

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Операционные системы

Студент: Ким Реачна

Группа: НПИбд-02-20

Москва

2021г.

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.

Теоретические данные:

Команда для создания текстовых файлов `touch .`

Формат команды: `touch имя-файла`

Команды просмотра текстовых файлов:

- Для просмотра небольших файлов : пользоваться командой `cat .`

Формат: `cat имя-файла`

- Для просмотра больших файлов : используйте команду `less` .

Формат: `less имя-файла`

- Для просмотра начала файла можно воспользоваться командой `head` . По умолчанию она выводит первые 10 строк файла.

Формат: `head [-n] имя-файла` , где `n` — количество выводимых строк.

- Команда `tail` выводит несколько (по умолчанию 10) последних строк файла.

Формат: `tail [-n] имя-файла` , где `n` — количество выводимых строк.

Копирование файлов и каталогов `cp` :

Формат: `cp [-опции] исходный_файл целевой_файл` .

Перемещение и переименование файлов и каталогов `mv` :

Формат: `mv [-опции] старый_файл новый_файл` .

Выполнение работы:

1. Выполните все примеры, приведённые в первой части описания лабораторной работы [Material link](#):

Копирование файлов и каталогов

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
```

```
kim@kim-VirtualBox:~$ cd
kim@kim-VirtualBox:~$ touch abcl
kim@kim-VirtualBox:~$ cp abcl april
kim@kim-VirtualBox:~$ cp abcl may
kim@kim-VirtualBox:~$ ls
abcl  Desktop  file.txt  Music  playful  Videos
april Documents Laboratory3 newdir  Public  work
bin   Downloads may      Pictures Templates
kim@kim-VirtualBox:~$
```

С помощью команды `ls` мы можем видеть новые файлы.

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

```
mkdir monthly
cp april may monthly
```

```
kim@kim-VirtualBox:~$ mkdir monthly
kim@kim-VirtualBox:~$ cp april may monthly
kim@kim-VirtualBox:~$ ls -/monthly
ls: invalid option -- '/'
Try 'ls --help' for more information.
kim@kim-VirtualBox:~$ ls ~/monthly
april  may
kim@kim-VirtualBox:~$
```

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

```
cp monthly/may monthly/june
ls monthly
```

```
kim@kim-VirtualBox:~$ cp monthly/may monthly/june
kim@kim-VirtualBox:~$ ls monthly
april  june  may
kim@kim-VirtualBox:~$
```

Мы видим, что отображаются скопированные файлы

Команда `cp` с опцией `r` (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами: `cp -r`

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

```
mkdir monthly.00
cp -r monthly monthly.00
```

```
kim@kim-VirtualBox:~$ mkdir monthly.00
kim@kim-VirtualBox:~$ cp -r monthly monthly.00
kim@kim-VirtualBox:~$ ls monthly.00
monthly
kim@kim-VirtualBox:~$
```

5. Копирование каталогов в произвольном каталоге. Скопировать каталог monthly.00 в каталог /tmp:

```
cp -r monthly.00 /tmp
```

```
kim@kim-VirtualBox:~$ cp -r monthly.00 /tmp
kim@kim-VirtualBox:~$ ls /tmp
config-err-xmYKKA
monthly.00
ssh-9i4cPv2Kua5B
systemd-private-eaade02237f94ab597312fcb5610f781-colord.service-AJL4rf
systemd-private-eaade02237f94ab597312fcb5610f781-ModemManager.service-jY3hyh
systemd-private-eaade02237f94ab597312fcb5610f781-switcheroo-control.service-Sgsx
bf
systemd-private-eaade02237f94ab597312fcb5610f781-systemd-logind.service-pyezqg
systemd-private-eaade02237f94ab597312fcb5610f781-systemd-resolved.service-p3wJJg
systemd-private-eaade02237f94ab597312fcb5610f781-systemd-timesyncd.service-mUNNr
g
systemd-private-eaade02237f94ab597312fcb5610f781-upower.service-EVdW9h
tracker-extract-files.1000
VMwareDnD
```

Теперь мы видим, что каталог monthly находится в каталоге /tmp.

Перемещение и переименование файлов и каталогов

Примеры:

1. Переименование файлов в текущем каталоге. Изменить название файла april на july в домашнем каталоге:

```
cd
mv april july
```

```
kim@kim-VirtualBox:~$ cd
kim@kim-VirtualBox:~$ mv april july
kim@kim-VirtualBox:~$ ls
abcl      Documents  july        monthly    newdir     Public     work
bin       Downloads Laboratory3 monthly.00  Pictures   Templates
Desktop   file.txt   may         Music      playful    Videos
kim@kim-VirtualBox:~$
```

мы видим, что файл успешно переименован.

2. Перемещение файлов в другой каталог. Переместить файл july в каталог monthly.00:

```
mv july monthly.00
ls monthly.00
```

```
kim@kim-VirtualBox:~$ mv july monthly.00
kim@kim-VirtualBox:~$ ls monthly.00
july  monthly
kim@kim-VirtualBox:~$
```

Мы смотрим на каталог monthly.00 и видим, что файл july успешно перемещен в него.

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

```
mv monthly.00 monthly.01
```

```
kim@kim-VirtualBox:~$ mv monthly.00 monthly.01
kim@kim-VirtualBox:~$ ls monthly.00
ls: cannot access 'monthly.00': No such file or directory
kim@kim-VirtualBox:~$ ls
abcl      Documents  Laboratory3  monthly.01  Pictures   Templates
bin       Downloads  may         Music      playful    Videos
Desktop   file.txt   monthly     newdir     Public     work
kim@kim-VirtualBox:~$
```

Мы смотрим домашний каталог и видим, что каталог monthly.00 был переименован в monthly.01.

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

```
mkdir reports
mv monthly.01 reports
```

```
kim@kim-VirtualBox:~$ mkdir reports
kim@kim-VirtualBox:~$ mv monthly.01 reports
kim@kim-VirtualBox:~$ ls reports
monthly.01
kim@kim-VirtualBox:~$
```

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

```
mv reports/monthly.01 reports/monthly
```

```
kim@kim-VirtualBox:~$ mv reports/monthly.01 reports/monthly
kim@kim-VirtualBox:~$ ls reports
monthly
kim@kim-VirtualBox:~$
```

Изменение прав доступа

Примеры:

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

```
cd
touch may
ls -l may
chmod u+x may
ls -l may
```

```
kim@kim-VirtualBox:~$ cd
kim@kim-VirtualBox:~$ touch may
kim@kim-VirtualBox:~$ ls -l may
-rw-rw-r-- 1 kim kim 0 мая 10 23:14 may
kim@kim-VirtualBox:~$ chmod u+x may
kim@kim-VirtualBox:~$ ls -l may
-rwxrw-r-- 1 kim kim 0 мая 10 23:14 may
kim@kim-VirtualBox:~$
```

С помощью команды `ls` мы можем видеть, что выполненное правильно.

2. Требуется лишить владельца файла ~/may права на выполнение(u-x):

```
chmod u-x may
ls -l may
```

```
kim@kim-VirtualBox:~$ chmod u-x may
kim@kim-VirtualBox:~$ ls -l may
-rw-rw-r-- 1 kim kim 0 мая 10 23:14 may
kim@kim-VirtualBox:~$
```

3. Требуется создать каталог monthly с запретом на чтение для членов группы и всех остальных пользователей (g-r, o-r):

```
cd
mkdir monthly
chmod g-r monthly
chmod o-r monthly
```

```
kim@kim-VirtualBox:~$ cd
kim@kim-VirtualBox:~$ mkdir monthly
mkdir: cannot create directory 'monthly': File exists
kim@kim-VirtualBox:~$ ls -l monthly
total 0
-rw-rw-r-- 1 kim kim 0 мая 10 23:00 april
-rw-rw-r-- 1 kim kim 0 мая 10 23:02 june
-rw-rw-r-- 1 kim kim 0 мая 10 23:00 may
kim@kim-VirtualBox:~$ chmod g-r monthly
kim@kim-VirtualBox:~$ chmod o-r monthly
kim@kim-VirtualBox:~$ ls -l monthly
```

4. Требуется создать файл ~/abc1 с правом записи для членов группы(g+w):

```
cd
touch abc1
chmod g+w abc1
```

```
kim@kim-VirtualBox:~$ cd
kim@kim-VirtualBox:~$ touch abc1
kim@kim-VirtualBox:~$ chmod g+w abc1
kim@kim-VirtualBox:~$ ls -l abc1
-rw-rw-r-- 1 kim kim 0 мая 10 23:21 abc1
kim@kim-VirtualBox:~$
```

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

- 2.1. Скопируйте файл /usr/include/sys/io.h в домашний каталог и назовите его equipment. Если файла io.h нет, то используйте любой другой файл в каталоге /usr/include/sys/ вместо него.

```

kim@kim-VirtualBox:~$ cp /usr/include/sys/io.h /home/kim
kim@kim-VirtualBox:~$ ls
abcl      Documents  io.h      monthly  Pictures  reports  work
bin       Downloads Laboratory3 Music     playful  Templates
Desktop   file.txt   may      newdir   Public    Videos
kim@kim-VirtualBox:~$ mv io.h equipment
kim@kim-VirtualBox:~$ ls
abcl      Documents  file.txt   monthly  Pictures  reports  work
bin       Downloads Laboratory3 Music     playful  Templates
Desktop   equipment  may      newdir   Public    Videos
kim@kim-VirtualBox:~$

```

Сначала у меня возникла проблема с тем, что при запуске команды `cd /usr/include/sys` я не смог найти файл `usr/include/sys/`, поэтому я установил его с помощью `sudo apt-get install libc6-dev-i386`, а затем он заработал.

2.2. В домашнем каталоге создайте директорию `~/ski.places`:

```

kim@kim-VirtualBox:~$ cd
kim@kim-VirtualBox:~$ mkdir ski.places
kim@kim-VirtualBox:~$ ls
abcl      Documents  file.txt   monthly  Pictures  reports  Videos
bin       Downloads Laboratory3 Music     playful  ski.places  work
Desktop   equipment  may      newdir   Public    Templates
kim@kim-VirtualBox:~$

```

мы создаем каталог `ski.places` с помощью команды `mkdir` и смотрим каталог с помощью команды `ls`.

2.3. Переместите файл `equipment` в каталог `~/ski.places` (Команду `mv`):

```

kim@kim-VirtualBox:~$ mv equipment ~/ski.places
kim@kim-VirtualBox:~$ ls ~/ski.places
equipment
kim@kim-VirtualBox:~$

```

Мы видим, что файл `equipment` находится в этом каталоге.

2.4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.

```

kim@kim-VirtualBox:~$ mv ~/ski.places/equipment ~/ski.places/equiplist
kim@kim-VirtualBox:~$ ls ~/ski.places
equiplist
kim@kim-VirtualBox:~$

```

2.5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`:


```
kim@kim-VirtualBox:~$ cd
kim@kim-VirtualBox:~$ touch abcl
kim@kim-VirtualBox:~$ cp abcl ~/ski.plases
kim@kim-VirtualBox:~$ ls ~/ski.plases
abcl  equiplist
kim@kim-VirtualBox:~$ mv ~/ski.plases/abcl ~/ski.plases/equiplist2
kim@kim-VirtualBox:~$ ls ~/ski.plases
equiplist  equiplist2
kim@kim-VirtualBox:~$
```

2.6. Создайте каталог с именем equipment в каталоге ~/ski.plases:

```
kim@kim-VirtualBox:~$ cd ~/ski.plases
kim@kim-VirtualBox:~/ski.plases$ mkdir equipment
kim@kim-VirtualBox:~/ski.plases$ ls
equiplist  equiplist2  equipment
kim@kim-VirtualBox:~/ski.plases$
```

Используйте команду `cd ~/ski.plases`, чтобы перейти в каталог `~/ski.plases` и создать файловое `equipment` с помощью команды `mkdir` и посмотреть каталог с помощью команды `ls`.

2.7. Переместите файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment`:

```
kim@kim-VirtualBox:~/ski.plases$ mv equiplist equiplist2 equipment
kim@kim-VirtualBox:~/ski.plases$ ls equiment
ls: cannot access 'equiment': No such file or directory
kim@kim-VirtualBox:~/ski.plases$ ls equipment
equiplist  equiplist2
kim@kim-VirtualBox:~/ski.plases$
```

2.8. Создайте и переместите каталог `~/newdir` в каталог `~/ski.plases` и назовите его `plan`:

```
kim@kim-VirtualBox:~$ mkdir newdir
mkdir: cannot create directory 'newdir': File exists
kim@kim-VirtualBox:~$ mv ~/newdir ~/ski.plases
kim@kim-VirtualBox:~$ ls ~/ski.plases
equipment  newdir
kim@kim-VirtualBox:~$ mv ~/ski.plases/newdir ~/ski.plases/plans
kim@kim-VirtualBox:~$ ls ~/ski.plases
equipment  plans
kim@kim-VirtualBox:~$
```

3. Определите опции команды `chmod`:

3.1. `drwxr--r--` ... `australia`:

```

kim@kim-VirtualBox:~$ cd
kim@kim-VirtualBox:~$ mkdir australia
kim@kim-VirtualBox:~$ ls -al australia
total 8
drwxrwxr-x  2 kim kim 4096 мая 11 00:14 .
drwxr-xr-x 28 kim kim 4096 мая 11 00:14 ..
kim@kim-VirtualBox:~$ chmod u+r+w+x australia
kim@kim-VirtualBox:~$ chmod g+r-w-x australia
kim@kim-VirtualBox:~$ chmod o+r-w-x australia
kim@kim-VirtualBox:~$ ls -al australia
total 8
drwxr--r--  2 kim kim 4096 мая 11 00:14 .
drwxr-xr-x 28 kim kim 4096 мая 11 00:14 ..
kim@kim-VirtualBox:~$

```

Сначала мы делаем доступ к файлу australia (с помощью команды `mkdir`), затем проверяем подробную информацию о файле (с помощью команды `ls -al`), затем устанавливаем новый доступ с помощью команды `chmod` со следующими комбинациями `u+r+w+x` `g+r-w-x` and `o+r-w-x`, а затем мы снова проверяем свойства файла (`ls -al`), и он точно такой же, как и требовалось.

3.2. drwx--x--x ... play

```

kim@kim-VirtualBox:~$ mkdir play
kim@kim-VirtualBox:~$ ls -al play
total 8
drwxrwxr-x  2 kim kim 4096 мая 11 00:17 .
drwxr-xr-x 29 kim kim 4096 мая 11 00:17 ..
kim@kim-VirtualBox:~$ chmod u+r+w+x play
kim@kim-VirtualBox:~$ chmod g-r-w+x play
kim@kim-VirtualBox:~$ chmod o-r-w+x play
kim@kim-VirtualBox:~$ ls -al play
total 8
drwx--x--x  2 kim kim 4096 мая 11 00:17 .
drwxr-xr-x 29 kim kim 4096 мая 11 00:17 ..
kim@kim-VirtualBox:~$

```

Сначала мы делаем доступ к файлу play (с помощью команды `mkdir`), затем проверяем подробную информацию о файле (с помощью команды `ls -al`), затем устанавливаем новый доступ с помощью команды `chmod` со следующими комбинациями `u+r+w+x` `g-r-w+x` and `o-r-w+x`, а затем мы снова проверяем свойства файла (`ls -al`), и он точно такой же, как и требовалось.

3.3. -r-xr--r-- ... my_os

```
kim@kim-VirtualBox:~$ touch my_os
kim@kim-VirtualBox:~$ ls -al my_os
-rw-rw-r-- 1 kim kim 0 мая 11 00:20 my_os
kim@kim-VirtualBox:~$ chmod u+r-w+x my_os
kim@kim-VirtualBox:~$ chmod g+r-w-x my_os
kim@kim-VirtualBox:~$ chmod o+r-w-x my_os
kim@kim-VirtualBox:~$ ls -al my_os
-r-xr--r-- 1 kim kim 0 мая 11 00:20 my_os
kim@kim-VirtualBox:~$
```

для этого сначала мы создаем файл `my_os` (с помощью команды `touch`), затем проверяем подробную информацию о файле (с помощью команды `ls -al`), затем устанавливаем новый доступ с помощью команды `chmod` со следующими комбинациями `u+r-w+x` `g+r-w-x` and `o+r-w-x`, а затем мы снова проверяем свойства файла (`ls -al`), и он точно такой же, как и требовалось.

3.4. -rw-rw-r-- ... feathers

```
kim@kim-VirtualBox:~$ touch feathers
kim@kim-VirtualBox:~$ ls -l feathers
-rw-rw-r-- 1 kim kim 0 мая 11 00:22 feathers
kim@kim-VirtualBox:~$
```

При этом мы видим, что свойства файла уже совпадают с требуемыми.

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

4.1. Просмотрите содержимое файла `/etc/passwd` (с помощью командной `cat`)

```
kim@kim-VirtualBox:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
```

4.2.Скопируйте файл ~/feathers в файл ~/file.old. (с помощью команд `cp` и `ls`)

```
kim@kim-VirtualBox:~$ cp ~/feathers ~/file.old
kim@kim-VirtualBox:~$ ls ~/file.old
/home/kim/file.old
kim@kim-VirtualBox:~$
```

4.3.Переместите файл ~/file.old в каталог ~/play. (с помощью команд `mv`)

```
kim@kim-VirtualBox:~$ mv ~/file.old ~/play
kim@kim-VirtualBox:~$ ls ~/play
file.old
kim@kim-VirtualBox:~$
```

4.4. Скопируйте каталог ~/play в каталог ~/fun.

```
kim@kim-VirtualBox:~$ cp -r ~/play ~/fun
kim@kim-VirtualBox:~$ ls ~/fun
file.old
```

4.5. Переместите каталог ~/fun в каталог ~/play(с помощью команд `mv`) и назовите его games

```
kim@kim-VirtualBox:~$ mv ~/fun ~/play
kim@kim-VirtualBox:~$ ls ~/play
file.old  fun
kim@kim-VirtualBox:~$ mv ~/play/fun ~/play/game
kim@kim-VirtualBox:~$ ls ~/play
file.old  game
kim@kim-VirtualBox:~$
```

4.6. Лишите владельца файла ~/feathers права на чтение:

```
kim@kim-VirtualBox:~$ chmod u-r ~/feathers
kim@kim-VirtualBox:~$
```

4.7. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?

```
kim@kim-VirtualBox:~$ cat ~/feathers
kim@kim-VirtualBox:~$
```

Там ничего не происходит, когда мы используем команду cat для просмотра файла ~/feathers .

4.8. Что произойдёт, если вы попытаетесь скопировать файл ~/feathers?

```
kim@kim-VirtualBox:~$ cp ~/feathers
cp: missing destination file operand after '/home/kim/feathers'
Try 'cp --help' for more information.
kim@kim-VirtualBox:~$
```

это показывает, что в нем отсутствует файл назначения, даже если мы все сделали правильно в предыдущем упражнении.

4.9. Дайте владельцу файла ~/feathers право на чтение с помощью команд chmod

```
kim@kim-VirtualBox:~$ chmod u-r ~/play
kim@kim-VirtualBox:~$ chmod u-x ~/play
```

4.10. Лишите владельца каталога ~/play права на выполнение.

```
kim@kim-VirtualBox:~$ chmod u-x ~/play
kim@kim-VirtualBox:~$ chmod u+x ~/play
kim@kim-VirtualBox:~$
```

4.11. Перейдите в каталог ~/play

```
kim@kim-VirtualBox:~$ cd ~/play
bash: cd: /home/kim/play: Permission denied
kim@kim-VirtualBox:~$
```

как мы видим, в разрешении отказано.

4.12. Дайте владельцу каталога ~/play право на выполнение.

```
kim@kim-VirtualBox:~$ chmod u+x ~/play
kim@kim-VirtualBox:~$ cd ~/play
kim@kim-VirtualBox:~/play$
```

сейчас он успешно, как и мы, свои права выполнение.

5. Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приводя примеры:

```
kim@kim-VirtualBox:~/play$ cd
kim@kim-VirtualBox:~$ man mount
kim@kim-VirtualBox:~$ man fsck
kim@kim-VirtualBox:~$ man mkfs
kim@kim-VirtualBox:~$ man kill
kim@kim-VirtualBox:~$
```

- `mount` : Команда Mount присоединяет файловую систему внешнего устройства к файловой системе системы. Он сообщает операционной системе, что файловая система готова к использованию, и связывает ее с определенной точкой в иерархии системы. Монтирование сделает файлы, каталоги и устройства доступными для пользователей.

NAME

mount - mount a filesystem

SYNOPSIS

mount [-l|-h|-V]

mount -a [-fFnrsvw] [-t fstype] [-O optlist]

mount [-fFnrsvw] [-o options] device|dir

mount [-fFnrsvw] [-t fstype] [-o options] device dir

DESCRIPTION

All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at `/`. These files can be spread out over several devices. The `mount` command serves to attach the filesystem found on some device to the big file tree. Conversely, the `umount(8)` command will detach it again. The filesystem is used to control how data is stored on the device or provided in a virtual way by network or another services.

The standard form of the `mount` command is:

`mount -t type device dir`

This tells the kernel to attach the filesystem found on device (which is of type type) at the directory dir. The option `-t type` is optional. The `mount` command is usually able to detect a filesystem. The root permissions are necessary to mount a filesystem by default. See section "Non-superuser mounts" below for more details. The previous contents (if any) and owner and mode of dir become invisible, and as long as this filesystem remains mounted, the pathname dir refers to the root of the filesystem on device.

If only the directory or the device is given, for example:

`mount /dir`

- `fsck` : (проверка файловой системы)-это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивные исправления в одной или нескольких файловых системах Linux. Вы можете использовать команду `fsck` для восстановления поврежденных файловых систем в ситуациях, когда система не загружается или раздел не может быть смонтирован.


```

FSCK(8)                                     System Administration                                     FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-options]

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystems can be a device name (e.g. /dev/hdc1, /dev/sdb2), a mount point (e.g. /, /usr, /home), or an filesystem label or UUID specifier (e.g. UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the fsck program will try to handle filesystems on different physical disk drives in parallel to reduce the total amount of time needed to check all of them.

    If no filesystems are specified on the command line, and the -A option is not specified, fsck will default to checking filesystems in /etc/fstab serially. This is equivalent to the -As options.

    The exit code returned by fsck is the sum of the following conditions:

        0      No errors
        1      Filesystem errors corrected
        2      System should be rebooted
        4      Filesystem errors left uncorrected
        8      Operational error
        16     Usage or syntax error
        32     Checking canceled by user request
        128    Shared-library error

    The exit code returned when multiple filesystems are checked is the bit-wise OR of the exit codes for each filesystem that is checked.

    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 the PATH environment variable. If the PATH is undefined then fallback to "/sbin".

    Please see the filesystem-specific checker manual pages for further details.

```

- **mkfs** : Команда **mkfs** расшифровывается как “создать файловую систему” и используется для создания файловой системы (то есть системы для организации иерархии каталогов, подкаталогов и файлов) на отформатированном устройстве хранения, как правило, разделе на жестком диске (HDD) или это также может быть USB-накопитель и т. Д.

```

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 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 via your PATH environment setting only. Please see the filesystem-specific builder manual pages for further details.

OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2) is used.

    fs-options
        Filesystem-specific options to be passed to the real filesystem builder.

    -V, --verbose
        Produce verbose output, including all filesystem-specific commands that are executed. Specifying this option more than once inhibits 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 text and exit.

```


- **kill** : Команда kill посылает сигнал (по умолчанию сигнал SIGTERM) запущенному процессу. Это действие по умолчанию обычно останавливает процессы.

```
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.

    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is explained in signal(7) manual page.

    -l, --list [signal]
        List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.

    -L, --table
        List signal names in a nice table.

NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may need to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -9 -1
        Kill all processes you can kill.

    kill -l 11
```

Вывод:

Я познакомилась с файловой системой Linux, ее структурой, именами и содержимым каталогов. Приобретение практических навыков использования команд для работы с файлами и каталогами, управления процессами(и работой), мониторинга использования дисков и обслуживания файловой системы.