

Отчет по лабораторной работе №5

Анализ файловой системы Linux. Команды для работы с файлами и каталогами

Валиева Марина Русланбековна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	19
4	Контрольные вопросы	20

Список иллюстраций

2.1	Копирование файла в домашний каталог	9
2.2	Создание директории	9
2.3	Перемещение файла	10
2.4	Переименование файла	10
2.5	Создание файла	11
2.6	Создание каталога	11
2.7	Перемещение файлов	12
2.8	Создание и перемещение каталогов	13
2.9	Определение опции команды	14
2.10	Просмотр содержимого	15
2.11	Проделка заданий	16
2.12	Читам ман по командам	16
2.13	fsck	17
2.14	mkfs	17
2.15	kill	18

Список таблиц

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

1. Выполним все примеры, приведённые в первой части описания лабораторной работы.

```
mrvalieva@dk8n57 ~ $ cd
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ cp abc1 apri
mrvalieva@dk8n57 ~ $ cp abc1 may
mrvalieva@dk8n57 ~ $ mkdir monthly
mrvalieva@dk8n57 ~ $ cp april may monthly
cp: не удалось выполнить stat для 'april': Нет такого файла или каталога
mrvalieva@dk8n57 ~ $ cp abc1 april
mrvalieva@dk8n57 ~ $ cp april may monthly
mrvalieva@dk8n57 ~ $ cp monthly/may monthly/june
mrvalieva@dk8n57 ~ $ ls monthly
april  june  may
mrvalieva@dk8n57 ~ $ mkdir monthly.00
mrvalieva@dk8n57 ~ $ cp -r monthly monthly.00
mrvalieva@dk8n57 ~ $ cp -r monthly.00 /tmp
```

```

mrvalieva@dk8n57 ~ $ cd
mrvalieva@dk8n57 ~ $ touch may
mrvalieva@dk8n57 ~ $ s -l may
bash: s: команда не найдена
mrvalieva@dk8n57 ~ $ ls -l may
-rw-r--r-- 1 mrvalieva studsci 0 map  9 16:02 may
mrvalieva@dk8n57 ~ $ chmod u+x may
mrvalieva@dk8n57 ~ $ ls -l may
-rwxr--r-- 1 mrvalieva studsci 0 map  9 16:02 may
mrvalieva@dk8n57 ~ $ chmod u-x may
mrvalieva@dk8n57 ~ $ ls -l may
-rw-r--r-- 1 mrvalieva studsci 0 map  9 16:02 may
mrvalieva@dk8n57 ~ $ cd
mrvalieva@dk8n57 ~ $ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
mrvalieva@dk8n57 ~ $ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ chmod g-r,o-r monthly

mrvalieva@dk8n57 ~ $ cd
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ chmod g+w abc1
mrvalieva@dk8n57 ~ $
mrvalieva@dk8n57 ~ $
mrvalieva@dk8n57 ~ $ cp /usr/include/sys/io.h
cp: после '/usr/include/sys/io.h' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ cp /usr/include/linux/sysinfo.h ~
mrvalieva@dk8n57 ~ $ mv sysinfo.h equipment
mrvalieva@dk8n57 ~ $ mkdir ski.places
mrvalieva@dk8n57 ~ $ mv equipment ski.places/
mrvalieva@dk8n57 ~ $ mv ski.places/equipment ski.places/equiplist
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ cp abc1
cp: после 'abc1' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.

```

```

mrvalieva@dk8n57 ~ $ cd
mrvalieva@dk8n57 ~ $ mv april july
mrvalieva@dk8n57 ~ $ mv july monthly.00
mrvalieva@dk8n57 ~ $ ls monthly.00
july  monthly
mrvalieva@dk8n57 ~ $ cd
mrvalieva@dk8n57 ~ $ mv april july
mv: не удалось выполнить stat для 'april': Нет такого файла или каталога
mrvalieva@dk8n57 ~ $ mv july monthly.00
mv: не удалось выполнить stat для 'july': Нет такого файла или каталога
mrvalieva@dk8n57 ~ $ ls monthly.00
july  monthly
mrvalieva@dk8n57 ~ $ ls monthly.00
july  monthly
mrvalieva@dk8n57 ~ $ ls monthly.00
april july  june  may  monthly
mrvalieva@dk8n57 ~ $ mv monthly.00 monthly.01
mrvalieva@dk8n57 ~ $ mkdir reports
mrvalieva@dk8n57 ~ $ mv monthly.01 reports
mrvalieva@dk8n57 ~ $ mv reports/monthly.01 reports/monthly
mrvalieva@dk8n57 ~ $ touch may
mrvalieva@dk8n57 ~ $ s -l may
bash: s: команда не найдена
mrvalieva@dk8n57 ~ $ ls -l may
-rw-r--r-- 1 mrvalieva studsci 0 map  9 16:02 may
mrvalieva@dk8n57 ~ $ chmod u+x may
mrvalieva@dk8n57 ~ $ ls -l may
-rwxr--r-- 1 mrvalieva studsci 0 map  9 16:02 may
mrvalieva@dk8n57 ~ $ chmod u-x may
mrvalieva@dk8n57 ~ $ ls -l may
-rw-r--r-- 1 mrvalieva studsci 0 map  9 16:02 may
mrvalieva@dk8n57 ~ $ cd
mrvalieva@dk8n57 ~ $ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
mrvalieva@dk8n57 ~ $ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию
mrvalieva@dk8n57 ~ $ chmod g-r,o-r monthly
mrvalieva@dk8n57 ~ $ cd
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ chmod g+w abc1

```


2. Выполним следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:

2.1. Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовем его `equipment`. Если файла `io.h` нет, то используем любой другой файл в каталоге `/usr/include/sys/` вместо него.

```
mrvalieva@dk8n57 ~ $ cp /usr/include/sys/io.h
cp: после '/usr/include/sys/io.h' пропущен операнд, задающий целев
По команде «cp --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ cp /usr/include/linux/sysinfo.h ~
mrvalieva@dk8n57 ~ $ mv sysinfo.h equipment
mrvalieva@dk8n57 ~ $ mkdir ski.places
mrvalieva@dk8n57 ~ $ mv equipment ski.places/
mrvalieva@dk8n57 ~ $ mv ski.places/equipment ski.places/equiplist
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ cp abc1
```

Рис. 2.1: Копирование файла в домашний каталог

2.2. В домашнем каталоге создадим директорию `~/ski.places`.

```
mrvalieva@dk8n57 ~ $ cp /usr/include/sys/io.h
cp: после '/usr/include/sys/io.h' пропущен операнд, задающий целев
По команде «cp --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ cp /usr/include/linux/sysinfo.h ~
mrvalieva@dk8n57 ~ $ mv sysinfo.h equipment
mrvalieva@dk8n57 ~ $ mkdir ski.places
mrvalieva@dk8n57 ~ $ mv equipment ski.places/
mrvalieva@dk8n57 ~ $ mv ski.places/equipment ski.places/equiplist
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ cp abc1
```

Рис. 2.2: Создание директории

2.3. Переместим файл `equipment` в каталог `~/ski.places`.

```

mrvalieva@dk8n57 ~ $ cp /usr/include/sys/io.h
cp: после '/usr/include/sys/io.h' пропущен операнд, задающий целев
По команде «cp --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ cp /usr/include/linux/sysinfo.h ~
mrvalieva@dk8n57 ~ $ mv sysinfo.h equipment
mrvalieva@dk8n57 ~ $ mkdir ski.places
mrvalieva@dk8n57 ~ $ mv equipment ski.places/
mrvalieva@dk8n57 ~ $ mv ski.places/equipment ski.places/equiplist
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ cp abc1

```

Рис. 2.3: Перемещение файла

2.4. Переименуем файл ~/ski.places/equipment в ~/ski.places/equiplist

```

mrvalieva@dk8n57 ~ $ cp /usr/include/sys/io.h
cp: после '/usr/include/sys/io.h' пропущен операнд, задающий целев
По команде «cp --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ cp /usr/include/linux/sysinfo.h ~
mrvalieva@dk8n57 ~ $ mv sysinfo.h equipment
mrvalieva@dk8n57 ~ $ mkdir ski.places
mrvalieva@dk8n57 ~ $ mv equipment ski.places/
mrvalieva@dk8n57 ~ $ mv ski.places/equipment ski.places/equiplist
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ cp abc1

```

Рис. 2.4: Переименование файла

2.5. Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.places, назовем его equiplist2

```

mrvalieva@dk8n57 ~ $ cp /usr/include/sys/io.h
cp: после '/usr/include/sys/io.h' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ cp /usr/include/linux/sysinfo.h ~
mrvalieva@dk8n57 ~ $ mv sysinfo.h equipment
mrvalieva@dk8n57 ~ $ mkdir ski.places
mrvalieva@dk8n57 ~ $ mv equipment ski.places/
mrvalieva@dk8n57 ~ $ mv ski.places/equipment ski.places/equiplist
mrvalieva@dk8n57 ~ $ touch abc1
mrvalieva@dk8n57 ~ $ cp abc1

```

Рис. 2.5: Создание файла

2.6. Создадим каталог с именем equipment в каталоге ~/ski.places

```

mrvalieva@dk8n57 ~ $ cp abc1 ski.places/equiplist2
mrvalieva@dk8n57 ~ $ cd ski.places/
mrvalieva@dk8n57 ~/ski.places $ mkdir equipment
mrvalieva@dk8n57 ~/ski.places $ mv equiplist equipment/
mrvalieva@dk8n57 ~/ski.places $ mv equiplist2 equipment/
mrvalieva@dk8n57 ~/ski.places $ cd
mrvalieva@dk8n57 ~ $ mkdir newdir
mrvalieva@dk8n57 ~ $ mv newdir/ski.places
mv: после 'newdir/ski.places' пропущен операнд, задающий целевой файл
По команде «mv --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ mv newdir/ski.places/
mv: после 'newdir/ski.places/' пропущен операнд, задающий целевой файл
По команде «mv --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ mv newdir/ ski.places/
mrvalieva@dk8n57 ~ $ mv ski.places/newdir/ ski.places/plans
mrvalieva@dk8n57 ~ $

```

Рис. 2.6: Создание каталога

2.7. Переместим файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment.

```

mrvalieva@dk8n57 ~ $ cp abc1 ski.places/equiplist2
mrvalieva@dk8n57 ~ $ cd ski.places/
mrvalieva@dk8n57 ~/ski.places $ mkdir equipment
mrvalieva@dk8n57 ~/ski.places $ mv equiplist equipment/
mrvalieva@dk8n57 ~/ski.places $ mv equiplist2 equipment/
mrvalieva@dk8n57 ~/ski.places $ cd
mrvalieva@dk8n57 ~ $ mkdir newdir
mrvalieva@dk8n57 ~ $ mv newdir/ski.places
mv: после 'newdir/ski.places' пропущен операнд, задающий целевой файл
По команде «mv --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ mv newdir/ski.places/
mv: после 'newdir/ski.places/' пропущен операнд, задающий целевой файл
По команде «mv --help» можно получить дополнительную информацию.
mrvalieva@dk8n57 ~ $ mv newdir/ ski.places/
mrvalieva@dk8n57 ~ $ mv ski.places/newdir/ ski.places/plans
mrvalieva@dk8n57 ~ $

```

Рис. 2.7: Перемещение файлов

2.8. Создадим и переместим каталог ~/newdir в каталог ~/ski.places и назовем его plans.

```

mrvalieva@dk8n57 ~ $ mv newdir/ ski.plases/
mrvalieva@dk8n57 ~ $ mv ski.plases/newdir/ ski.plases/plans
mrvalieva@dk8n57 ~ $
mrvalieva@dk8n57 ~ $ mkdir australia play
mrvalieva@dk8n57 ~ $ touch my_os feathers
mrvalieva@dk8n57 ~ $ chmod 744 australia/
mrvalieva@dk8n57 ~ $ chmod 711 play/
mrvalieva@dk8n57 ~ $ chmod 544 my_os
mrvalieva@dk8n57 ~ $ chmod 664 feathers
mrvalieva@dk8n57 ~ $ ls -l
итого 49
-rw-rw-r-- 1 mrvalieva studsci 0 map 9 16:11 abc1
-rw-r--r-- 1 mrvalieva studsci 0 map 9 15:51 apri
drwxr--r-- 2 mrvalieva studsci 2048 map 9 16:16 australia
-rw-rw-r-- 1 mrvalieva studsci 0 map 9 16:18 feathers
drwxr-xr-x 3 mrvalieva studsci 2048 сен 21 11:09 GNUstep
-rw-r--r-- 1 mrvalieva studsci 812 ноя 9 14:52 hello.asm
-rw-r--r-- 1 mrvalieva studsci 0 map 9 16:02 may
drwxr-xr-x 2 mrvalieva studsci 2048 map 1 16:48 milk
drwx--x--x 2 mrvalieva studsci 2048 map 9 15:50 monthly
-r-xr--r-- 1 mrvalieva studsci 0 map 9 16:18 my_os
drwxr-xr-x 3 mrvalieva studsci 2048 сен 29 16:19 parentdir
drwxr-xr-x 3 mrvalieva studsci 2048 сен 29 16:39 parentdir1
drwxr-xr-x 3 mrvalieva studsci 2048 сен 29 16:22 parentdir2
drwxr-xr-x 2 mrvalieva studsci 2048 сен 29 16:36 parentdir3

```

Рис. 2.8: Создание и перемещение каталогов

3. Определите опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет

3.1. `drwxr-r- ... australia` 3.2. `drwx-x-x ... play` 3.3. `-r-xr-r- ... my_os` 3.4. `-rw-rw-r- ... feathers`

```

mrvalieva@dk8n57 ~ $ mv newdir/ ski.plases/
mrvalieva@dk8n57 ~ $ mv ski.plases/newdir/ ski.plases/plans
mrvalieva@dk8n57 ~ $
mrvalieva@dk8n57 ~ $ mkdir australia play
mrvalieva@dk8n57 ~ $ touch my_os feathers
mrvalieva@dk8n57 ~ $ chmod 744 australia/
mrvalieva@dk8n57 ~ $ chmod 711 play/
mrvalieva@dk8n57 ~ $ chmod 544 my_os
mrvalieva@dk8n57 ~ $ chmod 664 feathers
mrvalieva@dk8n57 ~ $ ls -l
итого 49
-rw-rw-r-- 1 mrvalieva studsci 0 map 9 16:11 abc1
-rw-r--r-- 1 mrvalieva studsci 0 map 9 15:51 apri
drwxr--r-- 2 mrvalieva studsci 2048 map 9 16:16 australia
-rw-rw-r-- 1 mrvalieva studsci 0 map 9 16:18 feathers
drwxr-xr-x 3 mrvalieva studsci 2048 сен 21 11:09 GNUstep
-rw-r--r-- 1 mrvalieva studsci 812 ноя 9 14:52 hello.asm
-rw-r--r-- 1 mrvalieva studsci 0 map 9 16:02 may
drwxr-xr-x 2 mrvalieva studsci 2048 map 1 16:48 milk
drwx--x--x 2 mrvalieva studsci 2048 map 9 15:50 monthly
-r-xr--r-- 1 mrvalieva studsci 0 map 9 16:18 my_os
drwxr-xr-x 3 mrvalieva studsci 2048 сен 29 16:19 parentdir
drwxr-xr-x 3 mrvalieva studsci 2048 сен 29 16:39 parentdir1
drwxr-xr-x 3 mrvalieva studsci 2048 сен 29 16:22 parentdir2
drwxr-xr-x 2 mrvalieva studsci 2048 сен 29 16:36 parentdir3

```

Рис. 2.9: Определение опции команды

При необходимости создадим нужные файлы

4. Прделаем приведённые ниже упражннения, записывая в отчёт по лабораторной работе используемые при этом команды: 4.1. Просмотрим содержимое файла /etc/password.

```

mrvalieva@dk8n57 ~ $ cat /etc/passwd
root:x:0:0:System user; root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:Mail program user:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucppublic:/bin/false
operator:x:11:0:operator:/root:/bin/bash
man:x:13:15:System user; man:/dev/null:/sbin/nologin
postmaster:x:14:12:Postmaster user:/var/spool/mail:/sbin/nologin
cron:x:16:16:A user for sys-process/cronbase:/var/spool/cron:/sbin/nolog
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:42:42:User for running GDM:/var/lib/gdm:/sbin/nologin

```

Рис. 2.10: Просмотр содержимого

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 вы попытаетесь просмотреть файл ~/feathers командой cat? 4.8. Что произойдёт, если вы попытаетесь скопировать файл ~/feathers? 4.9. Дадим владельцу файла ~/feathers право на чтение. 4.10. Лишим владельца каталога ~/play права на выполнение. 4.11. Перейдем в каталог ~/play. Что произошло? 4.12. Дадим владельцу каталога ~/play право на выполнение.

```

mrvalieva@dk8n57 ~ $ cp feathers file.old
mrvalieva@dk8n57 ~ $ mv file.old play
mrvalieva@dk8n57 ~ $ mkdir fun
mrvalieva@dk8n57 ~ $ vp -R play/ fun
bash: vp: команда не найдена
mrvalieva@dk8n57 ~ $ vp -R play/ fun/
bash: vp: команда не найдена
mrvalieva@dk8n57 ~ $ mv fun/ play/games
mrvalieva@dk8n57 ~ $ chmod u-r feathers
mrvalieva@dk8n57 ~ $ cat feathers
cat: feathers: Отказано в доступе
mrvalieva@dk8n57 ~ $ cp feathers feathers2
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
mrvalieva@dk8n57 ~ $ chmod u+r feathers
mrvalieva@dk8n57 ~ $ chmod u-r play
mrvalieva@dk8n57 ~ $ cd play
mrvalieva@dk8n57 ~/play $ cd
mrvalieva@dk8n57 ~ $ cd play/
mrvalieva@dk8n57 ~/play $ cd
mrvalieva@dk8n57 ~ $ chmod +x play

```

Рис. 2.11: Проделка заданий

5. Прочитаем ман по командам mount, fsck, mkfs, kill и кратко их охарактери-
зуем, приведя примеры

```

mrvalieva@dk8n57 ~ $ man mount
mrvalieva@dk8n57 ~ $ man mount
mrvalieva@dk8n57 ~ $ man fsck
mrvalieva@dk8n57 ~ $ man mkfs
mrvalieva@dk8n57 ~ $ man kill

```

Рис. 2.12: Читам ман по командам


```
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.
    filesystem can be a device name (e.g., /dev/hdc1, /dev/sdb2), a mount point
    (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g.,
    UUID=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.

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

Рис. 2.13: fsck

```
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.

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

Рис. 2.14: mkfs

```
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - send a signal to a process

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

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Particularly 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 processes except the kill process itself and init.

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

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

Рис. 2.15: kill

3 Выводы

В результате данной лабораторной работы я знакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрела практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

4 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.

Ответ: Ext2FS (расширенная файловая система номер два). Многие годы ext2 была файловой системой по умолчанию в GNU/Linux. Ext2 заменила собой Extended File System (вот откуда появилось “Second” в названии). В “новой” файловой системе были исправлены некоторые проблемы, а также убраны ограничения. Отличная стабильность, комплексные инструментальные средства для спасения удаленных файлов, очень долгое время перезагрузки после аварии, есть вероятность частичной или полной потери данных после аварии. Одним из главных недостатков “традиционных” файловых систем, подобных Ext2FS, является низкая сопротивляемость к резким системным сбоям (сбой питания или авария программного обеспечения) Ext3 (Расширенная файловая система номер три) - является наследником файловой системы Ext2FS. Ext3 совместима с Ext2, но обладает одной новой и очень интересной особенностью – запись. Процесс сохранения объекта происходит прежде чем запись в журнал. В результате мы получаем всегда последовательную файловую систему. Это приводит к тому, что при появлении проблем, проверка и восстановление происходят очень быстро. Время, потраченное на то, чтобы проверить файловую систему таким образом, пропорционально его фактическому использованию и не больше его размера. ReiserFS (Это тоже журналируемая файловая система подобно Ext3FS, но их внутренняя структура радикально отличается. В ReiserFS используется

концепция 15 бинарных деревьев (binary-tree), позаимствованная из программного обеспечения баз данных. JFS (журналируемая файловая система). JFS была разработана и использовалась IBM. Вначале JFS была закрытой системой, но недавно IBM решила открыть доступ для движения свободного программного обеспечения. Внутренняя структура JFS близка к ReiserFS. Средняя стабильность, нет комплексных инструментальных средств для спасения удаленных файлов, очень быстрая перезагрузка после аварии, очень хорошее восстановление данных после аварии.

2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры. Ответ: • Загрузочный блок занимает первый блок файловой системы. Только корневая файловая система имеет активный загрузочный блок, хотя место для него резервируется в каждой файловой системе. • Суперблок располагается непосредственно за загрузочным блоком и содержит самую общую информацию о ФС (размер ФС, размер области индексных дескрипторов, их число, список свободных блоков, свободные индексные дескрипторы и т. д.). Суперблок всегда находится в оперативной памяти. Различные версии ОС Unix способны поддерживать разные типы файловых систем. Поэтому у структуры суперблока могут быть варианты (сведения о свободных блоках, например, часто хранятся не как список, а как шкала бит), но суперблок всегда располагается за загрузочным блоком. При монтировании файловой системы в оперативной памяти создается копия ее суперблока. Все последующие операции по созданию и удалению файлов влекут изменения копии суперблока в оперативной памяти. Эта копия периодически записывается на магнитный диск. Обычно причиной повреждения файловой системы является отключение электропитания (или зависание ОС) в тот момент, когда система производит копирование суперблока из оперативной памяти на магнитный диск. • Область индексных дескрипторов содержит описатели файлов (inode). С каждым файлом связан один inode, но одному inode

может соответствовать несколько файлов. Binode хранится вся информация о файле, кроме его имени. Область индексных дескрипторов имеет фиксированный формат и располагается непосредственно за суперблоком. Общее число описателей и, следовательно, максимальное число файлов задается в момент создания файловой системы. Описатели нумеруются натуральными числами. Первый описатель используется ОС для описания специального файла (файла «Плохих блоков»). То есть поврежденные блоки раздела рассматриваются ОС как принадлежащие к специальному файлу и поэтому считаются «занятыми». Второй – описывает корневой каталог файловой системы. • В области данных расположены как обычные файлы, так и файлы каталогов (в том числе корневой каталог). Специальные файлы представлены в ФС только записями в соответствующих каталогах и индексными дескрипторами специального формата, т. е. места в области памяти не занимают.

3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе? Ответ: Команда `cat` - позволяет вывести на экран содержимое любого файла, однако в таком виде эта команда практически не используется. Если файл слишком большой, то его содержимое пролистается на экране, а Вы увидите только последние строки файла. С помощью этой команды можно комбинировать и объединять копии файлов, а также создавать новые файлы. Если набрать просто в командной строке `cat` и нажать `Enter`, то можно вводить (и соответственно видеть) текст на экране. Повторное нажатие клавиши `Enter` удвоит строку и позволит начать следующую. Когда текст набран, следует одновременно нажать клавиши `Ctrl` и `d`.
4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы? Ответ: Некорректность файловой системы может возникать: • В результате насильственного преры-

вания операций ввода-вывода, выполняемых непосредственно с диском. • В результате нарушения работы дискового кэша. Кэширование данных с диска предполагает, что в течение некоторого времени результаты операций ввода-вывода никак не сказываются на содержимом диска — все изменения происходят с копиями блоков диска, временно хранящихся в буферах оперативной памяти (в этих буферах оседают данные из пользовательских файлов и служебная информация файловой системы, такая как каталоги, индексные дескрипторы, списки свободных, занятых и поврежденных блоков и т. п.)

5. Как создаётся файловая система? Ответ: Общее дерево файлов и каталогов системы Linux формируется из отдельных “ветвей”, соответствующих различным физическим носителям. В UNIX нет понятия “форматирования диска” (и команды форматирования), а используется понятие “создание файловой системы”. Когда мы получаем новый носитель, например, жесткий диск, мы должны создать на нем файловую систему. То есть каждому носителю ставится в соответствие отдельная файловая система. Чтобы эту файловую систему использовать для записи в нее файлов, надо ее вначале подключить в общее дерево каталогов (“смонтировать”). Вот и получается, что можно говорить о монтировании файловых систем или о монтировании носителей (с созданными на них файловыми системами). Например, создается файловая система типа ext2fs. Создание файловой системы типа ext2fs подразумевает создание в данном разделе на диске суперблока, таблицы индексных дескрипторов и совокупности блоков данных. Делается все это все с помощью команды mkfs. В простейшем случае достаточно дать эту команду в следующем формате: [root]# mkfs -t ext2 /dev/hda5, где /dev/hda5 надо заменить указанием на соответствующее устройство или раздел. Например, если вы хотите создать файловую систему на дискете, то команда примет вид: [root]# mkfs -t ext2 /dev/fd0 После выполнения команды mkfs в указанном разделе будет создана файловая система ext2fs. В

новой файловой системе автоматически создается один каталог с именем `lost+found`. Он используется в экстренных случаях программой `fsck`, поэтому не удаляйте его. Для того, чтобы начать работать с новой файловой системой, необходимо подключить ее в общее дерево каталогов, что делается с помощью команды `mount`. В качестве параметров команде `mount` надо, как минимум, указать устройство и “точку монтирования”. Точкой монтирования называется тот каталог в уже существующем и известном системе дереве каталогов, который будет теперь служить корневым каталогом для подключаемой файловой системы. После монтирования файловой системы в каталог `/mnt/disk2` прежнее содержимое этого каталога станет для вас недоступно до тех пор, пока вы не размонтируете вновь подключенную файловую систему. Прежнее содержимое не уничтожается, а просто становится временно недоступным. Поэтому в качестве точек монтирования лучше использовать пустые каталоги (заранее заготовленные).

6. Дайте характеристику командам, которые позволяют просмотреть текстовые файлы. Ответ: Для просмотра небольших файлов удобно пользоваться командой `cat`. Формат команды: `cat имя-файла` Для просмотра больших файлов используйте команду `less` — она позволяет осуществлять постраничный просмотр файлов (длина страницы соответствует размеру экрана). Формат команды: `less имя-файла` Для управления процессом просмотра можно использовать следующие управляющие клавиши: - `Space` — переход на следующую страницу, - `ENTER` — сдвиг вперёд на одну строку, - `b` — возврат на предыдущую страницу, - `h` — обращение за подсказкой, - `q` — выход в режим командной строки. Для просмотра начала файла можно воспользоваться командой `head`. По умолчанию она выводит первые 10 строк файла. Формат команды: `head [-n] имя-файла`, где `n` — количество выводимых строк. Команда `tail` выводит несколько (по умолчанию 10) последних строк файла. Формат команды: `tail [-n] имя-файла`, где `n` — количество выводимых строк.

7. Приведите основные возможности команды `cp` в Linux. Ответ: Копирование отдельных файлов Для копирования файла следует использовать утилиту `cp` аргументами, представленными путями к исходному и целевому файлам. Копирование файлов в другую директорию В том случае, если в качестве пути к целевому файлу используется путь к директории, исходные файлы будут скопированы в эту целевую директорию. Команда `cp -r` Для копирования директорий целиком следует использовать команду `cp -r` (параметр `-r` позволяет осуществлять рекурсивное копирование всех файлов из всех поддиректорий). Копирование множества файлов в директорию Вы также можете использовать утилиту `cp` для копирования множества файлов в одну директорию. В этом случае последний аргумент (аргумент, указывающий на цель) должен быть представлен путем к директории. Команда `cp -i` Для предотвращения перезаписи существующих файлов в ходе использования утилиты `cp` следует использовать параметр `-i` (для активации интерактивного режима копирования).
8. Назовите и дайте характеристику командам перемещения и переименования файлов и каталогов. Ответ: Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов. Формат команды `mv`: `mv [-опции] старый_файл новый_файл` Примеры: • Переименование файлов в текущем каталоге. Изменить название файла `april` на `july` в домашнем каталоге: `cd mv april july` • Перемещение файлов в другой каталог. Переместить файл `july` в каталог `monthly.00`: `mv july monthly.00 ls monthly.00` Результат: `april july june may`. Если необходим запрос подтверждения о перезаписи файла, то нужно использовать опцию `i`. • Переименование каталогов в текущем каталоге. Переименовать каталог `monthly.00` в `monthly.01` `mv monthly.00 monthly.01` • Перемещение каталога в другой каталог. Переместить каталог `monthly.01` в каталог `reports`: `mkdir reports mv monthly.01 reports` • Переименование каталога, не являющегося текущим. Переименовать каталог `reports/monthly.01` в `reports/monthly`: `mv reports/monthly.01 reports/monthly`

9. Что такое права доступа? Как они могут быть изменены? Ответ: Права доступа — совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам). Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Формат команды: `chmod режим имя_файла Режим` (в формате команды) 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 В работе с правами доступа можно использовать их цифровую запись (восьмеричное значение) вместо символьной.