

Отчёт по лабораторной работе 5

Дисциплина: Операционные системы

Волчок Кристина Александровна

Содержание

1	Цель работы	4
2	Последовательность выполнения работы	5
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	17
5	Выводы	25

Список иллюстраций

3.1	Создание текстового файла и копирование	7
3.2	Копирование каталогов в текущем каталоге. Копируем каталог monthly в другой каталог	7
3.3	Переименование файлов в текущем каталоге и перемещение . . .	7
3.4	Копируем файл /usr/include/sys/io.h в домашний каталог и называ- ем его equipment	8
3.5	В домашнем каталоге создаем директорию ~/ski.plases	8
3.6	Перемещаем файл equipment в каталог ~/ski.plases	8
3.7	Переименование	8
3.8	Создание и копирование	9
3.9	Создание и перемещение	9
3.10	Создание и перемещение	9
3.11	Опции команды chmod	10
3.12	содержимое файла /etc/password	10
3.13	Копирование	11
3.14	Перемещение	11
3.15	Копирование	11
3.16	Перемещение и переименование	11
3.17	Лишение и получение прав владельцаам	12
3.18	mount	13
3.19	fsck	14
3.20	kill	15
3.21	mkfs	16
4.1	19
4.2	Формы записи прав доступа	24

1 Цель работы

В ходе выполнения лабораторной работы я должна ознакомиться с файловой системой Linux, её структурой, именами и содержанием каталогов. А также 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 Последовательность выполнения работы

1. Выполняем все примеры, приведённые в первой части описания лабораторной работы.
2. Выполняем следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
 - 2.1. Копируем файл `/usr/include/sys/io.h` в домашний каталог и называем его `equipment`. Если файла `io.h` нет, то используем любой другой файл в каталоге `/usr/include/sys/` вместо него.
 - 2.2. В домашнем каталоге создаем директорию `~/ski.plases`.
 - 2.3. Перемещаем файл `equipment` в каталог `~/ski.plases`.
 - 2.4. Переименоуем файл `~/ski.plases/equipment` в `~/ski.plases/equiplist`.
 - 2.5. Создаем в домашнем каталоге файл `abc1` и копируем его в каталог `~/ski.plases`, называем его `equiplist2`.
 - 2.6. Создаем каталог с именем `equipment` в каталоге `~/ski.plases`.
 - 2.7. Перемещаем файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment`.
 - 2.8. Создаем и перемещаем каталог `~/newdir` в каталог `~/ski.plases` и называем его `plans`.
3. Определяем опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет. При необходимости создаем нужные файлы.
4. Проведем приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 - 4.1. Просмотрим содержимое

- файла `/etc/passwd`. 4.2. Скопируем файл `~/feathers` в файл `~/file.old`. 4.3. Переместим файл `~/file.old` в каталог `~/play`. 4.4. Скопируем каталог `~/play` в каталог `~/fun`. 4.5. Переместим каталог `~/fun` в каталог `~/play` и назовите его `games`. 4.6. Лишим владельца файла `~/feathers` права на чтение. 4.7. Что произойдёт, если мы попытаемся просмотреть файл `~/feathers` командой `cat`? 4.8. Что произойдёт, если мы попытаемся скопировать файл `~/feathers`? 4.9. Даем владельцу файла `~/feathers` право на чтение. 4.10. Лишите владельца каталога `~/play` права на выполнение. 4.11. Переходим в каталог `~/play`. Что произошло? 4.12. Даем владельцу каталога `~/play` право на выполнение.
5. Прочитаем ман по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуем.

3 Выполнение лабораторной работы

1. Выполняем все примеры, приведённые в первой части описания лабораторной работы. Описываются проведённые действия, в качестве иллюстрации даётся ссылка на иллюстрацию. Рисунки 3.1; 3.2; 3.3.

```
kavolchok@dk6n51 ~ $ cd
kavolchok@dk6n51 ~ $ touch abc1
kavolchok@dk6n51 ~ $ cp abc1 april
kavolchok@dk6n51 ~ $ cp abc1 may
kavolchok@dk6n51 ~ $ mkdir monthly
kavolchok@dk6n51 ~ $ cp april may monthly
kavolchok@dk6n51 ~ $ cp monthly/may monthly/june
kavolchok@dk6n51 ~ $ ls monthly
april  june  may
```

Рис. 3.1: Создание текстового файла и копирование

```
kavolchok@dk6n51 ~ $ mkdir monthly.00
kavolchok@dk6n51 ~ $ cp -r monthly monthly.00
kavolchok@dk6n51 ~ $ cp -r monthly.00 /tmp
```

Рис. 3.2: Копирование каталогов в текущем каталоге. Копируем каталог monthly в другой каталог

```
kavolchok@dk6n51 ~ $ cd
kavolchok@dk6n51 ~ $ mv april july
kavolchok@dk6n51 ~ $ mv july monthly.00
kavolchok@dk6n51 ~ $ ls monthly.00
july  monthly
kavolchok@dk6n51 ~ $ mv monthly.00 monthly.01
kavolchok@dk6n51 ~ $ mkdir reports
kavolchok@dk6n51 ~ $ mv monthly.01 reports
kavolchok@dk6n51 ~ $ mv reports/monthly.01 reports/monthly
```

Рис. 3.3: Переименование файлов в текущем каталоге и перемещение

2. Копируем файл /usr/include/sys/io.h в домашний каталог и называем его equipment. Если файла io.h нет, то используем любой другой файл в каталоге /usr/include/sys/ вместо него.(рис.3.4).

```
kavolchok@dk6n51 ~ $ cd
kavolchok@dk6n51 ~ $ cp /usr/include/sys/io.h -
kavolchok@dk6n51 ~ $ cp /usr/include/sys/io.h ~equipment
kavolchok@dk6n51 ~ $ ls
-          may      public      tmp      Изображения  Шаблоны
abc1       monthly    public_html Видео      Музыка
Architecture_PC newdir    r          Документы  Общедоступные
'equipment' os-intro  reports    Загрузки   'Рабочий стол'
kavolchok@dk6n51 ~ $ mv io.h equipment
mv: не удалось выполнить stat для 'io.h': Нет такого файла или каталога
kavolchok@dk6n51 ~ $ cp /usr/include/sys/io.h equipment
kavolchok@dk6n51 ~ $ ls
-          may      public_html Документы   'Рабочий стол'
abc1       monthly    r          Загрузки   Шаблоны
Architecture_PC newdir    reports    Изображения
'equipment' os-intro  tmp        Музыка
equipment   public    Видео      Общедоступные
```

Рис. 3.4: Копируем файл /usr/include/sys/io.h в домашний каталог и называем его equipment

В домашнем каталоге создаем директорию ~/ski.plases (рис.3.5).

```
kavolchok@dk6n51 ~ $ mkdir ski.plases
kavolchok@dk6n51 ~ $ mv equipment ski.plases
```

Рис. 3.5: В домашнем каталоге создаем директорию ~/ski.plases

Перемещаем файл equipment в каталог ~/ski.plases (рис.3.6).

```
kavolchok@dk6n51 ~ $ mkdir ski.plases
kavolchok@dk6n51 ~ $ mv equipment ski.plases
```

Рис. 3.6: Перемещаем файл equipment в каталог ~/ski.plases

Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist(рис.3.7).

```
kavolchok@dk6n51 ~ $ mv ski.plases/equipment ski.plases/equiplis
kavolchok@dk6n51 ~ $ touch abc1
```

Рис. 3.7: Переименование

Создаем в домашнем каталоге файл abc1 и копируем его в каталог ~/ski.plases, называем его equiplist2.(рис.3.8).


```
kavolchok@dk6n51 ~ $ touch abc1
kavolchok@dk6n51 ~ $ cp abc1 ski.places
kavolchok@dk6n51 ~ $ mv ski.places/abc1 ski.places/equiplist2
```

Рис. 3.8: Создание и копирование

Создаем каталог с именем equipment в каталоге ~/ski.places и перемещаем его(рис.3.9).

```
kavolchok@dk6n51 ~ $ mkdir ski.places/equipment
kavolchok@dk6n51 ~ $ mv ski.places/equiplist ski.places/equipment
```

Рис. 3.9: Создание и перемещение

Создаем и перемещаем каталог ~/newdir в каталог ~/ski.places и называем его plans(рис.3.10).

```
kavolchok@dk6n51 ~ $ mv newdir plans
kavolchok@dk6n51 ~ $ mv plans ski.places
kavolchok@dk6n51 ~ $ ls
-          monthly      reports      Документы      'Рабочий стол'
abc1       os-intro      ski.        Загрузки       Шаблоны
Architecture_PC public      ski.places  Изображения
equipment  public_html tmp         Музыка
may        r           Видео      Общедоступные
kavolchok@dk6n51 ~ $ ls ski.places
equipment plans
```

Рис. 3.10: Создание и перемещение

3. Определяем опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет.(рис.3.11)

```

kavolchok@dk6n51 ~ $ mkdir australia
kavolchok@dk6n51 ~ $ mkdir play
kavolchok@dk6n51 ~ $ touch my_os
kavolchok@dk6n51 ~ $ touch feathers
kavolchok@dk6n51 ~ $ chmod 744 australia
kavolchok@dk6n51 ~ $ chmod 711 play
kavolchok@dk6n51 ~ $ chmod 544 my_os
kavolchok@dk6n51 ~ $ chmod 664 feathers
kavolchok@dk6n51 ~ $ ls -l
итого 51
-rw-r--r-- 1 kavolchok studsci 5086 апр 27 15:18 -
-rw-r--r-- 1 kavolchok studsci 0 апр 27 15:32 abc1
drwxr-xr-x 11 kavolchok studsci 2048 окт 21 2021 Architecture_PC
drwxr--r-- 2 kavolchok studsci 2048 апр 27 16:09 australia
-rw-r--r-- 1 kavolchok studsci 5086 апр 27 15:47 equipment
-rw-rw-r-- 1 kavolchok studsci 0 апр 27 16:10 feathers
-rw-r--r-- 1 kavolchok studsci 0 апр 27 14:22 may
drwx--x--x 2 kavolchok studsci 2048 апр 27 14:38 monthly
-r-xr--r-- 1 kavolchok studsci 0 апр 27 16:10 my_os
drwxr-xr-x 7 kavolchok studsci 2048 апр 26 13:07 os-intro
drwx--x--x 2 kavolchok studsci 2048 апр 27 16:10 play
drwxr-xr-x 3 kavolchok studsci 2048 сен 1 2021 public
lrwxr-xr-x 1 kavolchok root 18 апр 14 21:35 public_html -> public/public_html
drwxr-xr-x 7 kavolchok studsci 2048 апр 21 18:08 r
drwxr-xr-x 3 kavolchok studsci 2048 апр 27 14:16 reports
-rw-r--r-- 1 kavolchok studsci 0 апр 27 15:28 ski.
drwxr-xr-x 4 kavolchok studsci 2048 апр 27 15:56 ski_places
drwxr-xr-x 2 kavolchok studsci 2048 мар 24 16:38 tmp
drwxr-xr-x 2 kavolchok studsci 2048 сен 2 2021 Видео
drwxr-xr-x 3 kavolchok studsci 2048 апр 26 13:02 Документы
drwxr-xr-x 2 kavolchok studsci 4096 апр 26 20:04 Загрузки
drwxr-xr-x 2 kavolchok studsci 4096 апр 27 13:50 Изображения
drwxr-xr-x 2 kavolchok studsci 2048 сен 2 2021 Музыка
drwxr-xr-x 2 kavolchok studsci 2048 сен 2 2021 Общедоступные

```

Рис. 3.11: Опции команды chmod

4. Просмотрим содержимое файла /etc/passwd.(рис.3.12)

```

kavolchok@dk6n51 ~ $ 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: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:User for running GDM:/var/lib/gdm:/sbin/nologin

```

Рис. 3.12: содержимое файла /etc/passwd

Скопируем файл ~/feathers в файл ~/file.old.(рис.3.13)

```
FileBIRD.X:430.430: added by portage for file  
kavolchok@dk6n51 ~ $ cp feathers file.old
```

Рис. 3.13: Копирование

Переместим файл ~/file.old в каталог ~/play.(рис.3.14)

```
kavolchok@dk6n51 ~ $ cp feathers file.old  
kavolchok@dk6n51 ~ $ mv file.old play
```

Рис. 3.14: Перемещение

Скопируем каталог ~/play в каталог ~/fun.(рис.3.15)

```
kavolchok@dk6n51 ~ $ cp -r play fun
```

Рис. 3.15: Копирование

Переместим каталог ~/fun в каталог ~/play и называем его games.(рис.3.16)

```
equipment monthly public_html tmp музыка  
kavolchok@dk6n51 ~ $ mv fun play  
kavolchok@dk6n51 ~ $ mv play/fun play/games
```

Рис. 3.16: Перемещение и переименование

1)Лишим владельца файла ~/feathers права на чтение (команда «chmod u-r feathers»)..(рис.3.17) 2) 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..(рис.3.17) 3) Если мы попытаемся скопировать файл ~/feathers, например, в каталог monthly, то получим отказ в доступе, по причине, описанной в предыдущем пункте.(рис.3.17) 4)Дадим владельцу файла ~/feathers право на чтение (команда «chmod u+r feathers»)(рис.3.17) 5) Лишим владельца каталога ~/play права на выполнение (команда «chmod u-x play»).(рис.3.17) 6) Перейдем в каталог ~/play (команда «cd play»). Получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на выполнение данного каталога..(рис.3.17) 7) Дадим владельцу каталога ~/play право на выполнение (команда «chmod u+x play»).(рис.3.17)

```

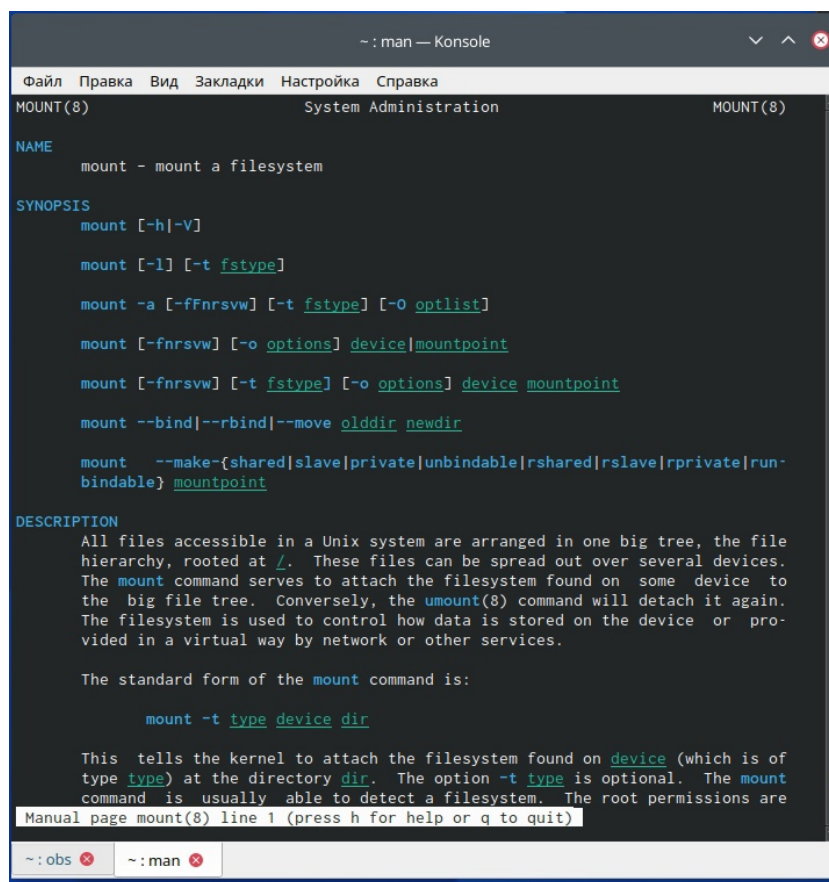
kavolchok@dk6n51 ~ $ chmod u-r feathers
kavolchok@dk6n51 ~ $ chmod u+r feathers
kavolchok@dk6n51 ~ $ chmod u-r play
kavolchok@dk6n51 ~ $ cd play
kavolchok@dk6n51 ~/play $ cd
kavolchok@dk6n51 ~ $ chmod u-x play
kavolchok@dk6n51 ~ $ cd play
kavolchok@dk6n51 ~/play $ cd
kavolchok@dk6n51 ~ $ chmod u*x play
chmod: неверный режим: «u*x»
По команде «chmod --help» можно получить дополнительную информацию.
kavolchok@dk6n51 ~ $ chmod u-x play
kavolchok@dk6n51 ~ $ chmod u+x play

```

Рис. 3.17: Лишение и получение прав владельца

5. Прочитаем man по командам mount, fsck, mkfs, kill. Рисунки 3.18;3.19;3.20;3.21

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.3.18)



```
~: man — Konsole
Файл Правка Вид Закладки Настройка Справка
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|rslave|rprivate|run-
bindable} mountpoint

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 pro-
vided 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

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

~: obs ~: man
```

Рис. 3.18: mount

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.3.19)

```
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 seri-
    ally. 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.

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

Рис. 3.19: fsck

Команда kill: посылает сигнал процессу или выводит список допустимых сиг-
налов. 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» по-
сылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить про-
цесс.(рис.3.20)


```
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 sig-
    nals. 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 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.

    -q, --queue value
        Use sigqueue(3) rather than kill(2) and the value argument is used to
        specify an integer to be sent with the signal. If the receiving
        process has installed a handler for this signal using the SA_SIGINFO
        flag to sigaction(2), then it can obtain this data via the si_value
        field of the siginfo_t structure.

    -l, --list [signal]
        List signal names. This option has optional argument, which will

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

Рис. 3.20: kill

Команда `mkfs`: создаёт новую файловую систему Linux. Имеет следующий синтаксис: `mkfs [-V] [-t fstype] [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 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.3.21)

```

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 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.

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

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

```

Рис. 3.21: mkfs

4 Контрольные вопросы

- 1) 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 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 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 CDRом. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

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.(рис.4.1)

```

/-
|-/etc-|-/etc/X11-|-/etc/X11/xinit.d
|      |      |      |...
|      |      |      |-files
|      |      |      |...
|-/opt
|-/home <- |-/user1-|-/user1/Desktop # примонтированный раздел ext3,
|          |          |-/user1/Documents # содержащий свое дерево каталогов
|          |          |... # (/home - точка монтирования)
|          |          |-/user2
|
|.....
|-/usr
|-/var

```

Рис. 4.1: .

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

дить 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) Команда `mv` используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий. Синтаксис: `mv [-опции] старый_файл новый_файл` Основные опции: -help – выводит на экран официальную документацию об утилите -version – отображает версию `mv` -b – создает копию файлов, которые были перемещены или перезаписаны -f – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла -i – наоборот, будет спрашивать разрешение у владельца -n – отключает перезапись уже существующих объектов -strip-trailing-slashes – удаляет завершающий символ / у файла при его наличии -t [директория] — перемещает все

файлы в указанную директорию -и – 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 r чтение w запись x выполнение u (user) владелец файла g (group) группа, к которой принадлежит владелец файла o (others) все остальные(рис.4.2)

Формы записи прав доступа

Двоичная	Восьмеричная	Символьная
111	7	rwx
110	6	rw-
101	5	r-x
100	4	r--
011	3	-wx
010	2	-w-
001	1	--x
000	0	---

Рис. 4.2: Формы записи прав доступа

5 Выводы

В ходе выполнения лабораторной работы я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов. А также 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.