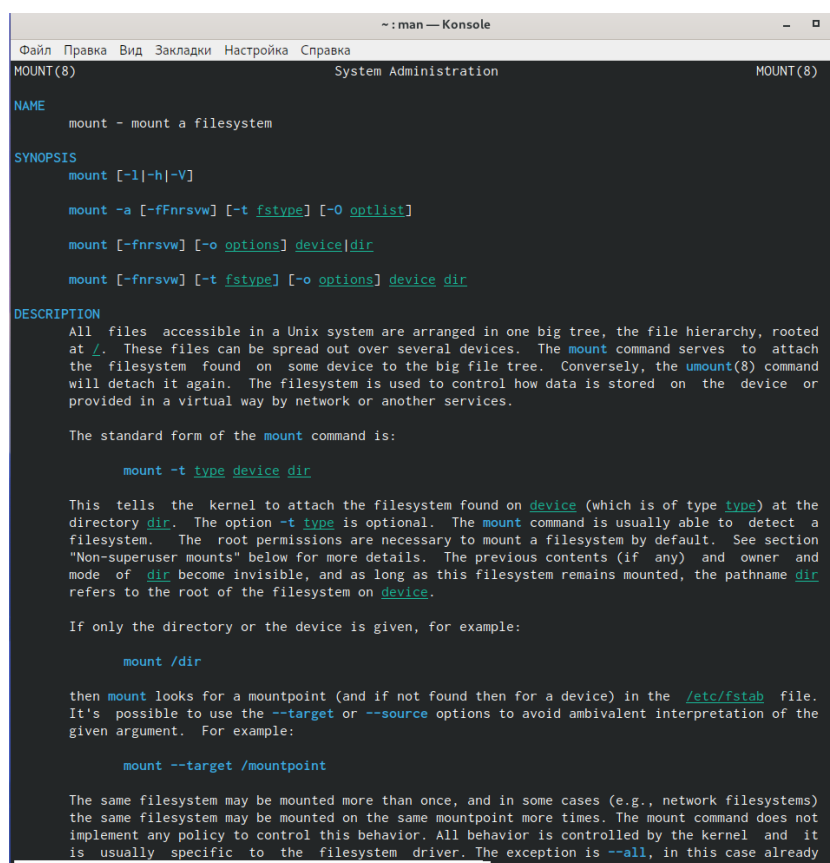
```

eepermyakova@dk3n59 ~ $ cp feathers file.old
eepermyakova@dk3n59 ~ $ mv file.old play
eepermyakova@dk3n59 ~ $ cp -r play fun
eepermyakova@dk3n59 ~ $ mv fun play
eepermyakova@dk3n59 ~ $ mv play/fun play/games
eepermyakova@dk3n59 ~ $ chmod u-r feathers
eepermyakova@dk3n59 ~ $ cat feathers
cat: feathers: Отказано в доступе
eepermyakova@dk3n59 ~ $ cp feathers monthly
ср: невозможно открыть 'feathers' для чтения: Отказано в доступе
eepermyakova@dk3n59 ~ $ chmod u+r feathers
eepermyakova@dk3n59 ~ $ chmod u-x play
eepermyakova@dk3n59 ~ $ cd play
eepermyakova@dk3n59 ~/play $ cd ~
eepermyakova@dk3n59 ~ $ chmod u+x play
eepermyakova@dk3n59 ~ $ 

```

Figure 3.8: Лишаем владельца файла или каталога прав и даем их ему

- 5) Используя команды «man mount», «man fsck», «man mkfs», «man kill», получим информацию о соответствующих командах.
1. Команда mount: предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого дерева обозначается как /. Физически файлы могут располагаться на различных устройствах. Команда mount служит для подключения файловых систем разных устройств к этому большому дереву. Наиболее часто встречающаяся форма команды mount выглядит следующим образом: «mount -t vfstype device dir» Такая команда предлагает ядру смонтировать (подключить) файловую систему указанного типа vfstype, расположенную на устройстве device, к заданному каталогу dir, который часто называют точкой монтирования. (рис. 3.9)



```
~: man — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
MOUNT(8)                                     System Administration                                     MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-l|-h|-V]

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

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

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

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 another 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 ambivalent 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 more 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
```

Figure 3.9: Команда man mount

2. Команда fsck: это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Он использует программы, специфичные для типа файловой системы, которую он проверяет. У команды fsck следующий синтаксис:

`fsck [параметр] -[параметры ФС] [...]`

Например, если нужно восстановить («починить») файловую систему на некотором устройстве `/dev/sdb2`, следует воспользоваться командой: «`sudo fsck -y /dev/sdb2`». Опция `-y` необходима, т.к. при её отсутствии придётся слишком часто давать подтверждение.(рис. 3.10)

```
~:man - Konsole
Файл Правка Вид Закладки Настройка Справка
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. filesystems can be a device name (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the fsck program will try to handle filesystems on different physical disk drives in parallel to reduce the total amount of time needed to check all of them.

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

    The exit 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 bit-wise 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 the PATH environment variable. If the PATH is undefined then fallback to "/sbin".

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

OPTIONS
    -l      Create an exclusive flock(2) lock file (/run/fsck/<diskname>.lock) for whole-disk device. This option can be used with one device only (this means that -A and -l are mutually exclusive). This option is recommended when more fsck(8) instances are executed in the same time. The option is ignored when used for multiple devices or for non-rotating disks. fsck does not lock underlying devices when executed to check stacked devices (e.g., MD or DM) - this feature is not implemented yet.
```

Figure 3.10: Команда man fsck

3. Команда mkfs: создаёт новую файловую систему Linux. Имеет следующий синтаксис:

`mkfs[-V] [-tfstype] [fs-options]filesys[blocks]`

mkfs используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента filesys для файловой системы может выступать или название устройства (например, /dev/hda1, /dev/sdb2) или точка монтирования (например, /, usr, home). Аргументом blocks указывается количеств блоков, которые выделяются для использования этой файловой системой. По окончании работы mkfs возвращает 0 - в случае успеха, а 1 - при неудачной операции. Например, команда «mkfs -t ext2 /dev/hdb1» создаёт файловую систему типа ext2 в разделе /dev/hdb1 (второй жёсткий диск). (рис. 3.11)


```
MKFS(8)                                System Administration                                MKFS(8)

NAME
    mkfs - build a Linux filesystem

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

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

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

    The exit 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 via your PATH environment setting only. Please see the filesystem-specific builder manual pages for further details.

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

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

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

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

    -h, --help
        Display help text and exit.

BUGS
    All generic options must precede and not be combined with filesystem-specific options. Some filesystem-specific programs do not automatically detect the device size and require the size parameter to be specified.
```

Figure 3.11: Команда man mkfs

4. Команда kill: посылает сигнал процессу или выводит список допустимых сигналов. Имеет следующий синтаксис: kill [опции] PID, где PID – это PID (числовой идентификатор) процесса или несколько PID процессов, если требуется послать сигнал сразу нескольким процессам. Например, команда «kill -KILL 3121» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс. (рис. 3.12)

```
~: man — Konsole
Файл Правка Вид Закладки Настройка Справка
KILL(1) User Commands KILL(1)

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Particularly use-
    ful 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 command output. A PID of -1 is special; it indicates all pro-
    cesses except the kill process itself and init.

OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<signal>
    -s <signal>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or number. The
        behavior of signals is explained in signal(7) manual page.

    -l, --list [signal]
        List signal names. This option has optional argument, which will convert signal number
        to signal name, or other way round.

    -L, --table
        List signal names in a nice table.

NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may need to
    run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -9 -1
        Kill all processes you can kill.

    kill -l 11
        Translate number 11 into a signal name.

    kill -L
        List the available signal choices in a nice table.

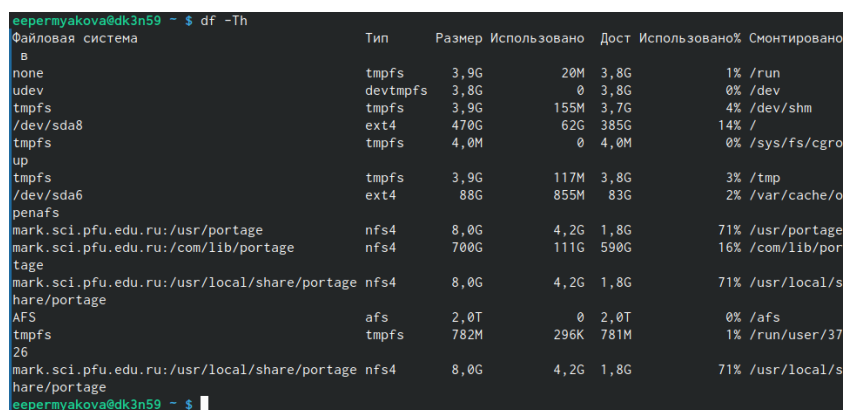
    kill 123 543 2341 3453

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

Figure 3.12: Команда man kill

4 Контрольные вопросы

- 1) Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th».(рис. 4.1)



Файловая система	Тип	Размер	Использовано	Дост	Использовано%	Смонтировано
none	tmpfs	3,9G	20M	3,8G	1%	/run
udev	devtmpfs	3,8G	0	3,8G	0%	/dev
tmpfs	tmpfs	3,9G	155M	3,7G	4%	/dev/shm
/dev/sda8	ext4	470G	62G	385G	14%	/
tmpfs	tmpfs	4,0M	0	4,0M	0%	/sys/fs/cgro
up						
tmpfs	tmpfs	3,9G	117M	3,8G	3%	/tmp
/dev/sda6	ext4	88G	855M	83G	2%	/var/cache/o
penafs						
mark.sci.pfu.edu.ru:/usr/portage	nfs4	8,0G	4,2G	1,8G	71%	/usr/portage
mark.sci.pfu.edu.ru:/com/lib/portage	nfs4	700G	111G	590G	16%	/com/lib/por
tage						
mark.sci.pfu.edu.ru:/usr/local/share/portage	nfs4	8,0G	4,2G	1,8G	71%	/usr/local/s
hare/portage						
AFS	afs	2,0T	0	2,0T	0%	/afs
tmpfs	tmpfs	782M	296K	781M	1%	/run/user/37
26						
mark.sci.pfu.edu.ru:/usr/local/share/portage	nfs4	8,0G	4,2G	1,8G	71%	/usr/local/s
hare/portage						

Figure 4.1: Команда df -Th

Из рисунка видно, что на моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4.

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

также предназначенная для быстрого и ненадёжного хранения временных данных. Хорошо 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.

ext4- имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «extent file system», который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (delayed allocation – delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных.

Характеристики:

- максимальный размер файла: 16 ТБ;
- максимальный размер раздела: 16 ТБ;
- максимальный размер имени файла: 255 символов.

Рекомендации по использованию:

- наилучший выбор для SSD;
- наилучшая производительность по сравнению с предыдущими Ext-системами;
- она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3.

2)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, флеш-ки или оптические диски подключаются в папки корневого каталога.

Только пользователь `root` имеет право читать и изменять файлы в этом каталоге.

`/BIN` –бинарные файлы пользователя

Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления.

`/SBIN` –системные исполняемые файлы

Так же как и `/bin`, содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки, когда не примонтирован каталог `/usr`. Но здесь находятся программы, которые можно выполнять только с правами суперпользователя.

`/ETC` –конфигурационные файлы

В этой папке содержатся конфигурационные файлы всех программ, установленных в системе.Кроме конфигурационных файлов, в системе инициализации `Init Scripts`, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ.

`/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.

/ 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.

/ 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 кеши, базы данных и так далее.

/ TMP–временные файлы

В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию.

/USR –программы пользователя

Это самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию.

/ HOME –домашняя папка

В этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т.д.

/ BOOT –файлы загрузчика

Содержит все файлы,связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящие в каталоге /boot/grub.

/ LIB –системные библиотеки

Содержит файлы системных библиотек, которые используются исполняемыми

файлами в каталогах /bin и /sbin.

/ OPT–дополнительные программы

В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями.

/ MNT –монтирование

В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы.

/ MEDIA –съёмные носители

В этот каталог система монтирует все подключаемые внешние накопители –USB флешки, оптические диски и другие носители информации.

/ SRV –сервер

В этом каталоге содержатся файлы серверов и сервисов.

/ 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.

3)Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount.

4)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)Файловую систему можно создать, используя команду mkfs. Ее краткое описание дано в пункте 5) в ходе выполнения заданий лабораторной работы.

6)Для просмотра текстовых файлов существуют следующие команды:

- cat

Задача команды cat очень проста – она читает данные из файла или стандартного ввода и выводит их на экран. Синтаксис утилиты:

cat [опции] файл1 файл2 ...

Основные опции:

- 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

- nl

Команда nl действует аналогично команде cat, но выводит еще номера строк в столбце слева.

- 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 аналогичный синтаксису команды cat.

Некоторые опции:

- 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:

- 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

- tail

Эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды cat.

Основные опции:

-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) Утилита 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)Командаmvиспользуется для перемещения одного или нескольких файлов (или директорий) вдругую директорию, атакже для переименования файлов идиректорий.

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) 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 чтения

w запись

x выполнение

u (user) владелец файла

g (group) группа, к которой принадлежит владелец файла

o (others) все остальные

5 Выводы

В ходе выполнения данной лабораторной работы я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов, получила навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.