

Лабораторная работа №6

Дисциплина: Операционные системы

Колчева Юлия Вячеславовна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	15
4	Контрольные вопросы	16

List of Tables

List of Figures

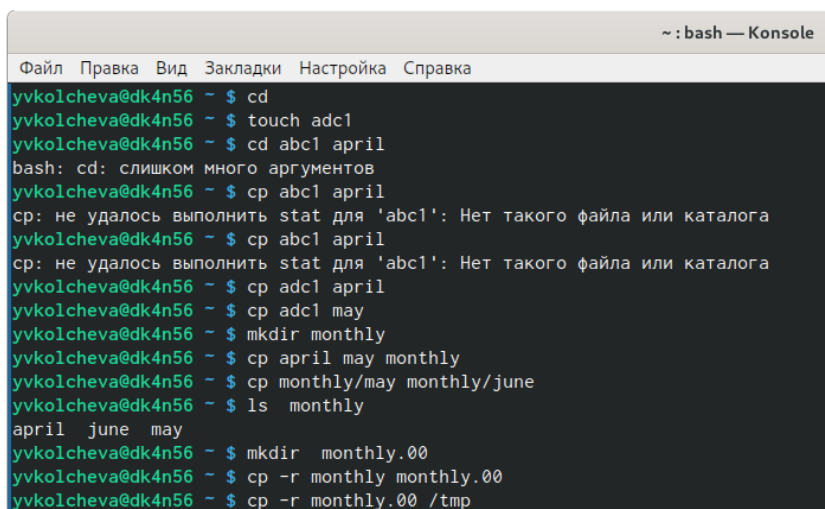
2.1	Первая часть	6
2.2	Перемещение и копирование	7
2.3	Права доступа	8
2.4	Скриншот консоли	9
2.5	Команды	9
2.6	Скриншот консоли	10
2.7	Скриншот консоли	11
2.8	Скриншот консоли	11
2.9	Скриншот консоли	12
2.10	Скриншот консоли	13
2.11	Скриншот консоли	14
2.12	Скриншот консоли	14

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 Выполнение лабораторной работы

Для начала выполним примеры из первой части лабораторной работы. 1) Скопируем файл “~/abc1” в файл april и в файл “may”. Для этого создадим файл “abc1” (touch abc1), далее осуществим копирование (ср abc1 april и ср abc1 may). (рис. 2.1) 2) Скопируем файлы “april” и “may” в каталог monthly (mkdir monthly – создаём каталог monthly и ср april may monthly – копируем). (рис. 2.1) 3) Скопируем файл monthly/may в файл с именем june. Выполним команды «срmonthly/maymonthly/june» и «lsmonthly». (рис. 2.1) 4) Скопируем каталог monthly в каталог monthly.00. Для этого создадим каталог monthly.00 (mkdirmonthly.00) и осуществим копирование, используя команду «ср -r monthly monthly.00». (рис. 2.1) 5) Скопируем каталог monthly.00 в каталог /tmp (ср -r monthly.00 /tmp). (рис. 2.1)



```
~ : bash — Konsole
Файл  Правка  Вид  Закладки  Настройка  Справка
yvkolcheva@dk4n56 ~ $ cd
yvkolcheva@dk4n56 ~ $ touch abc1
yvkolcheva@dk4n56 ~ $ cd abc1 april
bash: cd: слишком много аргументов
yvkolcheva@dk4n56 ~ $ cp abc1 april
cp: не удалось выполнить stat для 'abc1': Нет такого файла или каталога
yvkolcheva@dk4n56 ~ $ cp abc1 april
cp: не удалось выполнить stat для 'abc1': Нет такого файла или каталога
yvkolcheva@dk4n56 ~ $ cp abc1 may
yvkolcheva@dk4n56 ~ $ mkdir monthly
yvkolcheva@dk4n56 ~ $ cp april may monthly
yvkolcheva@dk4n56 ~ $ cp monthly/may monthly/june
yvkolcheva@dk4n56 ~ $ ls monthly
april  june  may
yvkolcheva@dk4n56 ~ $ mkdir monthly.00
yvkolcheva@dk4n56 ~ $ cp -r monthly monthly.00
yvkolcheva@dk4n56 ~ $ cp -r monthly.00 /tmp
```

Figure 2.1: Первая часть

- 1) Изменим название файла april на july в домашнем каталоге (mv april july). (рис. 2.2)
- 2) Переместим файл july в каталог monthly.00 (mv july monthly.00). Проверим результат командой «ls». (рис. 2.2)
- 3) Переименуем каталог monthly.00 в monthly.01 (mv monthly.00 monthly.01). (рис. 2.2)
- 4) Переместим каталог monthly.01 в каталог reports. Для этого создадим каталог reports помощью команды «mkdir reports» и выполним перемещение командой «mv monthly.01 reports». (рис. 2.2)
- 5) Переименуем каталог reports/monthly.01 в reports/monthly командой «mv reports/monthly.01 reports/monthly». (рис. 2.2)

```
yvkolcheva@dk4n56 ~ $ mv april july
yvkolcheva@dk4n56 ~ $ mv july monthly.00
yvkolcheva@dk4n56 ~ $ ls monthly.00
july  monthly
yvkolcheva@dk4n56 ~ $ mv monthly.00 monthly.01
yvkolcheva@dk4n56 ~ $ mkdir repots
yvkolcheva@dk4n56 ~ $ mv monthly.01 reports
yvkolcheva@dk4n56 ~ $ mv reports/monthly.01 reports/monthly
```

Figure 2.2: Перемещение и копирование

- 1) Создадим файл “~/may” с правом выполнения для владельца. Для этого выполним следующие команды: «touch may», «ls -l may», «chmod u+x may», «ls -l may». (рис. 2.3)
- 2) Лишаем владельца файла “~/may” права на выполнение, используя команды: «chmod u-x may», «ls -l may». (рис. 2.3)
- 3) Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly», «chmodgo -r monthly». (рис. 2.3)
- 4) Создаем файл “~/abc1” с правом записи для членов группы, используя команды: «touch abc1», «chmod g+w abc1». (рис. 2.3)

```

yvkolcheva@dk4n56 ~ $ cd
yvkolcheva@dk4n56 ~ $ touch may
yvkolcheva@dk4n56 ~ $ ls -l may
-rw-r--r-- 1 yvkolcheva studsci 0 мая 12 14:14 may
yvkolcheva@dk4n56 ~ $ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
yvkolcheva@dk4n56 ~ $ chmod u+x may
yvkolcheva@dk4n56 ~ $ ls -l may
-rwxr--r-- 1 yvkolcheva studsci 0 мая 12 14:14 may
yvkolcheva@dk4n56 ~ $ chmod u-x may
yvkolcheva@dk4n56 ~ $ ls -l may
-rw-r--r-- 1 yvkolcheva studsci 0 мая 12 14:14 may
yvkolcheva@dk4n56 ~ $ chmod go-r monthly
yvkolcheva@dk4n56 ~ $ cd
yvkolcheva@dk4n56 ~ $ touch abc1
yvkolcheva@dk4n56 ~ $ chmod g+w abc1

```

Figure 2.3: Права доступа

- 1) Копируем файл /usr/include/io.h и называем его equipment (mv io.h equipment). (рис. 2.4)
- 2) В домашнем каталоге создаем (mkdir ski.plases). (рис. 2.4)
- 3) Перемещаем файл equipment в каталог ski.plases (mvequipmentski.plases). (рис. 2.4)
- 4) Переименовываем файл (mv ski.plases/equipment ski.plases/equiplist). (рис. 2.4)
- 5) Создаем в домашнем каталоге файла abc1 и копируем его (cp abc1 ski.plases), называем его equiplist2 (mv ski.plases/abc1 ski.plases/equiplist2). (рис. 2.4)
- 6) Создаем каталог equipment в каталоге ski.plases. (рис. 2.4)
- 7) Перемещаем файлы (mv ski.plases/equiolist ski.plases/equiplist2 ski.plases/equipment). (рис. 2.4)
- 8) Создаем (команда «mkdirnewdir») и перемещаем каталог newdir в каталог ski.plases (mv newdir ski.plases) и называем его plans (mv ski.plases/newdir ski.plases/plans)(рис. 2.4)


```

yvkolcheva@dk4n56 ~ $ cp /usr/include/sys/io.h ~
yvkolcheva@dk4n56 ~ $ mv io.h equipment
yvkolcheva@dk4n56 ~ $ mkdir ski.plases
yvkolcheva@dk4n56 ~ $ mv equipment ski.plases
yvkolcheva@dk4n56 ~ $ mv ski.plases/equipmentski.plases/equipment
mv: после 'ski.plases/equipmentski.plases/equipment' пропущен операнд, задающий целевой файл
По команде «mv --help» можно получить дополнительную информацию.
yvkolcheva@dk4n56 ~ $ mv ski.plases/equipment ski.plases/equiplist
yvkolcheva@dk4n56 ~ $ touch abc1
yvkolcheva@dk4n56 ~ $ cp abc1 ski.plases
yvkolcheva@dk4n56 ~ $ mv ski.plases/abc1 ski.plases/equiplist2
yvkolcheva@dk4n56 ~ $ mkdir ski.plases/equipment
yvkolcheva@dk4n56 ~ $ mv ski.plases/equiplist ski.plases/equiplist2 ski.plases/equipment
yvkolcheva@dk4n56 ~ $ mv newdir ski.plases
mv: не удалось выполнить stat для 'newdir': Нет такого файла или каталога
yvkolcheva@dk4n56 ~ $ mkdir newdir
yvkolcheva@dk4n56 ~ $ mv newdir ski.plases
yvkolcheva@dk4n56 ~ $ mv ski.plases/newdir ski.plases/plans

```

Figure 2.4: Скриншот консоли

3) Определяем опции команды `chmod`, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа. Предварительно создаем необходимые файлы, используя команды: `mkdir australia`, `mkdir play`, `touch my_os`, `touch feathers` (рис. 2.5)

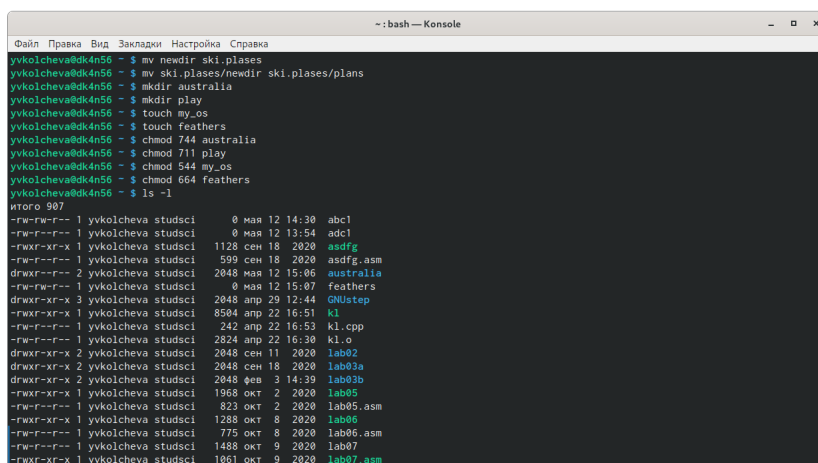
```

yvkolcheva@dk4n56 ~ $ mkdir australia
yvkolcheva@dk4n56 ~ $ mkdir play
yvkolcheva@dk4n56 ~ $ touch my_os
yvkolcheva@dk4n56 ~ $ touch feathers

```

Figure 2.5: Команды

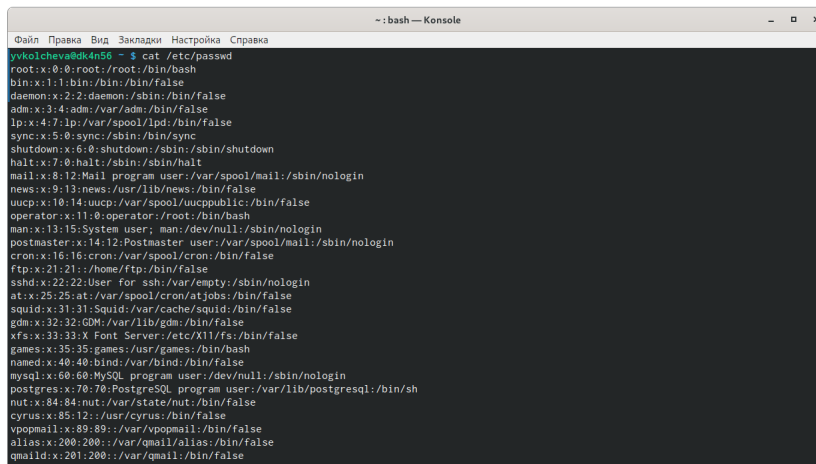
`drwxr-r-... australia`: команда `chmod 744 australia` (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение). (рис. 2.6) `drwx-x-x ... play`: команда `chmod 711 play` (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение). (рис. 2.6) `-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). (рис. 2.6) `-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. 2.6)



```
~: bash — Konsole
Файл Правка Вид Закладки Настройка Справка
yvkolcheva@dk4n56 ~$ mv newdir ski.plases
yvkolcheva@dk4n56 ~$ mv ski.plases/newdir ski.plases/plans
yvkolcheva@dk4n56 ~$ mkdir australia
yvkolcheva@dk4n56 ~$ mkdir play
yvkolcheva@dk4n56 ~$ touch my_os
yvkolcheva@dk4n56 ~$ touch feathers
yvkolcheva@dk4n56 ~$ chmod 744 australia
yvkolcheva@dk4n56 ~$ chmod 711 play
yvkolcheva@dk4n56 ~$ chmod 544 my_os
yvkolcheva@dk4n56 ~$ chmod 664 feathers
yvkolcheva@dk4n56 ~$ ls -l
итого 907
-rw-rw-r-- 1 yvkolcheva studsci 0 мая 12 14:30 abc1
-rw-rw-r-- 1 yvkolcheva studsci 0 мая 12 13:54 adc1
-rwxr-xr-x 1 yvkolcheva studsci 1128 сен 18 2020 asdfg
-rw-rw-r-- 1 yvkolcheva studsci 599 сен 18 2020 asdfg.asm
drwxr-xr-x 2 yvkolcheva studsci 2048 мая 12 15:06 australia
-rw-rw-r-- 1 yvkolcheva studsci 0 мая 12 15:07 feathers
drwxr-xr-x 3 yvkolcheva studsci 2048 апр 29 12:44 GNUstep
-rwxr-xr-x 1 yvkolcheva studsci 8504 апр 22 16:51 kl
-rw-rw-r-- 1 yvkolcheva studsci 242 апр 22 16:53 kl.cpp
-rw-rw-r-- 1 yvkolcheva studsci 2824 апр 22 16:30 kl.o
drwxr-xr-x 2 yvkolcheva studsci 2048 сен 11 2020 lab02
drwxr-xr-x 2 yvkolcheva studsci 2048 сен 18 2020 lab03a
drwxr-xr-x 2 yvkolcheva studsci 2048 фев 3 14:39 lab03b
-rwxr-xr-x 1 yvkolcheva studsci 1968 окт 2 2020 lab05
-rw-rw-r-- 1 yvkolcheva studsci 823 окт 2 2020 lab05.asm
-rwxr-xr-x 1 yvkolcheva studsci 1288 окт 8 2020 lab06
-rw-rw-r-- 1 yvkolcheva studsci 775 окт 8 2020 lab06.asm
-rw-rw-r-- 1 yvkolcheva studsci 1488 окт 9 2020 lab07
-rwxr-xr-x 1 yvkolcheva studsci 1061 окт 9 2020 lab07.asm
```

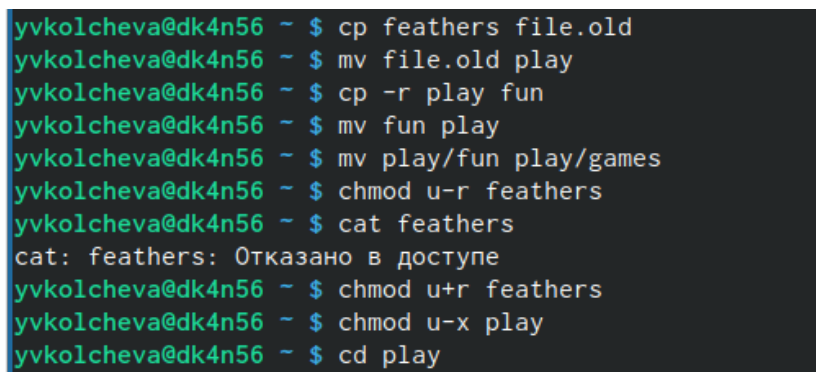
Figure 2.6: Скриншот консоли

Все следующие действия (рис. 2.7) и (рис. 2.8) 1) Владелец и группа владельца имеют право на чтение и запись, остальные – только чтение). Командой «ls-l» проверяем правильность выполненных действий. Выполняем следующие действия, отображенные. Просмотрим содержимое файла /etc/passwd (cat/etc/passwd). 2) Копируем файл feathers в файл file.old(команда «cpfeathersfile.old»). 3) Переместим файл file.old в каталог play(команда «mvfile.ordplay»). 4) Скопируем каталог play в каталог fun(команда «cp-rplayfun»). 5) Переместим каталог fun в каталог play(команда «mvfunplay»)и назовем его games(команда «mvplay/funplay/games»). 6) Лишим владельца файла feathers права на чтение(команда «chmod u-r feathers»). 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+r feathers»). 10) Лишим владельца каталога play права на выполнение(команда «chmod u-x play»). 11) Перейдем в каталог play(команда «cdplay»). Получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на выполнение данного каталога. 12) Дадим владельцу каталога play право на выполнение(команда «chmod u+xp lay»).

A terminal window titled '~: bash — Konsole' displays the output of the command 'cat /etc/passwd'. The output lists system and user accounts with their IDs, names, shells, and home directories. The window has a menu bar with 'Файл', 'Правка', 'Вид', 'Закладки', 'Настройка', and 'Справка'.

```
yvkolcheva@dk4n56 ~ $ 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
```

Figure 2.7: Скриншот консоли

A terminal window shows a series of commands being executed to create a directory structure and manage file permissions. The user is yvkolcheva@dk4n56. The commands include copying a file, moving it, creating a directory, and setting permissions.

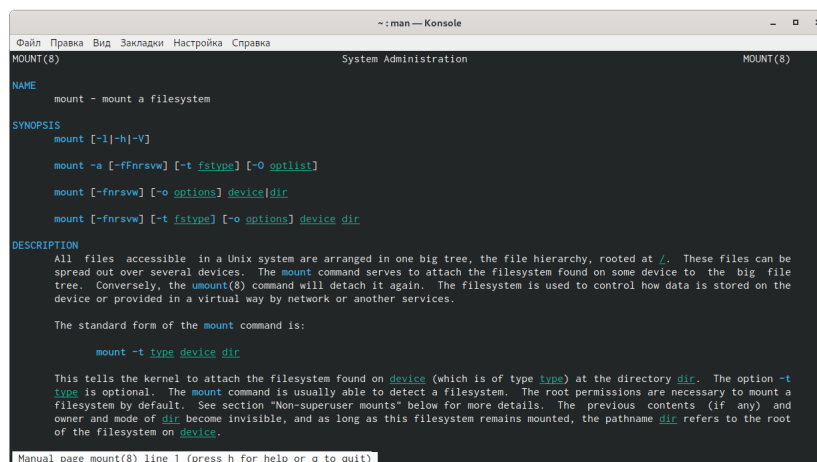
```
yvkolcheva@dk4n56 ~ $ cp feathers file.old
yvkolcheva@dk4n56 ~ $ mv file.old play
yvkolcheva@dk4n56 ~ $ cp -r play fun
yvkolcheva@dk4n56 ~ $ mv fun play
yvkolcheva@dk4n56 ~ $ mv play/fun play/games
yvkolcheva@dk4n56 ~ $ chmod u-r feathers
yvkolcheva@dk4n56 ~ $ cat feathers
cat: feathers: Отказано в доступе
yvkolcheva@dk4n56 ~ $ chmod u+r feathers
yvkolcheva@dk4n56 ~ $ chmod u-x play
yvkolcheva@dk4n56 ~ $ cd play
```

Figure 2.8: Скриншот консоли

Используя команды «man mount», «man fsck», «man mkfs», «man kill», получим информацию о соответствующих командах.

Команда mount предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки и листья. 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 -tvfstype 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. 2.9)



```
~: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:

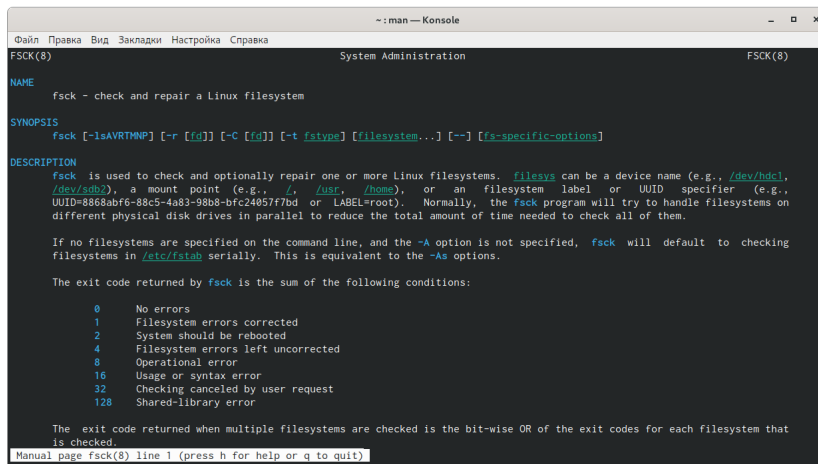
        mount -t type device dir

    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.

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

Figure 2.9: Скриншот консоли

(рис. 2.10): Команда `fsck`: (рис. 2.10) 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.



```
~: 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.

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

Figure 2.10: Скриншот консоли

Команда `mkfs` Создаёт новую файловую систему Linux. Имеет следующий синтаксис: `mkfs[-V] [-tfstype] [fs-options] filesystems[blocks]` `mkfs` используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента `filesystems` для файловой системы может выступать или название устройства (например, `/dev/hda1`, `/dev/sdb2`) или точка монтирования (например, `/`, `/usr`, `/home`). Аргументом `blocks` указывается количество 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. 2.11)

Figure 2.11: Скриншот консоли

Команда kill (рис. 2.12) 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.

Figure 2.12: Скриншот консоли

3 Выводы

В ходе выполнения 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, по проверке использования диска и обслуживанию файловой системы.

4 Контрольные вопросы

Ответы на контрольные вопросы: Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df-Th». Из рисунка видно, что на моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. 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 предназначена 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 запи-

си (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 TB; максимальный размер раздела: 16 TB; 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. ISO 9660 – стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows. 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 (Рисунок 15) 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 (Рисунок 16). (Рисунок 16) / – корень Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять фай-

лы в этом каталоге BIN –бинарные файлы пользователяЭтот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления. SBIN –системные исполняемые файлыТак 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 файлыВ 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 устройствВ 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. ROC –информация о процессахПо 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 кешы, базы данных и так далее. TMP–временные файлыВ этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию.USR –программы пользователяЭто самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию.HOME –домашняя папкаВ этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т.д. BOOT –файлы загрузки

ка Содержит все файлы, связанные с загрузчиком системы. Это ядро `vmlinuz`, образ `initrd`, а также файлы загрузчика, находящиеся 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** – съемные 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 и 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, не добавляя 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 файле⁷⁾ Утилита `rsync` позволяет полностью копировать файлы и директории. Синтаксис: `rsync [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-r –сохранять владельца, временные метки и флаги доступа при копировании-t –считать файл-приемник директорией и копировать файл-источник в эту директорию8)Командаmvиспользуется для перемещения одного или нескольких файлов (или директорий) вдругую директорию, а также для переименования файлов и директорий.Синтаксис:mv [-опции] старый_файл новый_файлОсновные 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 командыКоманда renатакже предназначена, чтобы переименовать файл.Синтаксис:rename [опции] старое_имя новое_имя файлыОсновные опции:-v –вывести список обработанных файлов-n –тестовый режим, на самом деле никакие действия выполнены не будут-f –принудительно перезаписывать существующие файлы9)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.Синтаксискоманды:chmod режим имя_файлаРежим имеет следующие компоненты структуры и способзаписи:= установить право-лишить

права+ даты+правов чтение+записей+выполнение (user) владелец файла (group)
группа, к которой принадлежит владелец файла (others) все остальные.