

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет Физико-Математических Наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Операционные системы

Студент: Манаева Варвара Евгеньевна

Группа: НФИбд-01-20

Преподаватель: Кулябов Дмитрий Сергеевич

МОСКВА

2021 г.

Техническое оснащение:

- Персональный компьютер с операционной системой Windows 7;
 - Планшет для записи видеосопровождения и голосовых комментариев;
 - Виртуальная коробка VirtualBox, виртуальная машина с установленной на ней операционной системой CentOS;
 - Microsoft Teams, использующийся для записи скринкаста лабораторной работы;
 - Приложение MarkPad 2 для редактирования файлов формата *md*;
 - *pandoc* для конвертации файлов отчётов и презентаций.
-

Объект и предмет исследования: Файловая система *Linux*, команды для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

Цель: 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.

Задачи:

- 1) Ознакомиться со структурой файловой системы;
 - 2) Ознакомиться с именованиями в файловой системе *Linux*;
 - 3) Ознакомиться с содержанием каталогов файловой системы *Linux*;
 - 4) Приобрести навыки по применению команд для работы с файлами;
 - 5) Приобрести навыки по применению команд для работы с каталогами;
 - 6) Приобрести навыки по применению команд по управлению процессами и работами;
 - 7) Приобрести навыки по применению команд по проверке использования диска и обслуживанию файловой системы.
-

Теоретические вводные данные:

Команда для создания текстовых файлов. Для создания текстового файла удобно воспользоваться командой *touch*.

Формат команды:

touch имя-файла

Команды просмотра текстовых файлов. 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.

Копирование файлов и каталогов. Копирование файлов и каталогов осуществляется при помощи команды *cp*.

Формат команды:

cp [-опции] исходныйфайл целевойфайл

Примеры: 1) Копирование файла в текущем каталоге. Скопировать файл *~/abc1* в файл *april* и в файл *may*: *cd*

touch abc1

cp abc1 april
cp abc1 may

2) Копирование нескольких файлов в каталог. Скопировать файлы *april* и *may* в каталог *monthly*:

mkdir monthly
cp april may monthly

3) Копирование файлов в произвольном каталоге. Скопировать файл *monthly/may* в файл с именем *june*:

cp monthly/may monthly/june
ls monthly

Опция *i* в команде *cp* выведет на экран запрос подтверждения о перезаписи файла, если на место целевого файла вы поставите имя уже существующего файла.

Команда *cp* с опцией *r (recursive)* позволяет копировать каталоги вместе с входящими в них файлами и каталогами. Примеры: 1)

Копирование каталогов в текущем каталоге. Скопировать каталог *monthly* в каталог *monthly.00*:

mkdir monthly.00
cp -r monthly monthly.00

2) Копирование каталогов в произвольном каталоге. Скопировать каталог *monthly.00* в каталог */tmp*:

cp -r monthly.00 /tmp

Перемещение и переименование файлов и каталогов. Команды *mv* и *mkdir* предназначены для перемещения и переименования файлов и каталогов.

Формат команды *mv*:

mv [-опции] старыйфайл новыйфайл

Примеры:

1) Переименование файлов в текущем каталоге. Изменить название файла *april* на *july* в домашнем каталоге: *cd mv april july*

2) 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*.

3) Переименование каталогов в текущем каталоге. Переименовать каталог *monthly.00* в *monthly.01*:

mv monthly.00 monthly.01

4) Перемещение каталога в другой каталог. Переместить каталог *monthly.01* в каталог *reports*:

mkdir reports
mv monthly.01 reports

5) Переименование каталога, не являющегося текущим. Переименовать каталог *reports/monthly.01* в *reports/monthly*:

mv reports/monthly.01 reports/monthly

Права доступа. Каждый файл имеет права доступа.

В сведениях о файле или каталоге указываются:

- тип файла (символ (-) обозначает файл, а символ (d) — каталог);
- права для владельца файла (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует);
- права для членов группы (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует);
- права для всех остальных (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует).

Примеры: 1) Для файла (крайнее левое поле имеет значение -) владелец файла имеет право на чтение и запись (rw-), группа, в которую входит владелец файла, может читать файл (r--), все остальные могут читать файл (r--):

-rw-r--r--

2) Только владелец файла имеет право на чтение, изменение и выполнение файла:

`-rwx-----`

3) Владелец каталога (крайнее левое поле имеет значение d) имеет право на просмотр, изменение и доступа в каталог, члены группы могут входить и просматривать его, все остальные — только входить в каталог:

`drwxr-x--x`

Изменение прав доступа. 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.

Примеры:

1) Требуется создать файл *~/may* с правом выполнения для владельца:

cd

touch may

ls -l may

chmod u+x may

ls -l may

2) Требуется лишить владельца файла *~/may* права на выполнение:

chmod u-x may

ls -l may

3) Требуется создать каталог *monthly* с запретом на чтение для членов группы и всех остальных пользователей:

cd

mkdir monthly

chmod g-r, o-r monthly

4) Требуется создать файл *~/abc1* с правом записи для членов группы:

cd

touch abc1

chmod g+w abc1

Анализ файловой системы. Файловая система в *Linux* состоит из фалов и каталогов. Каждому физическому носителю соответствует своя файловая система. Существует несколько типов файловых систем. Перечислим наиболее часто встречающиеся типы:

- ext2fs (second extended filesystem);
- ext3fs (third extended file system);
- ext4 (fourth extended file system);
- ReiserFS;
- xfs;
- fat (file allocation table);
- ntfs (new technology file system).

Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой *mount* без параметров. В результате её применения можно получить примерно следующее:

```
mount
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec)
udev on /dev type tmpfs (rw,nosuid)
devpts on /dev/pts type devpts (rw,nosuid,noexec)
/dev/sda1 on /mnt/a type ext3 (rw,noatime)
/dev/sdb2 on /mnt/docs type reiserfs (rw,noatime)
shm on /dev/shm type tmpfs (rw,noexec,nosuid,nodev)
usbfs on /proc/bus/usb type usbfs (rw,noexec,nosuid,devmode=0664,devgid=85)
binfmt_misc on /proc/sys/fs/binfmtmisc type binfmtmisc (rw,noexec,nosuid,nodev)
nfsd on /proc/fs/nfs type nfsd (rw,noexec,nosuid,nodev)
```

В данном случае указаны имена устройств, названия соответствующих им точек монтирования (путь), тип файловой системы и параметрами монтирования. В контексте команды *mount* устройство — специальный файл устройства, с помощью которого операционная система получает доступ к аппаратному устройству.

Файлы устройств обычно располагаются в каталоге */dev*, имеют сокращённые имена (например, *sdaN*, *sdbN* или *hdaN*, *hdbN*, где *N* — порядковый номер устройства, *sd* — устройства *SCSI*, *hd* — устройства *MFM/IDE*). Точка монтирования — каталог (путь к каталогу), к которому присоединяются файлы устройств.

Другой способ определения смонтированных в операционной системе файловых систем — просмотр файла */etc/fstab*. Сделать это можно например с помощью команды *cat*:

```
cat /etc/fstab
/dev/hda1 / ext2 defaults 1 1
/dev/hda5 /home ext2 defaults 1 2
/dev/hda6 swap swap defaults 0 0 /dev/hdc /mnt/cdrom auto umask=0,user,noauto,ro,exec,users 0 0
none /mnt/floppy supermount dev=/dev/fd0,fs=ext2:vfat,--,sync,umask=0 0 0
none /proc proc defaults 0 0
none /dev/pts devpts mode=0622 0 0
```

В каждой строке этого файла указано:

- имя устройство;
- точка монтирования;
- тип файловой системы;
- опции монтирования;
- специальные флаги для утилиты *dump*;
- порядок проверки целостности файловой системы с помощью утилиты *fsck*.

Для определения объёма свободного пространства на файловой системе можно воспользоваться командой *df*, которая выведет на экран список всех файловых систем в соответствии с именами устройств, с указанием размера и точки монтирования. Например:

```
df
Filesystem 1024-blocks Used Available Capacity Mounted on
/dev/hda3 297635 169499 112764 60% /
```

С помощью команды *fsck* можно проверить (а в ряде случаев восстановить) целостность файловой системы:

Формат команды:

```
fsck имя_устройства
```

Пример:

```
fsck /dev/sda1
```

Этапы работы:

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

Скопировала файл *~/abc1* в файл *april* и в файл *may*. Скопировала файлы *april* и *may* в каталог *monthly*. Скопировала файл *monthly/may* в файл с именем *june*. Скопировала каталог *monthly* в каталог *monthly.00*. Скопировала каталог *monthly.00* в каталог */tmp*.

```

[vemanaeva@vemanaeva ~]$ cd
[vemanaeva@vemanaeva ~]$ touch abc1
[vemanaeva@vemanaeva ~]$ cp abc1 april
[vemanaeva@vemanaeva ~]$ cp abc1 may
[vemanaeva@vemanaeva ~]$ mkdir monthly
[vemanaeva@vemanaeva ~]$ cp april may monthly
[vemanaeva@vemanaeva ~]$ cp monthly/may monthly/june
[vemanaeva@vemanaeva ~]$ ls monthly
april  june  may
[vemanaeva@vemanaeva ~]$ mkdir monthly.00
[vemanaeva@vemanaeva ~]$ cp -r monthly monthly.00
[vemanaeva@vemanaeva ~]$ cp -r monthly.00 /tmp

```

Изменила название файла *april* на *july* в домашнем каталоге. Переместила файл *july* в каталог *monthly.00*. Переименовала каталог *monthly.00* в *monthly.01*. Переместила каталог *monthly.01* в каталог *reports*. Переименовала каталог *reports/monthly.01* в *reports/monthly*.

```

[vemanaeva@vemanaeva ~]$ cd
[vemanaeva@vemanaeva ~]$ mv april july
[vemanaeva@vemanaeva ~]$ mv july monthly.00
[vemanaeva@vemanaeva ~]$ ls monthly.00
july  monthly
[vemanaeva@vemanaeva ~]$ ls monthly.00/monthly
april  june  may
[vemanaeva@vemanaeva ~]$ mv monthly.00 monthly.01
[vemanaeva@vemanaeva ~]$ mkdir reports
[vemanaeva@vemanaeva ~]$ mv monthly.01
mv: после «monthly.01» пропущен операнд, задающий целевой файл
По команде «mv --help» можно получить дополнительную информацию.
[vemanaeva@vemanaeva ~]$ mv monthly.01 reports
[vemanaeva@vemanaeva ~]$ mv reports/monthly.01 reports/monthly
[vemanaeva@vemanaeva ~]$ ls reports
monthly
[vemanaeva@vemanaeva ~]$ █

```

Создала файл *~/may* с правом выполнения для владельца. Лишила владельца файла *~/may* права на выполнение. Создала каталог *monthly* с запретом на чтение для членов группы и всех остальных пользователей. Создала файл *~/abc1* с правом записи для членов группы.

```

[vemanaeva@vemanaeva ~]$ cd
[vemanaeva@vemanaeva ~]$ touch may
[vemanaeva@vemanaeva ~]$ ls -l may
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 16:55 may
[vemanaeva@vemanaeva ~]$ chmod u+x may
[vemanaeva@vemanaeva ~]$ ls -l may
-rwxrw-r--. 1 vemanaeva vemanaeva 0 май 15 16:55 may
[vemanaeva@vemanaeva ~]$ chmod u-x may
[vemanaeva@vemanaeva ~]$ ls -l may
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 16:55 may
[vemanaeva@vemanaeva ~]$ cd
[vemanaeva@vemanaeva ~]$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
[vemanaeva@vemanaeva ~]$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
[vemanaeva@vemanaeva ~]$ chmod g-r o-r monthly
chmod: невозможно получить доступ к «o-r»: Нет такого файла или каталога
[vemanaeva@vemanaeva ~]$ chmod g-r,o-r monthly
[vemanaeva@vemanaeva ~]$ ls -l monthly
итого 0
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 16:48 april
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 16:48 june
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 16:48 may
[vemanaeva@vemanaeva ~]$ ls -l
итого 0
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 16:47 abc1
drwxrwxr-x. 3 vemanaeva vemanaeva 83 май 15 13:14 lab05
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 16:55 may
drwx-wx--x. 2 vemanaeva vemanaeva 42 май 15 16:48 monthly
drwxrwxr-x. 3 vemanaeva vemanaeva 21 май 15 16:54 reports
drwxr-xr-x. 2 vemanaeva vemanaeva 6 май 1 13:14 Видео
drwxr-xr-x. 2 vemanaeva vemanaeva 6 май 1 13:14 Документы
drwxr-xr-x. 2 vemanaeva vemanaeva 6 май 1 13:14 Загрузки
drwxr-xr-x. 2 vemanaeva vemanaeva 6 май 1 13:14 Изображения
drwxr-xr-x. 2 vemanaeva vemanaeva 6 май 1 13:14 Музыка
drwxr-xr-x. 2 vemanaeva vemanaeva 6 май 1 13:14 Общеизвестные
drwxr-xr-x. 2 vemanaeva vemanaeva 6 май 1 13:14 Рабочий стол
drwxr-xr-x. 2 vemanaeva vemanaeva 6 май 1 13:14 Шаблоны

```

Воспользовалась командой *df*, которая выведет на экран список всех файловых систем в соответствии с именами устройств, с указанием размера и точки монтирования, для определения объема свободного пространства на файловой системе. С помощью команды *fsck* проверила целостность файловой системы.

```

[vemanaeva@vemanaeva ~]$ df
Файловая система      1K-блоков  Использовано  Доступно  Использовано%  Смонтировано в
devtmpfs                490560           0    490560           0% /dev
tmpfs                   507376           0    507376           0% /dev/shm
tmpfs                   507376       7816    499560           2% /run
tmpfs                   507376           0    507376           0% /sys/fs/cgroup
/dev/mapper/centos-root 38770180    5040872   33729308          14% /
/dev/sda1               1038336    175100    863236           17% /boot
tmpfs                   101476         20    101456            1% /run/user/1000
[vemanaeva@vemanaeva ~]$ fsck /dev/sda1
fsck из util-linux 2.23.2
If you wish to check the consistency of an XFS filesystem or
repair a damaged filesystem, see xfs_repair(8).

```

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

- 2.1. Скопировала файл */usr/include/sys/io.h* в домашний каталог с помощью команды *cp* и назвала его *equipment*, с помощью команды *mv*.
- 2.2. В домашнем каталоге создала директорию *~/ski.plases*.
- 2.3. Переместила файл *equipment* в каталог *~/ski.plases* командой *mv*.
- 2.4. Переименовала файл *~/ski.plases/equipment* в *~/ski.plases/equiplist* командой *mv*.
- 2.5. Создала в домашнем каталоге файл *abc1* и скопировала его в каталог *~/ski.plases* командой *cp*, назвала его *equiplist2* командой *mv*.
- 2.6. Создала каталог с именем *equipment* в каталоге *~/ski.plases* командой *mkdir*.
- 2.7. Переместила файлы *~/ski.plases/equiplist* и *equiplist2* в каталог *~/ski.plases/equipment* командой *mv*.

2.8. Создала и переместила каталог ~/newdir в каталог ~/ski.plases командами mkdir и mv и назвала его plans командой mv.

```
[vemanaeva@vemanaeva ~]$ cp /usr/include/sys/io.h equipment
[vemanaeva@vemanaeva ~]$ ls
equipment lab05 Видео Документы Загрузки Изображения Музыка Общедоступные Рабочий стол Шаблоны
[vemanaeva@vemanaeva ~]$ mkdir ski.plases
[vemanaeva@vemanaeva ~]$ mv equipment ski.plases
[vemanaeva@vemanaeva ~]$ mv ski.plases/equipment ski.plases/equiplist
[vemanaeva@vemanaeva ~]$ touch abc1
[vemanaeva@vemanaeva ~]$ cp abc1 ~/ski.plases
[vemanaeva@vemanaeva ~]$ mv ~/ski.plases/abc1 ~/ski.plases/equiplist2
[vemanaeva@vemanaeva ~]$ cd
[vemanaeva@vemanaeva ~]$ cd ski.plases/
[vemanaeva@vemanaeva ski.plases]$ mkdir equipment
[vemanaeva@vemanaeva ski.plases]$ mv equiplist2 equiplist2 /equipment
mv: указанная цель «/equipment» не является каталогом
[vemanaeva@vemanaeva ski.plases]$ mv equiplist2 equiplist2 ~/ski.plases/equipment
mv: не удалось выполнить stat для «equiplist2»: Нет такого файла или каталога
[vemanaeva@vemanaeva ski.plases]$ ls
equiplist2 equipment
[vemanaeva@vemanaeva ski.plases]$ mv equiplist2 ~/ski.plases/equipment
[vemanaeva@vemanaeva ski.plases]$ ls
equipment
[vemanaeva@vemanaeva ski.plases]$ ls ./equipment
equiplist2 equiplist2
[vemanaeva@vemanaeva ski.plases]$ cd
[vemanaeva@vemanaeva ~]$ mkdir newdir
[vemanaeva@vemanaeva ~]$ mv newdir ~/ski.plases
[vemanaeva@vemanaeva ~]$ ls
abc1 lab05 ski.plases Видео Документы Загрузки Изображения Музыка Общедоступные Рабочий стол Шаблоны
[vemanaeva@vemanaeva ~]$ ls ski.plases/
equipment newdir
[vemanaeva@vemanaeva ~]$ mv ~/ski.plases/newdir ~/ski.plases/plans
[vemanaeva@vemanaeva ~]$ ls ski.plases/
equipment plans
[vemanaeva@vemanaeva ~]$
```

3) Определила опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет. При необходимости создала нужные файлы.

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


```

[vemanaeva@vemanaeva test]$ chmod 744 australia
[vemanaeva@vemanaeva test]$ ls -l
итого 0
drwxr--r--. 2 vemanaeva vemanaeva 6 май 15 17:25 australia
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 17:19 feathers
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 17:19 my_os
drwxrwxr-x. 2 vemanaeva vemanaeva 6 май 15 17:25 play
[vemanaeva@vemanaeva test]$ chmod 711 play
[vemanaeva@vemanaeva test]$ ls -l
итого 0
drwxr--r--. 2 vemanaeva vemanaeva 6 май 15 17:25 australia
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 17:19 feathers
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 17:19 my_os
drwx--x--x. 2 vemanaeva vemanaeva 6 май 15 17:25 play
[vemanaeva@vemanaeva test]$ chmod 544 my_os
[vemanaeva@vemanaeva test]$ ls -l
итого 0
drwxr--r--. 2 vemanaeva vemanaeva 6 май 15 17:25 australia
-rw-rw-r--. 1 vemanaeva vemanaeva 0 май 15 17:19 feathers
-r-xr--r--. 1 vemanaeva vemanaeva 0 май 15 17:19 my_os
drwx--x--x. 2 vemanaeva vemanaeva 6 май 15 17:25 play
[vemanaeva@vemanaeva test]$ chmod 644 feathers
[vemanaeva@vemanaeva test]$ ls -l
итого 0
drwxr--r--. 2 vemanaeva vemanaeva 6 май 15 17:25 australia
-rw-r--r--. 1 vemanaeva vemanaeva 0 май 15 17:19 feathers
-r-xr--r--. 1 vemanaeva vemanaeva 0 май 15 17:19 my_os
drwx--x--x. 2 vemanaeva vemanaeva 6 май 15 17:25 play
[vemanaeva@vemanaeva test]$

```

4) Проделала приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:

4.1) Не просмотрела содержимое файла /etc/passwd, так как у меня его нет.

```

[vemanaeva@vemanaeva ~]$ cd
[vemanaeva@vemanaeva ~]$ cat ~/etc/passwd
cat: /home/vemanaeva/etc/passwd: Нет такого файла или каталога
[vemanaeva@vemanaeva ~]$

```

4.2) Скопировала файл ~/feathers в файл ~/file.old командой cp.

4.3) Переместила файл ~/file.old в каталог ~/play командой mv.

```

[vemanaeva@vemanaeva ~]$ cp feathers file.old
[vemanaeva@vemanaeva ~]$ mkdir australia play
[vemanaeva@vemanaeva ~]$ chmod 711 play
[vemanaeva@vemanaeva ~]$ mv file.old ~/play

```

4.4) Скопировала каталог ~/play в каталог ~/fun командой cp -r.

4.5) Переместила каталог ~/fun в каталог ~/play командой mv и назвала его games командой mv.

```

[vemanaeva@vemanaeva ~]$ cp -r ~/play ~/fun
[vemanaeva@vemanaeva ~]$ mv ~/fun ~/play
[vemanaeva@vemanaeva ~]$ cd play
[vemanaeva@vemanaeva play]$ mv fun games
[vemanaeva@vemanaeva play]$ ls
file.old  games

```

4.6) Лишила владельца файла ~/feathers права на чтение командой chmod u-r.

4.7) Если попытаться просмотреть файл ~/feathers командой cat, то выведется:

4.8) Если попытаться скопировать файл ~/feathers командой cp, то выведется:

4.9) Дала владельцу файла ~/feathers право на чтение командой chmod u+r.

4.10) Лишила владельца каталога ~/play права на выполнение командой chmod u-x.

4.11) Попыталась перейти в каталог ~/play командой cd.

4.12) Дала владельцу каталога ~/play право на выполнение командой chmod u+x.

```
[vemanaeva@vemanaeva play]$ cd ..
[vemanaeva@vemanaeva ~]$ chmod u-r feathers
[vemanaeva@vemanaeva ~]$ cat feathers
cat: feathers: Отказано в доступе
[vemanaeva@vemanaeva ~]$ cp feathers
cp: после «feathers» пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
[vemanaeva@vemanaeva ~]$ cp feathers ~/play
cp: невозможно открыть «feathers» для чтения: Отказано в доступе
[vemanaeva@vemanaeva ~]$ chmod u+r feathers
[vemanaeva@vemanaeva ~]$ chmod u-x ~/play
[vemanaeva@vemanaeva ~]$ cd ~/play
bash: cd: /home/vemanaeva/play: Отказано в доступе
[vemanaeva@vemanaeva ~]$ chmod u+x ~/play
[vemanaeva@vemanaeva ~]$
```

5) Прочитала *man* по командам *mount*, *fsck*, *mkfs*, *kill*.

```

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

        mount --target /mountpoint

    The listing and help.
    The listing mode is maintained for backward compatibility only.

    For more robust and definable output use findmnt(8), especially in your scripts. Note that control characters in the mountpoint name are replaced with '?'.

    mount [-l] [-t type]
        lists all mounted filesystems (of type type). The option -l adds the labels in this listing. See below.

```

```

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 directories listed in the PATH environment variable. Please see the filesystem-specific checker manual pages for further details.

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

    -r      Report certain statistics for each fsck when it completes. These statistics include the exit status, the maximum run set size (in kilobytes), the elapsed all-clock time and the user and system CPU time used by the fsck run. For example:

        /dev/sda1: status 0, rss 92828, real 4.002804, user 2.677592, sys 0.86186

    -s      Serialize fsck operations. This is a good idea if you are checking multiple filesystems and the checkers are in an interactive mode. (Note: e2fsck(8) runs in an interactive mode by default. To make e2fsck(8) run in a non-interactive mode, you must either specify the -p or -a option, if you wish for errors to be corrected automatically, or the -n option if you do not.)

```

mkfs(8)	System Administration	mkfs(8)
NAME		
mkfs - build a Linux filesystem		
SYNOPSIS		
mkfs [options] [-t <i>type</i>] [<i>fs-options</i>] <i>device</i> [<i>size</i>]		
DESCRIPTION		
mkfs is used to build a Linux filesystem on a device, usually a hard disk partition. The <i>device</i> argument is either the device name (e.g. <i>/dev/hda1</i> , <i>/dev/sdb2</i>), or a regular file that shall contain the filesystem. The <i>size</i> 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 (<i>mkfs.<i>fstype</i></i>) available under Linux. The filesystem-specific builder is searched for in a number of directories, like perhaps <i>/sbin</i> , <i>/sbin/fs</i> , <i>/sbin/fs.d</i> , <i>/etc/fs</i> , <i>/etc</i> (the precise list is defined at compile time but at least contains <i>/sbin</i> and <i>/sbin/fs</i>), 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 <i>type</i> Specify the <i>type</i> of filesystem to be built. If not specified, the default filesystem type (currently ext2) is used.		
<i>fs-options</i> 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.		
-V, --version Display version information and exit. (Option -V will display version information only when it is the only parameter, otherwise it will work as --verbose.)		
-h, --help Display help and exit.		
BUGS		
All generic options must precede and not be combined with filesystem-specific options. Some filesystem-specific programs do not support the -V (verbose) option, nor return meaningful exit codes. Also, some filesystem-specific programs do not automatically detect the device size and require the <i>size</i> parameter to be specified.		
AUTHORS		
David Engel (david@ods.com) Fred N. van Kempen (waltje@uwalnt.nl.mugnet.org) Ron Sommeling (sommel@sci.kun.nl) The manual page was shamelessly adapted from Remy Card's version for the ext2 filesystem.		

kill(1)	User Commands	kill(1)
NAME		
kill - terminate a process		
SYNOPSIS		
kill [-s <i>signal</i>][-p] [-q <i>sigval</i>] [-n] [--] <i>pid</i> ...		
kill -l [<i>signal</i>]		
DESCRIPTION		
The command <i>kill</i> 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		
<i>pid</i> ... Specify the list of processes that kill should signal. Each <i>pid</i> can be one of five things:		
<i>n</i> where <i>n</i> is larger than 0. The process with <i>pid n</i> will be signaled.		
0 All processes in the current process group are signaled.		
-1 All processes with <i>pid</i> larger than 1 will be signaled.		
- <i>n</i> where <i>n</i> is larger than 1. All processes in process group <i>n</i> 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.		
<i>commandname</i> All processes invoked using that name will be signaled.		
-s, --signal <i>signal</i> Specify the signal to send. The signal may be given as a signal name or number.		
-l, --list [<i>signal</i>] Print a list of signal names, or convert signal given as argument to a name. The signals are found in <i>/usr/include/linux/signal.h</i>		
-L, --table Similar to -l, but will print signal names and their corresponding numbers.		
-a, --all Do not restrict the <i>commandname</i> -to- <i>pid</i> conversion to processes with the same uid as the present process.		

Краткая характеристика:

- *mount* применяется для монтирования файловых систем.
- *fsck* восстанавливает повреждённую файловую систему или проверяет на целостность.
- *mkfs* создаёт новую файловую систему.
- *kill* используется для принудительного завершения работы приложений.

Выводы: ознакомилась с файловой системой *Linux*, её структурой, именами и содержанием каталогов. 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.

Контрольные вопросы:

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

Ответ: Характеристика файловой системы, которая использовалась в данной лабораторной работе: Файлы: *abc1*, *april*, *may*, *june*, *july*, *isdv4.h*, *equipment*, *equiplist*, *equiplist2*, *my_os*, *feathers*, *file.old*. Каталоги: *monthly*, *monthly.00*, *tmp*, *monthly.01*, *reports*, *usr*, *include*, *xorg*, *ski.plases*, *equipment*, *newdir*, *plans*, *australia*, *play*, *etc*, *fun*, *games*.

2) Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.

Ответ: Пример общей структуры файловой системы: */home/pdarzhankina/monthly/april*, где */home/pdarzhankina* – домашний каталог, */monthly* – каталог, находящийся в домашнем и содержащий файл, */april* – файл, находящийся в каталоге.

3) Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?

Ответ: Чтобы содержимое некоторой файловой системы было доступно операционной системе, должно быть выполнено монтирование тома.

4) Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?

Ответ: Основные причины нарушения целостности файловой системы:

- Один блок адресуется несколькими *inode* (принадлежит нескольким файлам);
- Блок помечен как свободный, но в то же время занят (на него ссылается *inode*);
- Блок помечен как занятый, но в то же время свободен (ни один *inode* на него не ссылается);
- Неправильное число ссылок в *inode* (недостаток или избыток ссылающихся записей в каталогах);
- Несовпадение между размером файла и суммарным размером адресуемых *inode* блоков;
- Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы);
- "Потерянные" файлы (правильные *inode*, на которые не ссылаются записи каталогов);
- Недопустимые или неразмещенные номера *inode* в записях каталогов.

Чтобы устранить повреждения файловой системы используется команда *fsck*.

5) Как создаётся файловая система?

Ответ: Команда *mkfs* создаёт новую файловую систему.

6) Дайте характеристику командам, которые позволяют просмотреть текстовые файлы.

Ответ: Характеристика команд, которые позволяют просмотреть текстовые файлы:

- для просмотра небольших файлов удобно пользоваться командой *cat*.
- для просмотра больших файлов используйте команду *less* — она позволяет осуществлять постраничный просмотр файлов.
- для просмотра начала файла можно воспользоваться командой *head*, по умолчанию она выводит первые 10 строк файла.
- команда *tail* выводит несколько (по умолчанию 10) последних строк файла.

7) Приведите основные возможности команды *cp* в Linux.

Ответ: Основные возможности команды *cp*:

- копирование файла в текущем каталоге;
- копирование нескольких файлов в каталог;
- копирование файлов в произвольном каталоге.

Опция *i* в команде *cp* выведет на экран запрос подтверждения о перезаписи файла, если на место целевого файла вы поставите имя уже существующего файла. Команда *cp* с опцией *r* (*recursive*) позволяет копировать каталоги вместе с входящими в них файлами и каталогами.

8) Назовите и дайте характеристику командам перемещения и переименования файлов и каталогов.

Ответ: Характеристика команд перемещения и переименования файлов и каталогов:

- переименование файлов в текущем каталоге:
mv <старое_название_файла> <новое_название_файла>
- перемещение файлов в другой каталог:
mv <название_файла> <название_каталога>
Если необходим запрос подтверждения о перезаписи файла, то нужно использовать опцию *i*.
- переименование каталогов в текущем каталоге:
mv <старое_название_каталога> <новое_название_каталога>
- перемещение каталога в другой каталог:
mv <старый_каталог> <новый_каталог>
- переименование каталога, не являющегося текущим:
mv <каталог/староеназваниекаталога> <каталог/новоеназваниекаталога>

9) Что такое права доступа? Как они могут быть изменены?

Ответ: Каждый файл или каталог имеет права доступа: чтение (разрешены просмотр и копирование файла, разрешён просмотр списка входящих в каталог файлов), запись (разрешены изменение и переименование файла, разрешены создание и удаление файлов каталога), выполнение (разрешено выполнение файла, разрешён доступ в каталог и есть возможность сделать его текущим). Они могут быть изменены командой *chmod*.