

# Отчёт по лабораторной работе №5

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

Смирнов-Мальцев Егор Дмитриевич

## 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.

# Теоретическое введение

Для создания текстового файла можно использовать команду `touch`. Формат команды:

```
$ touch имя-файла
```

Для просмотра файлов небольшого размера можно использовать команду `cat`. Формат команды:

```
$ cat имя-файла
```

Для просмотра файлов постранично удобнее использовать команду `less`. Формат команды:

```
$ less имя-файла
```

Следующие клавиши используются для управления процессом просмотра: — `Space` — переход к следующей странице, — `ENTER` — сдвиг вперёд на одну строку, — `b` — возврат на предыдущую страницу, — `h` — обращение за подсказкой, — `q` — выход из режима просмотра файла.

Команда `head` выводит по умолчанию первые 10 строк файла. Формат команды:

```
$ head [-n] имя-файла,
```

где `n` — количество выводимых строк.

Команда `tail` выводит по умолчанию 10 последних строк файла. Формат команды:

```
$ tail [-n] имя-файла,
```

где  $n$  — количество выводимых строк.

Команда `cp` используется для копирования файлов и каталогов. Формат команды:

```
$ cp [-опции] исходный_файл целевой_файл
```

Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов. Формат команды `mv`:

```
$ mv [-опции] старый_файл новый_файл
```

Каждый файл или каталог 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).

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

о (others) все остальные

В работе с правами доступа можно использовать их цифровую запись (восьмеричное значение) вместо символьной.

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/binfmt_misc type binfmt_misc
```

```
(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
```

# Выполнение лабораторной работы

1. Выполнил примеры из указаний к работе.

1.1. Создал файл abc1 и скопировал его в файлы april и may.(рис. 1)

```
[edsmirnovmaljce@edsmirnovmaljce EgorSmM.github.io]$ cd  
[edsmirnovmaljce@edsmirnovmaljce ~]$ touch abc1  
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp abc1 april  
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp abc1 may
```

рис 1.

1.2. Скопировал april и may в каталог monthly.(рис. 2)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mkdir monthly  
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp april may monthly
```

рис. 2.

1.3. Скопировал may в june(рис. 3)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp monthly/may monthly/june  
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls monthly  
april june may
```

рис 3.

1.4. Скопировал каталог monthly в monthly.00.(рис. 4)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mkdir monthly.00  
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp -r monthly monthly.00
```

рис 4.

1.5. Скопировал каталог monthly.00 в /tmp.(рис. 5)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp -r monthly.00 /tmp
```

рис 5.



1.6. Переименовал файл april в july.(рис. 6)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv april july
```

рис 6.

1.7. Переместил файл july в каталог monthly.00.(рис. 7)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv july monthly.00
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls monthly.00
july  monthly
```

рис 7.

1.8. Переименовал каталог monthly.00 в monthly.01.(рис. 8)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv monthly.00 monthly.01
```

рис 8.

1.9. Переместил каталог monthly.01 в каталог reports.(рис. 9)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mkdir reports
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv monthly.01 reports
```

рис 9.

1.10. Переименовал каталог reports/monthly.01 в reports/monthly.(рис. 10)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv reports/monthly.01 reports/monthly
```

рис 10.

1.11. Создал файл ~/may с правом выполнения для владельца.(рис. 11)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ touch may
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls -l may
-rw-rw-r--. 1 edsmirnovmaljce edsmirnovmaljce 0 мая  2 16:17 may
[edsmirnovmaljce@edsmirnovmaljce ~]$ chmod u+x may
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls -l may
-rwxrw-r--. 1 edsmirnovmaljce edsmirnovmaljce 0 мая  2 16:17 may
```

рис 11.

1.12. Лишил владельца файла ~/may права на выполнение.(рис. 12)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls -l may
-rw-rw-r--. 1 edsmirnovmaljce edsmirnovmaljce 0 мая  2 16:17 may
```

рис 12.

1.13. Создал каталог monthly с запретом на чтение для членов группы и всех остальных пользователей(рис. 13)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ chmod go-r monthly
```

рис 13.

1.14. Создал файл ~/abc1 с правом записи для членов группы.(рис. 14)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ chmod g+w abc1
[edsmirnovmaljce@edsmirnovmaljce ~]$ ls -l abc1
-rw-rw-r--. 1 edsmirnovmaljce edsmirnovmaljce 0 мая  2 15:41 abc1
```

рис 14.

1.15. Проверил целостность файловой системы.(рис. 15)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ fsck /dev/sda1
fsck из util-linux 2.37.4
e2fsck 1.46.3 (27-Jul-2021)
/dev/sda1 is mounted.
```

рис 15.

2. Выполнил следующие действия:

2.1.Скопировал файл /usr/include/sys/io.h в домашний каталог и назвал его equipment(рис. 16)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp /usr/include/sys/io.h equipment
```

рис 16.

2.2. В домашнем каталоге создал директорию ~/ski.plases.(рис. 17)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mkdir ski.plases
```

рис 17.

2.3. Переместил файл equipment в каталог ~/ski.plases.(рис. 18)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv equipment ski.plases
```

рис 18.

2.4. Переименовал файл ~/ski.plases/equipment в ~/ski.plases/equiplist.(рис. 19)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv ski.plases/equipment ski.plases/equiplist
```

рис 19.

2.5. Создал в домашнем каталоге файл abc1 и скопировал его в каталог ~/ski.plases, назовите его equiplist2.(рис. 20)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ touch abc1  
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp abc1 ski.plases/equiplist2
```

рис 20.

2.6. Создал каталог с именем equipment в каталоге ~/ski.plases.(рис. 21)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mkdir ski.plases/equipment
```

рис 21.

2.7. Переместил файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment.(рис. 22)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv ski.plases/equiplist ski.plases/equiplist2 ski.plases/equipment
```

рис 22.

2.8. Создал и переместил каталог ~/newdir в каталог ~/ski.plases и назвал его plans.(рис. 23)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv newdir ski.plases  
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv ski.plases/newdir ski.plases/plans
```

рис 23.

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

- drwxr-r- ... australia
- drwx-x-x ... play
- -r-xr-r- ... my\_os
- -rw-rw-r- ... feathers (рис. 24)

```
[edsmirnovmaljce@edsmirnovmaljce 3]$ mkdir australia play
[edsmirnovmaljce@edsmirnovmaljce 3]$ touch my_os feathers
[edsmirnovmaljce@edsmirnovmaljce 3]$ chmod 744 australia
[edsmirnovmaljce@edsmirnovmaljce 3]$ chmod 711 play
[edsmirnovmaljce@edsmirnovmaljce 3]$ chmod 544 my_os
[edsmirnovmaljce@edsmirnovmaljce 3]$ chmod 664 feathers
[edsmirnovmaljce@edsmirnovmaljce 3]$ ls -l
итого 0
drwxr--r--. 1 edsmirnovmaljce edsmirnovmaljce 0 мая  2 17:31 australia
-rw-rw-r--. 1 edsmirnovmaljce edsmirnovmaljce 0 мая  2 17:31 feathers
-r-xr--r--. 1 edsmirnovmaljce edsmirnovmaljce 0 мая  2 17:31 my_os
drwx--x--x. 1 edsmirnovmaljce edsmirnovmaljce 0 мая  2 17:31 play
```

рис 24.

4. Проделал следующие упражнения:

4.1. Просмотрел содержимое файла /etc/passwd.(рис. 25)

```
[edsmirnovmaljce@edsmirnovmaljce 3]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:usr/sbin/nologin
systemd-oom:x:999:999:systemd Userspace OOM Killer:/:usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:usr/sbin/nologin
systemd-timesync:x:998:998:systemd Time Synchronization:/:usr/sbin/nologin
systemd-coredump:x:997:997:systemd Core Dumper:/:usr/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
polkitd:x:996:996:User for polkitd:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
unbound:x:995:994:Unbound DNS resolver:/etc/unbound:/sbin/nologin
dnsmasq:x:994:993:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/sbin/nologin
nm-openconnect:x:993:991:NetworkManager user for OpenConnect:/:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
gluster:x:992:990:GlusterFS daemons:/run/gluster:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
pipewire:x:991:989:PipeWire System Daemon:/var/run/pipewire:/sbin/nologin
geoclue:x:990:988:User for geoclue:/var/lib/geoclue:/sbin/nologin
chrony:x:989:986:/:/var/lib/chrony:/sbin/nologin
saslauthd:x:988:76:Saslauthd user:/run/saslauthd:/sbin/nologin
radvd:x:75:75:radvd user:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
openvpn:x:987:984:OpenVPN:/etc/openvpn:/sbin/nologin
nm-openvpn:x:986:983:Default user for running openvpn spawned by NetworkManager:/:/sbin/nologin
colord:x:985:982:User for colord:/var/lib/colord:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
abrt:x:173:173:/:/etc/abrt:/sbin/nologin
flatpak:x:984:981:User for flatpak system helper:/:/sbin/nologin
gdm:x:42:42:/:/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:983:980:/:run/gnome-initial-setup:/sbin/nologin
vboxadd:x:982:1:/:/var/run/vboxadd:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
```

рис 25.

4.2. Скопировал файл ~/feathers в файл ~/file.old.(рис. 26)

```
[edsmirnovmaljce@edsmirnovmaljce 3]$ cp ~/feathers ~/file.old
```

рис 26.

4.3. Переместил файл ~/file.old в каталог ~/play.(рис. 27)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv file.old play
```

рис 27.

4.4. Скопировал каталог ~/play в каталог ~/fun.(рис. 28)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp -r play fun
```

рис 28.

4.5. Переместил каталог ~/fun в каталог ~/play и назвал его games.(рис. 29)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv fun play
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv fun games
mv: не удалось выполнить stat для 'fun': Нет такого файла или каталога
[edsmirnovmaljce@edsmirnovmaljce ~]$ mv play/fun play/games
```

рис 29.

4.6. Лишил владельца файла ~/feathers права на чтение.(рис. 31)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ chmod u-r feathers
```

рис 31.

4.7. Попытался просмотреть файл ~/feathers командой cat, но у меня не получилось, поскольку нет на это прав.(рис. 32)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ cat feathers
cat: feathers: Отказано в доступе
```

рис 32.

4.8. Попытался скопировать файл ~/feathers, но у меня не получилось, поскольку нет на это прав.(рис. 33)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ cp feathers feathers1
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
```

рис 33.

4.9. Дал владельцу файла ~/feathers право на чтение.(рис. 34)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ chmod u+r feathers
```

рис 34.

4.10. Лишил владельца каталога ~/play права на выполнение.(рис. 35)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ chmod u-x play
```

рис 35.

4.11. Попытался перейти в каталог `~/play`, но у меня не получилось, поскольку нет на это прав.(рис. 36)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ cd play
bash: cd: play: Отказано в доступе
```

рис 36.

4.12. Дал владельцу каталога `~/play` право на выполнение.(рис. 37)

```
[edsmirnovmaljce@edsmirnovmaljce ~]$ chmod u+x play
```

рис 37.

5. Прочитал man по следующим командам:

5.1. Все файлы, доступные в системе Unix, расположены в одном большом дереве, файловая иерархия, с корнем в `/`. Эти файлы могут быть распределены по нескольким устройствам. Команда `mount` служит для подключения файловой системы, найденной на каком-то устройстве к большому файловому дереву. И наоборот, команда `umount` отсоединит его снова. Файловая система используется для управления тем, как данные хранятся на устройстве или предоставляется виртуальным способом по сети или другим услуги.(рис. 38)

```
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 other
services.
```

рис 38.

5.2. `fsck` используется для проверки и при необходимости, восстановления одного или нескольких файловых систем. Файловой системой может быть имя устройства (например, `/dev/hdc1`, `/dev/sdb2`), точка монтирования (например, `/`, `/usr`, `/home`) или файловая система метка или спецификатор UUID (например, `UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd` или `LABEL=root`). Как правило, программа `fsck` попытается обработать файловые системы на другом физическом диске параллельно,

чтобы сократить общее время, необходимое для проверки. Если в командной строке не указаны файловые системы, и параметр -A, fsck по умолчанию будет проверять файловые системы в /etc/fstab последовательно. Это эквивалентно параметрам -As.(рис. 39)

```
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. This is equivalent to the -As options.
```

рис 39.

5.3. mkfs используется для создания файловой системы Linux на устройстве, обычно на жестком разделе диска. Аргументом является либо имя устройства (например, /dev/hda1, /dev/sdb2), либо обычный файл, который должен содержать файловая система. Аргумент size - это количество блоков, которые будут использоваться для файловой системы. Статус выхода, возвращаемый mkfs, равен 0 при успешном завершении и 1 при сбое.(рис. 40)

```
DESCRIPTION
  This mkfs frontend is deprecated in favour of filesystem specific
  mkfs.<type> utils.

  mkfs is used to build a Linux filesystem on a device, usually a hard
  disk partition. The device argument is either the device name (e.g.,
/dev/hda1, /dev/sdb2), or a regular file that shall contain the
  filesystem. The size argument is the number of blocks to be used for
  the filesystem.

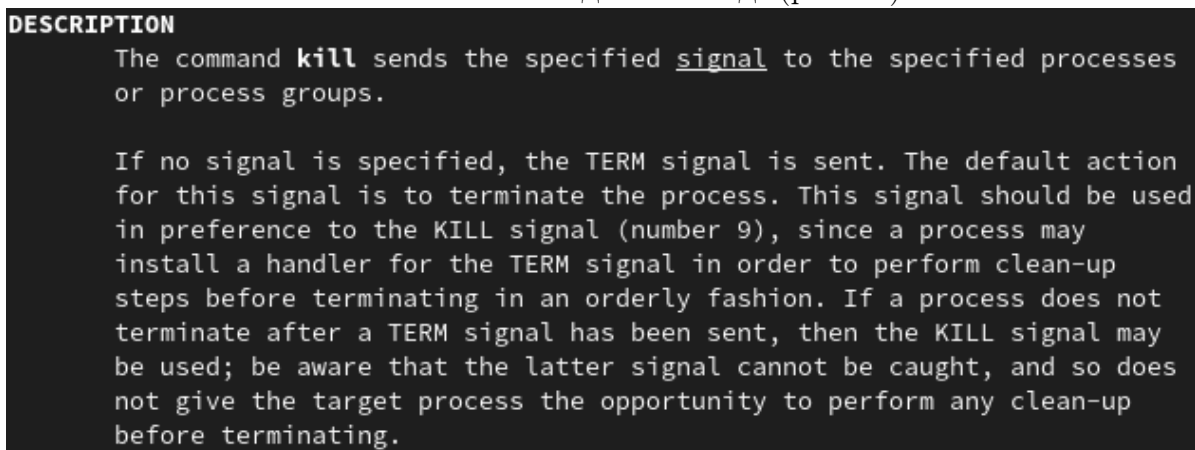
  The exit status returned by mkfs is 0 on success and 1 on failure.

  In actuality, mkfs is simply a front-end for the various filesystem
  builders (mkfs.fstype) available under Linux. The filesystem-specific
  builder is searched for via your PATH environment setting only. Please
  see the filesystem-specific builder manual pages for further details.
```



рис 40.

5.4. Команда `kill` отправляет указанный сигнал указанным процессам или группе процессов. Если сигнал не указан, отправляется сигнал `TERM`. Действие по умолчанию для этого сигнала является завершение процесса. Этот сигнал следует использовать в предпочтении к сигналу `KILL` (номер 9), поскольку процесс может установить обработчик для терминального сигнала, чтобы выполнить очистку шагов перед завершением упорядоченным образом. Если процесс не завершается после того, как был отправлен сигнал `TERM`, тогда сигнал `KILL` может быть использованным; имейте в виду, что последний сигнал не может быть перехвачен, как и не дать целевому процессу возможности выполнить какую-либо очистку перед завершением. Большинство современных оболочек имеют встроенную команду `kill`, с использованием аналогично описанной здесь команде.(рис. 41)

A screenshot of a terminal window with a dark background and light-colored text. The text describes the `kill` command, its default behavior, and the difference between `TERM` and `KILL` signals.

**DESCRIPTION**

The command `kill` sends the specified signal to the specified processes or process groups.

If no signal is specified, the `TERM` signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the `KILL` signal (number 9), since a process may install a handler for the `TERM` signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a `TERM` signal has been sent, then the `KILL` signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

рис 41.

# Контрольные вопросы

## 1. Установленные файловые системы:

1.1. `devtmpfs`, `tmpfs` - виртуальные файловые системы, которые напрямую обращаются к памяти.

1.2. `sda1`, `sda2` - жесткие диски.

2. `/` - КОРЕНЬ - Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Здесь нет дисков или чего-то подобного, как в Windows. Вместо этого, адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога.

Только пользователь `root` имеет право читать и изменять файлы в этом каталоге. Обратите внимание, что у пользователя `root` домашний каталог `/root`, но не сам `/`.

`/BIN` - (BINARIES) БИНАРНЫЕ ФАЙЛЫ ПОЛЬЗОВАТЕЛЯ Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления. Одним словом, те утилиты, которые могут использоваться пока еще не подключен каталог `/usr/`. Это такие общие команды, как `cat`, `ls`, `tail`, `ps` и т.д.

`/SBIN` - (SYSTEM BINARIES) СИСТЕМНЫЕ ИСПОЛНЯЕМЫЕ ФАЙЛЫ Так же как и `/bin`, содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки, когда не примонтирован каталог `/usr`. Но здесь находятся программы, которые можно выполнять только с правами суперпользователя. Это разные

утилиты для обслуживания системы. Например, iptables, reboot, fdisk, ifconfig, swapon и т.д.

**/ETC - (ETCETERA) КОНФИГУРАЦИОННЫЕ ФАЙЛЫ** В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, в системе инициализации Init Scripts, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ. Структура каталогов linux в этой папке может быть немного запутанной, но предназначение всех их - настройка и конфигурация.

**/DEV - (DEVICES) ФАЙЛЫ УСТРОЙСТВ** В Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры - это просто файлы в каталоге /dev/. Этот каталог содержит не совсем обычную файловую систему. Структура файловой системы Linux и содержащиеся в папке /dev файлы инициализируются при загрузке системы, сервисом udev. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов. Это такие устройства, как: /dev/sda, /dev/sr0, /dev/tty1, /dev/usbmon0 и т.д.

**/PROC - (PROCESS) ИНФОРМАЦИЯ О ПРОЦЕССАХ** Это тоже необычная файловая система, а подсистема, динамически создаваемая ядром. Здесь содержится вся информация о запущенных процессах в реальном времени. По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его Pid, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов, например, /proc/cpuinfo, /proc/meminfo или /proc/uptime. Кроме файлов в этом каталоге есть большая структура папок linux, из которых можно узнать достаточно много информации о системе.

**/VAR (VARIABLE) - ПЕРЕМЕННЫЕ ФАЙЛЫ** Название каталога /var говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кешы, базы данных и так далее. Дальше рассмотрим назначение каталогов

Linux в папке `/var/`.

3. Чтобы содержимое некоторой файловой системы было доступно операционной системе надо выполнить следующую команду:

```
$ mount <файловая система>
```

4. Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийной остановки является основной причиной повреждения файловой системы. Исправить повреждения можно следующей командой:

```
$ fsck
```

5. Создать файловую систему можно с помощью следующей команды:

```
$ mkfs <файловая система>
```

6. `Cat` - выводит содержимое файла на стандартное устройство вывода
7. `Cp` – копирует директорию или файлы.
8. `mv` - переименовывает или перемещает файл или директорию
9. Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

## Выводы

- Я ознакомился с файловой системой Linux и изучил основные команды для работы с файлами.