

Отчёт по лабораторной работе номер 6

Операционные системы

Нитусова Диана Денисовна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Контрольные вопросы	16
5	Выводы	23

List of Tables

List of Figures

3.1	Рисунок 1	8
3.2	Рисунок 2	8
3.3	Рисунок 3	9
3.4	Рисунок 4	10
3.5	Рисунок 5	10
3.6	Рисунок 6	11
3.7	Рисунок 7	12
3.8	Рисунок 8	12
3.9	Рисунок 9	13
3.10	Рисунок 10	14
3.11	Рисунок 11	15
3.12	Рисунок 12	15

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 Задание

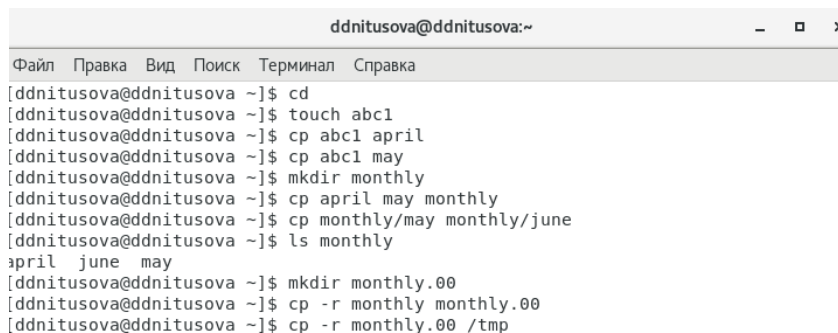
Ознакомиться с файловой системой.

3 Выполнение лабораторной работы

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

(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»
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»
5. Скопируем каталог monthly.00 в каталог /tmp, используя команду «cp -r monthly.00 /tmp».



```
ddnitusova@ddnitusova:~  
Файл Правка Вид Поиск Терминал Справка  
[ddnitusova@ddnitusova ~]$ cd  
[ddnitusova@ddnitusova ~]$ touch abc1  
[ddnitusova@ddnitusova ~]$ cp abc1 april  
[ddnitusova@ddnitusova ~]$ cp abc1 may  
[ddnitusova@ddnitusova ~]$ mkdir monthly  
[ddnitusova@ddnitusova ~]$ cp april may monthly  
[ddnitusova@ddnitusova ~]$ cp monthly/may monthly/june  
[ddnitusova@ddnitusova ~]$ ls monthly  
april june may  
[ddnitusova@ddnitusova ~]$ mkdir monthly.00  
[ddnitusova@ddnitusova ~]$ cp -r monthly monthly.00  
[ddnitusova@ddnitusova ~]$ cp -r monthly.00 /tmp
```

Figure 3.1: Рисунок 1

(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`».

```
[ddnitusova@ddnitusova ~]$ cd  
[ddnitusova@ddnitusova ~]$ mv april july  
[ddnitusova@ddnitusova ~]$ mv july monthly.00  
[ddnitusova@ddnitusova ~]$ ls monthly.00  
july monthly  
[ddnitusova@ddnitusova ~]$ mv monthly.00 monthly.01  
[ddnitusova@ddnitusova ~]$ mkdir reports  
mkdir: невозможно создать каталог «reports»: Файл существует  
[ddnitusova@ddnitusova ~]$ mv monthly.01 reports  
[ddnitusova@ddnitusova ~]$ mv reports/monthly.01 reports/monthly  
[ddnitusova@ddnitusova ~]$ █
```

Figure 3.2: Рисунок 2

(рис. 3.3)

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».
2. Лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may», «ls -l may».
3. Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly», «chmod go-r monthly».
4. Создаем файл ~/abc1 с правом записи для членов группы, используя команды: «touch abc1», «chmod g+w abc1».

```
[ddnitusova@ddnitusova ~]$ touch may
[ddnitusova@ddnitusova ~]$ ls -l may
-rw-rw-r--. 1 ddnitusova ddnitusova 0 май 14 21:55 may
[ddnitusova@ddnitusova ~]$ chmod u+x may
[ddnitusova@ddnitusova ~]$ ls -l may
-rwxrw-r--. 1 ddnitusova ddnitusova 0 май 14 21:55 may
[ddnitusova@ddnitusova ~]$ chmod u-x may
[ddnitusova@ddnitusova ~]$ ls -l may
-rw-rw-r--. 1 ddnitusova ddnitusova 0 май 14 21:55 may
[ddnitusova@ddnitusova ~]$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
[ddnitusova@ddnitusova ~]$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
То команде «chmod --help» можно получить дополнительную информацию.
[ddnitusova@ddnitusova ~]$ chmod go-r monthly
[ddnitusova@ddnitusova ~]$ touch abc1
[ddnitusova@ddnitusova ~]$ chmod g+w abc1
[ddnitusova@ddnitusova ~]$
```

Figure 3.3: Рисунок 3

Выполняем следующие действия: (рис. 3.4)

1. Копируем файл /usr/include/aio.h (т.к. у меня нет каталога /usr/include/sys/, то беру произвольный файл из каталога /usr/include/) в домашний каталог (команда «cp /usr/include/aio.h ~») и называем его equipment (команда «mv aio.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.plases/equipment` в `~/ski.plases/equiplist` (команда «`mv ski.plases/equipment ski.plases/equiplist`»).
5. Создаем в домашнем каталоге файл `abc1` (команда «`touch abc1`») и копируем его в каталог `~/ski.plases` (команда «`cp abc1 ski.plases`»), называем его `equiplist2` (команда «`mv ski.plases/abc1 ski.plases/equiplist2`»).
6. Создаем каталог с именем `equipment` в каталоге `~/ski.plases` (команда «`mkdir ski.plases/equipment`»).
7. Перемещаем файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment` (команда «`mv ski.plases/equiolist ski.plases/equiplist2 ski.plases/equipment`»).
8. Создаем (команда «`mkdir newdir`») и перемещаем каталог `~/newdir` в каталог `~/ski.plases` (команда «`mv newdir ski.plases`») и называем его `plans` (команда «`mv ski.plases/newdir ski.plases/plans`»).

```
[ddnitusova@ddnitusova ~]$ cp /usr/include/aio.h ~
[ddnitusova@ddnitusova ~]$ mv aio.h equipment
[ddnitusova@ddnitusova ~]$ mkdir ski.plases
[ddnitusova@ddnitusova ~]$ mv equipment ski.plases
[ddnitusova@ddnitusova ~]$ mv ski.plases/equipment ski.plases/equiplist
[ddnitusova@ddnitusova ~]$ touch abc1
[ddnitusova@ddnitusova ~]$ cp abc1 ski.plases
[ddnitusova@ddnitusova ~]$ mv ski.plases/abc1 ski.plases/equiplist2
[ddnitusova@ddnitusova ~]$ mv ski.plases/equiplist ski.plases/equiplist2 ski.plases/equipment
[ddnitusova@ddnitusova ~]$ mkdir newdir
[ddnitusova@ddnitusova ~]$ mv newdir ski.plases
mv: не удалось выполнить stat для «newdir»: Нет такого файла или каталога
[ddnitusova@ddnitusova ~]$ mkdir newdir
[ddnitusova@ddnitusova ~]$ mv newdir ski.plases
[ddnitusova@ddnitusova ~]$ mv ski.plases/newdir ski.plases/plans
[ddnitusova@ddnitusova ~]$
```

Figure 3.4: Рисунок 4

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)

```
[ddnitusova@ddnitusova ~]$ mkdir australia
[ddnitusova@ddnitusova ~]$ mkdir play
[ddnitusova@ddnitusova ~]$ touch my_os
[ddnitusova@ddnitusova ~]$ touch feathers
[ddnitusova@ddnitusova ~]$
```

Figure 3.5: Рисунок 5

`drwxr-r- ... australia`: команда «`chmod 744 australia`» (это каталог, владелец имеет

право на чтение, запись и выполнение, группа владельца и остальные – только чтение).

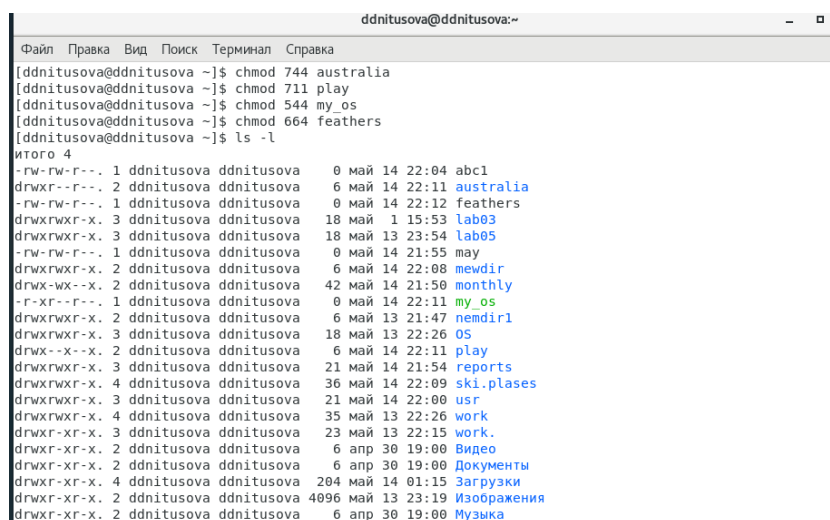
drwx-x-x ... play: команда «chmod 711 play» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение).

-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).

-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)



```
ddnitusova@ddnitusova:~$ chmod 744 australia
ddnitusova@ddnitusova:~$ chmod 711 play
ddnitusova@ddnitusova:~$ chmod 544 my_os
ddnitusova@ddnitusova:~$ chmod 664 feathers
ddnitusova@ddnitusova:~$ ls -l
итого 4
-rw-rw-r-- 1 ddnitusova ddnitusova 0 май 14 22:04 abc1
drwxr--r-- 2 ddnitusova ddnitusova 6 май 14 22:11 australia
-rw-rw-r-- 1 ddnitusova ddnitusova 0 май 14 22:12 feathers
drwxrwxr-x 3 ddnitusova ddnitusova 18 май 1 15:53 lab03
drwxrwxr-x 3 ddnitusova ddnitusova 18 май 13 23:54 lab05
-rw-rw-r-- 1 ddnitusova ddnitusova 0 май 14 21:55 may
drwxrwxr-x 2 ddnitusova ddnitusova 6 май 14 22:08 mewdir
drwx-wx--x 2 ddnitusova ddnitusova 42 май 14 21:50 monthly
-r-xr--r-- 1 ddnitusova ddnitusova 0 май 14 22:11 my_os
drwxrwxr-x 2 ddnitusova ddnitusova 6 май 13 21:47 nemdir1
drwxrwxr-x 3 ddnitusova ddnitusova 18 май 13 22:26 OS
drwx--x--x 2 ddnitusova ddnitusova 6 май 14 22:11 play
drwxrwxr-x 3 ddnitusova ddnitusova 21 май 14 21:54 reports
drwxrwxr-x 4 ddnitusova ddnitusova 36 май 14 22:09 ski.places
drwxrwxr-x 3 ddnitusova ddnitusova 21 май 14 22:00 usr
drwxrwxr-x 4 ddnitusova ddnitusova 35 май 13 22:26 work
drwxr-xr-x 3 ddnitusova ddnitusova 23 май 13 22:15 work.
drwxr-xr-x 2 ddnitusova ddnitusova 6 апр 30 19:00 Видео
drwxr-xr-x 2 ddnitusova ddnitusova 6 апр 30 19:00 Документы
drwxr-xr-x 4 ddnitusova ddnitusova 204 май 14 01:15 Загрузки
drwxr-xr-x 2 ddnitusova ddnitusova 4096 май 13 23:19 Изображения
drwxr-xr-x 2 ddnitusova ddnitusova 6 апр 30 19:00 Музыка
```

Figure 3.6: Рисунок 6

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.ord 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-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 «cd play»). 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+x play»).

```
ddnitusova@ddnitusova ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
bus:x:81:81:system message bus:/:/sbin/nologin
polkitd:x:999:999:user for polkitd:/:/sbin/nologin
libstoragemgmt:x:998:998:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
colord:x:997:995:user for colord:/var/lib/colord:/sbin/nologin
rpcbind:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
sane:x:996:994:SANE scanner daemon user:/usr/share/sane:/sbin/nologin
saslauthd:x:995:76:saslauthd user:/run/saslauthd:/sbin/nologin
abrt:x:173:173:/:/etc/abrt:/sbin/nologin
setroubleshoot:x:994:991:/:/var/lib/setroubleshoot:/sbin/nologin
realtimekit:x:172:172:RealtimeKit:/proc:/sbin/nologin
pulseaudio:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
radvd:x:75:75:radvd user:/:/sbin/nologin
chrony:x:993:988:/:/var/lib/chrony:/sbin/nologin
unbound:x:992:987:Unbound DNS resolver:/etc/unbound:/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
geoclue:x:991:985:user for geoclue:/var/lib/geoclue:/sbin/nologin
glusterfs:x:990:984:GlusterFS daemons:/run/gluster:/sbin/nologin
gdm:x:42:42:/:/var/lib/gdm:/sbin/nologin
pcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
gnome-initial-setup:x:989:983:/:/run/gnome-initial-setup:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
postfix:x:89:89:/:/var/spool/postfix:/sbin/nologin
ntp:x:38:38:/:/etc/ntp:/sbin/nologin
cpdump:x:72:72:/:/sbin/nologin
ddnitusova:x:1000:1000:ddnitusova:/home/ddnitusova:/bin/bash
vboxadd:x:988:1:/:/var/run/vboxadd:/bin/false
```

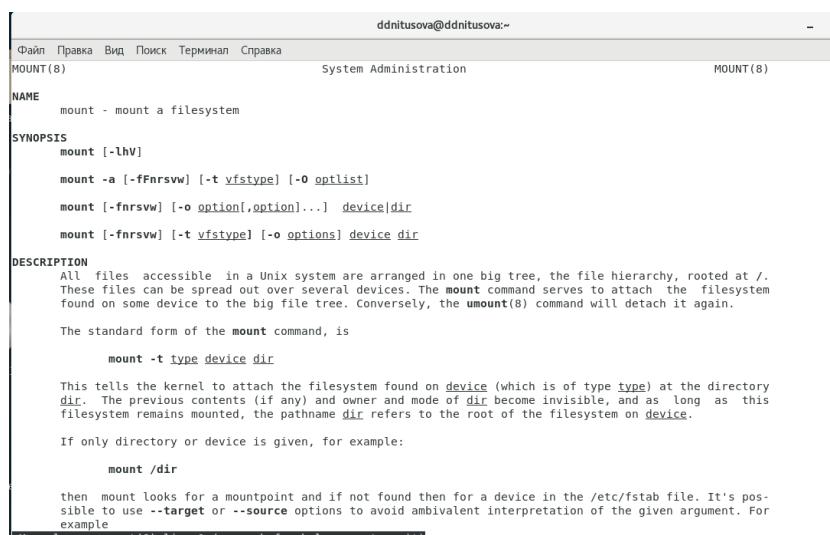
Figure 3.7: Рисунок 7

```
ddnitusova@ddnitusova ~]$ cp feathers file.old
ddnitusova@ddnitusova ~]$ mv file.old play
ddnitusova@ddnitusova ~]$ cp -r play fun
ddnitusova@ddnitusova ~]$ mv fun play
ddnitusova@ddnitusova ~]$ mv play/fun play/games
ddnitusova@ddnitusova ~]$ chmod u-r feathers
ddnitusova@ddnitusova ~]$ cat feathers
cat: feathers: Отказано в доступе
ddnitusova@ddnitusova ~]$ cp feathers monthly
cp: невозможно открыть «feathers» для чтения: Отказано в доступе
ddnitusova@ddnitusova ~]$ chmod u+r feathers
ddnitusova@ddnitusova ~]$ chmod u-x play
ddnitusova@ddnitusova ~]$ cd play
bash: cd: play: Отказано в доступе
ddnitusova@ddnitusova ~]$ chmod u+x play
```

Figure 3.8: Рисунок 8

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

Команда mount: (рис. 3.9) 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.



```
ddnitusova@ddnitusova:~  
Файл Правка Вид Поиск Терминал Справка  
MOUNT(8) System Administration MOUNT(8)  
NAME  
mount - mount a filesystem  
SYNOPSIS  
mount [-lhV]  
mount -a [-fFnrsvw] [-t vfstype] [-O optlist]  
mount [-fnrsvw] [-o option[,option]...] device|dir  
mount [-fnrsvw] [-t vfstype] [-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 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 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 directory or 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 --target or --source options to avoid ambivalent interpretation of the given argument. For  
example
```

Figure 3.9: Рисунок 9

Команда fsck: (рис. 3.10) 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.

```
Файл  Правка  Вид  Поиск  Терминал  Справка
FSCK(8)                                     System Administration                                     FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lrsAVRTMNP] [-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 ext2 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 /sbin first, then in /etc/fs and /etc, and finally in the

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

Figure 3.10: Рисунок 10

Команда `mkfs`: (рис. 3.11) 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).

```
ddnitusova@ddnitusova:~  
Файл Правка Вид Поиск Терминал Справка  
MKFS(8) System Administration MKFS(8)  
NAME  
mkfs - build a Linux filesystem  
SYNOPSIS  
mkfs [options] [-t type] [fs-options] device [size]  
DESCRIPTION  
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 in a number of directories, like perhaps /sbin, /sbin/fs, /sbin/fs_d, /etc/fs, /etc (the precise list is defined at compile time but at least contains /sbin and /sbin/fs), and finally in the directories listed in the PATH environment variable. 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. Although not guaranteed, the following options are supported by most filesystem builders.  
-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.  
Manual page mkfs(8) line 1 (press h for help or q to quit)
```

Figure 3.11: Рисунок 11

Команда kill: (рис. 3.12) 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.

```
ddnitusova@ddnitusova:~  
Файл Правка Вид Поиск Терминал Справка  
User Commands KILL(1)  
NAME  
kill - terminate a process  
SYNOPSIS  
kill [-s signal|-p] [-q signal] [-w] [-l] pid...  
kill -l [signal]  
DESCRIPTION  
The command kill sends the specified signal to the specified process or process group. If no signal is specified, the TERM signal is sent. The TERM signal will kill processes which do not catch this signal. For other processes, it may be necessary to use the KILL (9) signal, since this signal cannot be caught.  
Most modern shells have a builtin kill function, with a usage rather similar to that of the command described here. The 'a' and 'p' options, and the possibility to specify processes by command name are a local extension.  
If sig is 0, then no signal is sent, but error checking is still performed.  
OPTIONS  
pid... Specify the list of processes that kill should signal. Each pid can be one of five things:  
-p where p is larger than 0. The process with pid p will be signaled.  
-0 All processes in the current process group are signaled.  
-1 All processes with pid larger than 1 will be signaled.  
-n where n is larger than 1. All processes in process group n are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either the signal must be specified first, or the argument must be preceded by a '-' option, otherwise it will be taken as the signal to send.  
cassandra@centos  
Manual page kill(1) line 1 (press h for help or q to quit)
```

Figure 3.12: Рисунок 12

4 Контрольные вопросы

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

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.

- 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, 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 Содержит 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, 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.
- 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 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 файл_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 r чтение 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.