

Файловая система Linux

Бровкин Александр НБИбд-01-21¹

28 апреля, 2022, Москва, Россия

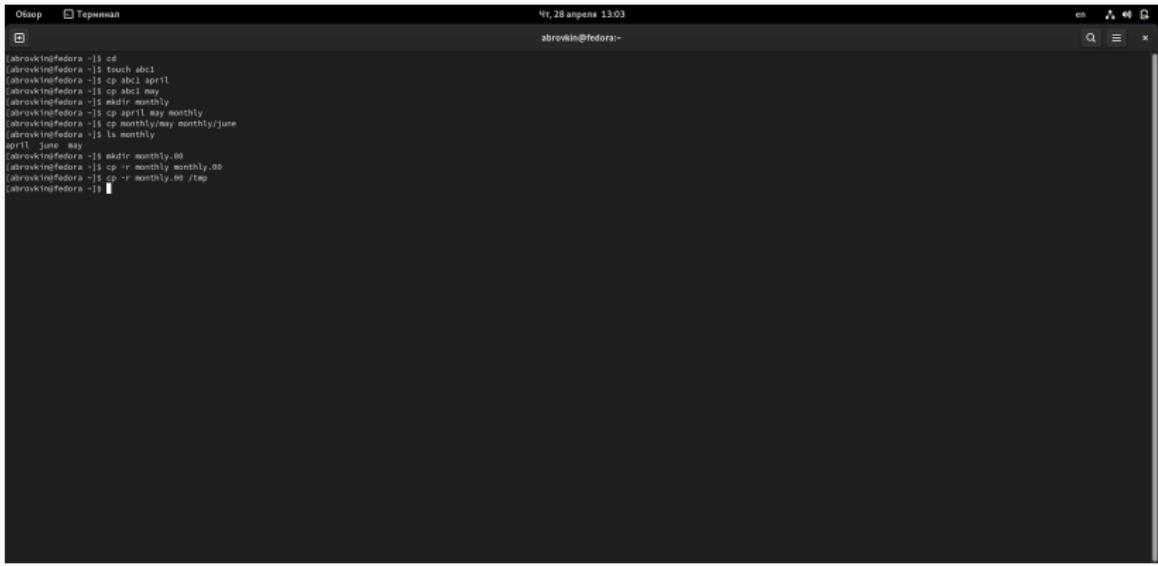
¹Российский Университет Дружбы Народов

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.

Выполнение лабораторной работы

1. Выполнил все примеры, приведённые в первой части описания лабораторной работы. Скопировал файл `~/abc1` в файл `april` и в файл `may`. Скопировал файлы `april` и `may` в каталог `monthly`. Скопировал файл `monthly/may` в файл с именем `june`. Скопировал каталог `monthly` в каталог `monthly.00`. Скопировал каталог `monthly.00` в каталог `/tmp`



Обзор Терминал

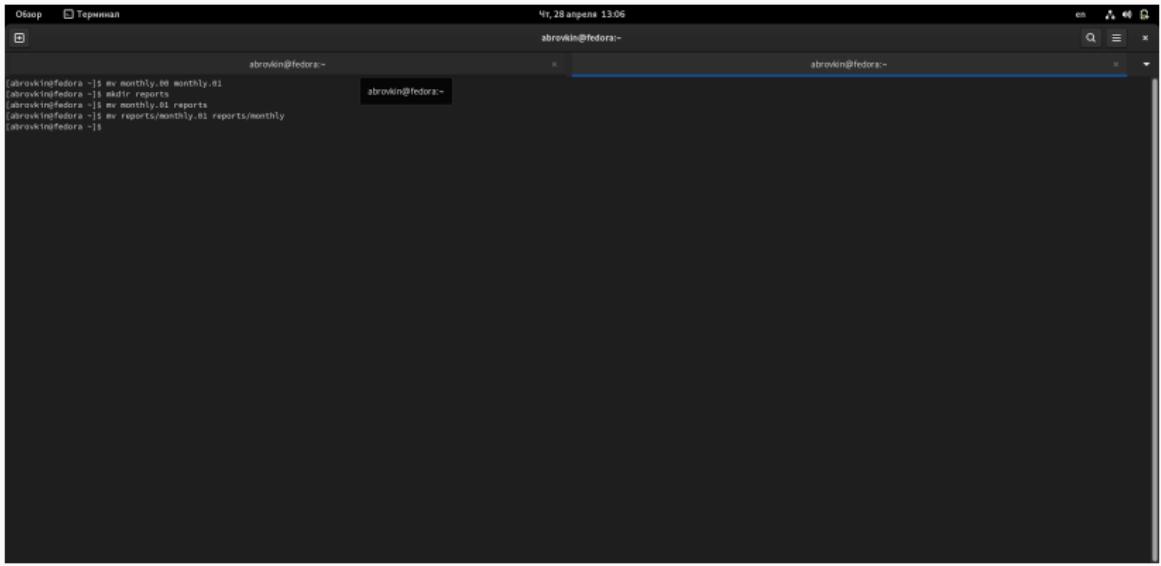
Чт, 28 апреля 13:03

abrovkin@fedora:~

```
abrovkin@fedora ~$ cd
abrovkin@fedora ~$ touch abc1
abrovkin@fedora ~$ cp abc1 april
abrovkin@fedora ~$ cp abc1 may
abrovkin@fedora ~$ cp abc1 june
abrovkin@fedora ~$ cp april may monthly
abrovkin@fedora ~$ cp monthly/may monthly/june
abrovkin@fedora ~$ ls monthly
abc1 monthly
abrovkin@fedora ~$ mkdir monthly.00
abrovkin@fedora ~$ cp -r monthly monthly.00
abrovkin@fedora ~$ cp -r monthly.00 /tmp
abrovkin@fedora ~$
```

Figure 1: Выполняю примеры из лабораторной

Изменил название файла april на july в домашнем каталоге.
Переместил файл july в каталог monthly.00. Переименовал
каталог monthly.00 в monthly.01. Переместил каталог
monthly.01 в каталог reports. Переименовал каталог
reports/monthly.01 в reports/monthly



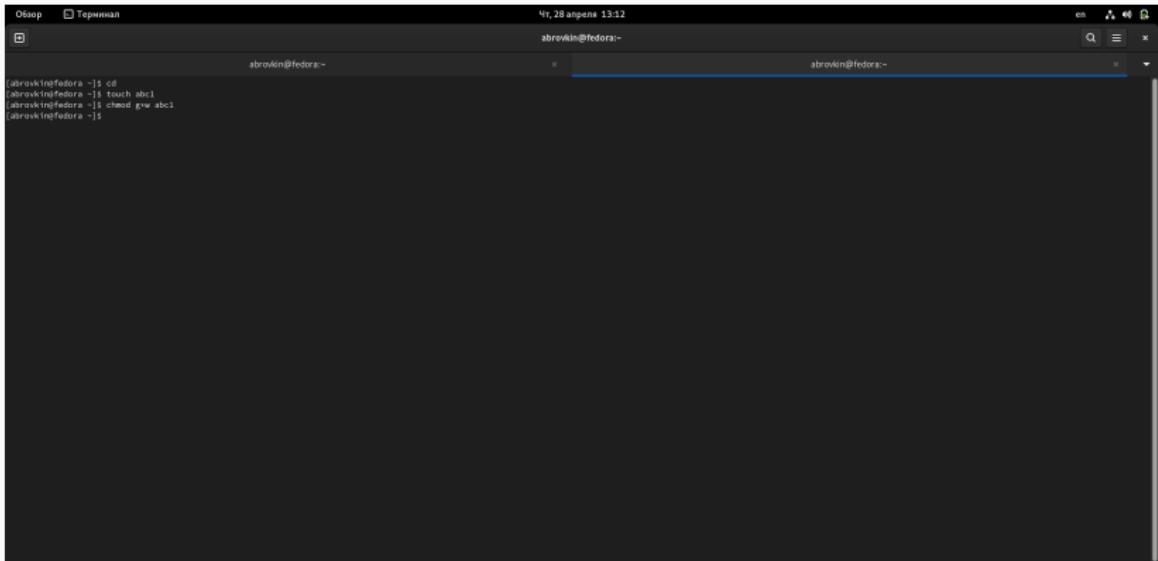
The screenshot shows a terminal window with three tabs open. The tabs are labeled 'Обзор' (Overview), 'Терминал' (Terminal), and another tab whose title is partially visible. The terminal tab displays a command-line session:

```
aurokin@fedora:~$ ls monthly.00 monthly.01
aurokin@fedora:~$ mkdir reports
aurokin@fedora:~$ mv monthly.01 reports
aurokin@fedora:~$ mv reports/monthly.01 reports/monthly
aurokin@fedora:~$
```

Figure 2: Продолжаю выполнять примеры

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

Figure 3: Продолжаю выполнять примеры

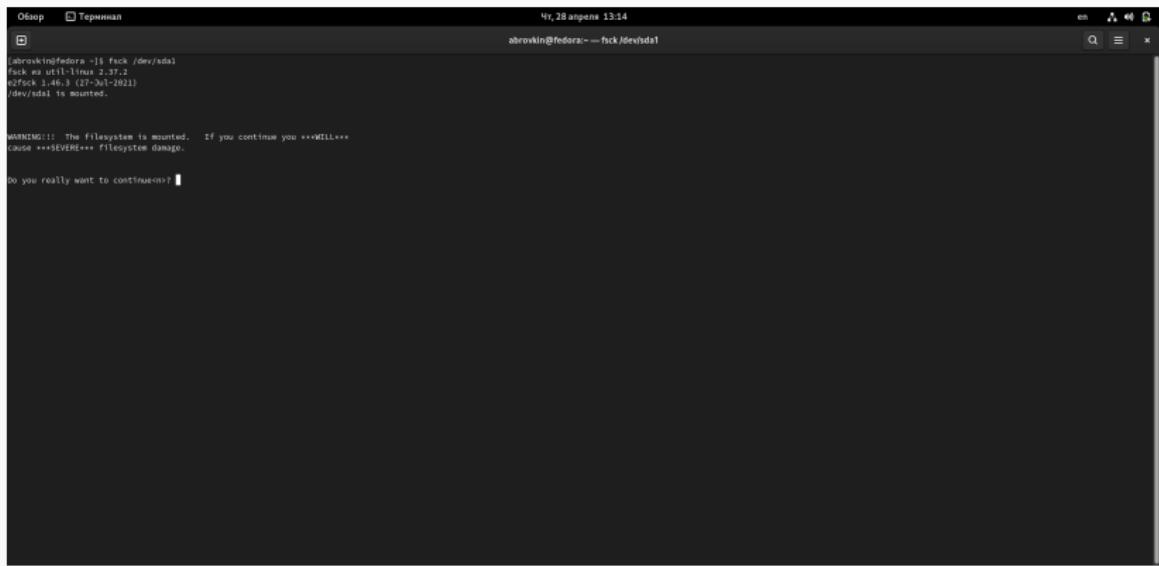


The screenshot shows a terminal window with three tabs. The leftmost tab is titled 'Обзор' (Overview) and the rightmost is titled 'Терминал' (Terminal). The central tab is active and displays the command-line session:

```
Чт, 28 апреля 13:12
abrovkin@fedora:-
abrovkin@fedora:~$ cd
abrovkin@fedora:~$ touch abc1
abrovkin@fedora:~$ chmod g+w abc1
abrovkin@fedora:~$
```

Figure 4: Продолжаю выполнять примеры

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



The screenshot shows a terminal window titled "Терминал" (Terminal) with the following content:

```
Чт, 28 апреля 13:14
abrovkin@fedora: ~ -- fsck /dev/sda1
fsck 3.42.1 (2013-07-01)
fsck 1.46.3 (27-Nov-2011)
/dev/sda1 is mounted.

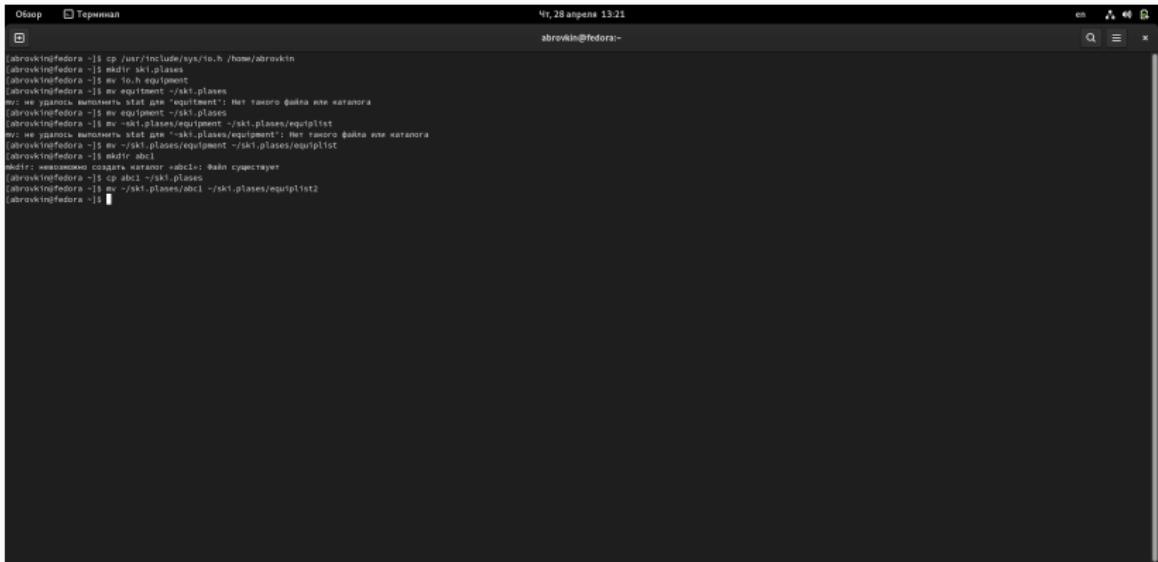
WARNING!!! The filesystem is mounted. If you continue you ***WILL***
cause ***SEVERE*** filesystem damage.

Do you really want to continue on? [
```

Figure 5: fsck

2. Выполнил следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
 - 2.1. Скопировал 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 создал директорию ~/ski.plases.
 - 2.3. Переместил файл equipment в каталог ~/ski.plases командой mv.
 - 2.4. Переименовал файл ~/ski.plases/equipment в ~/ski.plases/equiplist командой mv.
 - 2.5. Создал в домашнем каталоге файл abc1 и скопировал его в каталог ~/ski.plases командой cp, назвал его equiplist2 командой mv.

- 2.6. Создал каталог с именем equipment в каталоге ~/ski.plases командой mkdir.
- 2.7. Переместил файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment командой mv.
- 2.8. Создал и переместил каталог ~/newdir в каталог ~/ski.plases командами mkdir и mv и назвал его plans командой mv.

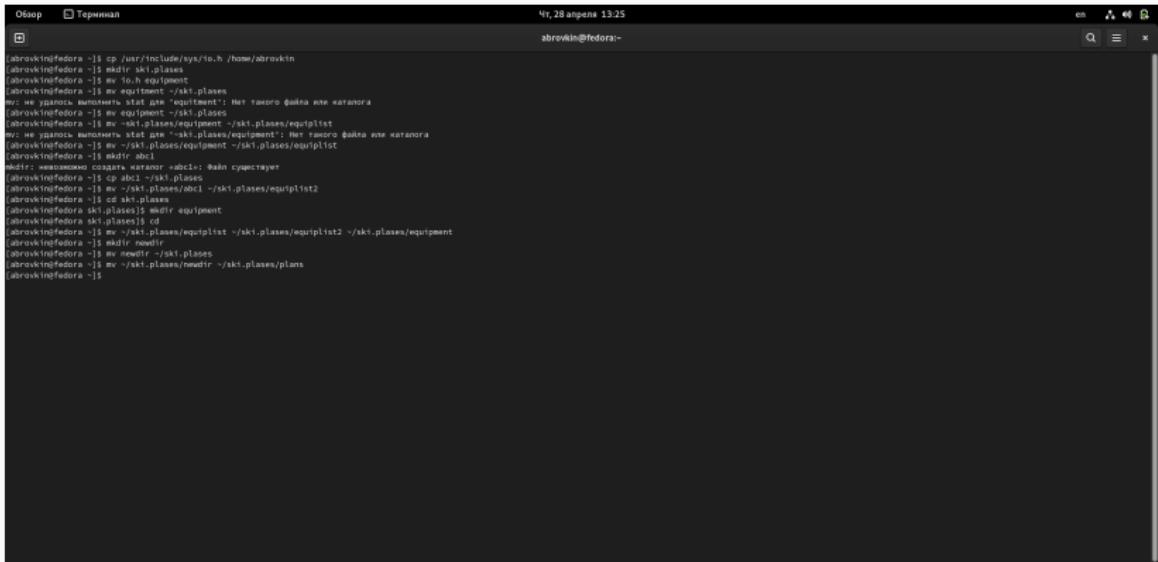


Обзор Терминал

Чт, 28 апреля 13:21
abrovkin@fedora:-

```
abrovkin@fedora: ~$ cp /usr/include/sys/io.h ./home/abrovkin
abrovkin@fedora: ~$ mkdir sk1_plases
abrovkin@fedora: ~$ mv io.h equipment
abrovkin@fedora: ~$ mv equipment ./sk1_plases
abrovkin@fedora: ~$ touch ./sk1_plases/equipment; echo "Нет такого файла или каталога"
abrovkin@fedora: ~$ mv equipment ./sk1_plases
abrovkin@fedora: ~$ mv ./sk1_plases/equipment ./sk1_plases/equiplist
mv: не удалось: equipment: нет такого файла или каталога
abrovkin@fedora: ~$ mv ./sk1_plases/equipment ./sk1_plases/equiplist
abrovkin@fedora: ~$ mkdir abc
mkdir: невозможно создать каталог `abc': файл существует
abrovkin@fedora: ~$ cp abc ./sk1_plases
abrovkin@fedora: ~$ mv ./sk1_plases/abc ./sk1_plases/equiplist2
abrovkin@fedora: ~$
```

Figure 6: Продолжаю выполнять примеры



Обзор Терминал

Чт, 28 апреля 13:25
abrovkin@fedora:~

```
abrovkin@fedora:~$ cp /usr/include/sys/io.h /home/abrovkin
abrovkin@fedora:~$ mkdir ski_plases
abrovkin@fedora:~$ mv io.h equipment
abrovkin@fedora:~$ mv equipment ~/ski_plases
abrovkin@fedora:~/ski_plases$ touch equipment
abrovkin@fedora:~/ski_plases$ touch equipment: Нет такого файла или каталога
abrovkin@fedora:~/ski_plases$ mv equipment ~/ski_plases
abrovkin@fedora:~/ski_plases$ mv ~/ski_plases/equipment ~/ski_plases/equiplist
abrovkin@fedora:~/ski_plases$ touch abc
abrovkin@fedora:~/ski_plases$ mv abc ~/ski_plases/equipment
abrovkin@fedora:~/ski_plases$ touch abc: Нет такого файла или каталога
abrovkin@fedora:~/ski_plases$ mv abc ~/ski_plases/equipment
abrovkin@fedora:~/ski_plases$ cd
abrovkin@fedora:~/ski_plases$ mv ~/ski_plases/equiplist ~/ski_plases/equiplist2 ~/ski_plases/equipment
abrovkin@fedora:~/ski_plases$ rm abc
abrovkin@fedora:~/ski_plases$ cd ..
abrovkin@fedora:~/ski_plases$ mv ~/ski_plases/mkdir ~/ski_plases/plans
abrovkin@fedora:~/ski_plases$ ls
abrovkin@fedora:~/ski_plases$ cd
abrovkin@fedora:~/ski_plases$ mv ~/ski_plases/equiplist ~/ski_plases/equiplist2 ~/ski_plases/equipment
abrovkin@fedora:~/ski_plases$ rm abc
abrovkin@fedora:~/ski_plases$ cd ..
abrovkin@fedora:~/ski_plases$ mv ~/ski_plases/mkdir ~/ski_plases/plans
abrovkin@fedora:~/ski_plases$ ls
```

Figure 7: Продолжаю выполнять примеры

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

3.1. drwxr-r- ... australia

The screenshot shows a Linux desktop environment with a terminal window and a browser window.

Terminal Window:

```
[abrovkin@fedora ~]$ mv io.h equipment
[abrovkin@fedora ~]$ mv equipment ->ski.plases
[abrovkin@fedora ~]$ mv equipment ->ski.plases/equipment
mv: не удастся изменить имя для `ski.plases/equipment': Нет такого файла или каталога
[abrovkin@fedora ~]$ mv equipment ->ski.plases/equiplist
[abrovkin@fedora ~]$ mv abc1 ->ski.plases/equiplist
[abrovkin@fedora ~]$ mv abc1 ->abc1
[abrovkin@fedora ~]$ mkdir abc1
mkdir: невозможно создать каталог `abc1': файл существует
[abrovkin@fedora ~]$ cp abc1 ~/ski.plases
[abrovkin@fedora ~]$ cp abc1 ~/ski.plases/abc1 ->ski.plases/equiplist2
[abrovkin@fedora ~]$ cd ~/ski.plases/equiplist2
[abrovkin@fedora ski.plases]$ ls
[abrovkin@fedora ski.plases]$ ls abc1
[abrovkin@fedora ski.plases]$ ls equipment
[abrovkin@fedora ~]$ cd
[abrovkin@fedora ~]$ mv ->/ski.plases/equiplist2 ~/ski.plases/equipment
[abrovkin@fedora ~]$ mv abc1 ->abc1
[abrovkin@fedora ~]$ mv newdir ->ski.plases
[abrovkin@fedora ~]$ mv newdir ->ski.plases/newdir ->ski.plases/plans
[abrovkin@fedora ~]$ ls -l
ls: command not found..
[abrovkin@fedora ~]$ ls -l
[abrovkin@fedora ~]$ ls -l
ls: command not found..
[abrovkin@fedora ~]$ ls -l
ls: command not found..
```

Browser Window:

The browser window displays a URL: https://fesystem.rudn.ru/pluginfile.php/1383579/mod_resource/content/4/005-lab_1.pdf. The page content is partially visible, showing instructions related to file operations and permissions.

Page Content (Visible Text):

2.8. Создайте и переместите каталог ~ /newdir в каталог ~ /ski.plases и назовите его plans.

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. drwxr-xr-x ... play
- 3.3. -rwxr--r-- ... my_os
- 3.4. -twxrwxr-- ... feathers

При необходимости создайте нужные файлы.

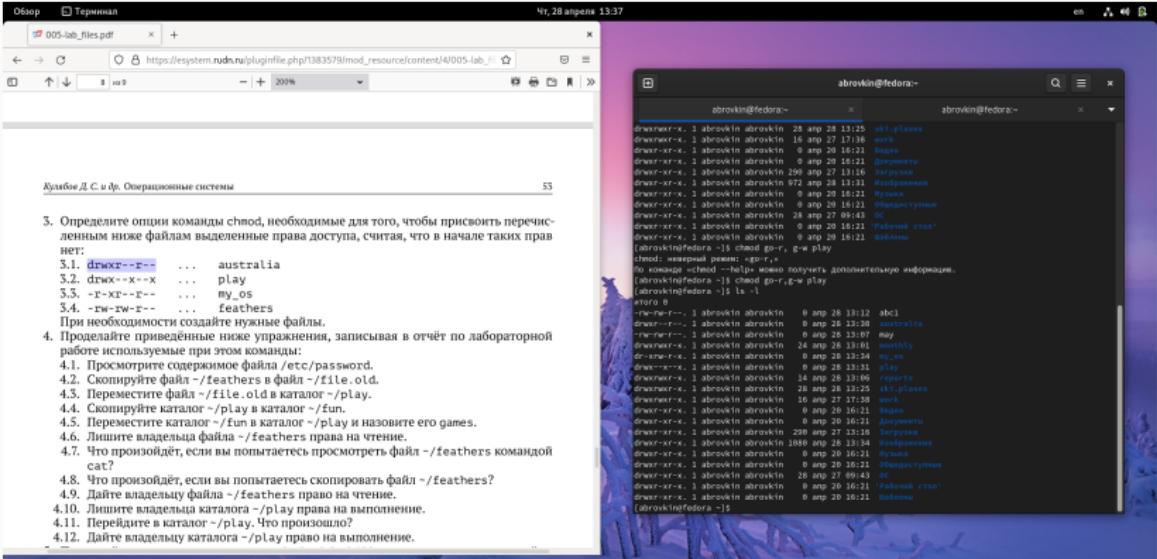
4. Проделайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:

- 4.1. Просмотрите содержимое файла /etc/password.
- 4.2. Скопируйте файл ~/feathers в файл ~/file.old.
- 4.3. Переместите файл ~/file.old в каталог ~/play.
- 4.4. Скопируйте каталог ~/play в каталог ~/fun.
- 4.5. Переместите каталог ~/fun в каталог ~/play и назовите его games.
- 4.6. Лишите владельца файла ~/feathers права на чтение.
- 4.7. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?

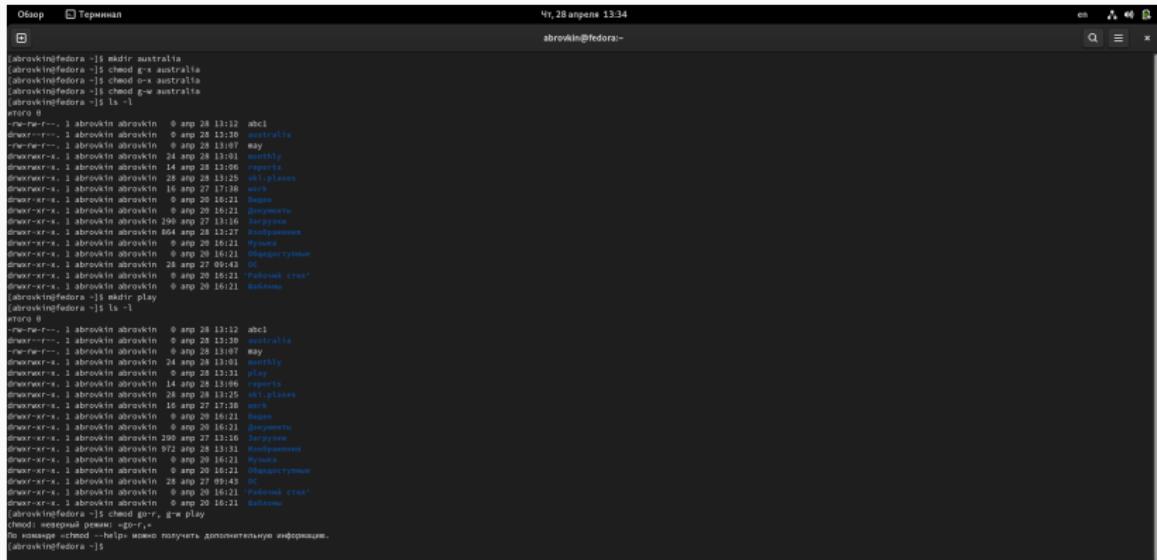
Обзор Терминал

Чт, 28 апреля 13:31
abrovkin@fedora:~

```
abrovkin@fedora:~$ ls a*
abrovkin@fedora:~$ ls aadir australia
abrovkin@fedora:~$ ls chmod g+* australia
abrovkin@fedora:~$ ls chmod o+* australia
abrovkin@fedora:~$ ls -l g+* australia
abrovkin@fedora:~$ ls -l
total 8
drwxr-xr-- 3 abrovkin abrovkin 0 апр 28 13:02 abc1
drwxr-xr-- 3 abrovkin abrovkin 0 апр 28 13:20 australia
drwxr-xr-- 3 abrovkin abrovkin 0 апр 28 13:07 bay
drwxr-xr-- 3 abrovkin abrovkin 24 апр 28 13:01 monthly
drwxr-xr-- 3 abrovkin abrovkin 14 апр 28 13:06 reports
drwxr-xr-- 3 abrovkin abrovkin 28 апр 28 13:05 skiplaces
drwxr-xr-- 3 abrovkin abrovkin 0 апр 28 13:04 test
drwxr-xr-- 3 abrovkin abrovkin 0 апр 20 16:21 вакансии
drwxr-xr-- 3 abrovkin abrovkin 0 апр 20 16:21 документы
drwxr-xr-- 3 abrovkin abrovkin 290 апр 27 13:16 загрузки
drwxr-xr-- 3 abrovkin abrovkin 0 апр 20 16:21 изображения
drwxr-xr-- 3 abrovkin abrovkin 0 апр 20 16:21 музыка
drwxr-xr-- 3 abrovkin abrovkin 0 апр 20 16:21 общедоступные
drwxr-xr-- 3 abrovkin abrovkin 28 апр 27 09:43 ис
drwxr-xr-- 3 abrovkin abrovkin 0 апр 20 16:21 Рабочий стол
drwxr-xr-- 3 abrovkin abrovkin 0 апр 20 16:21 рабочее
abrovkin@fedora:~$ ls
```



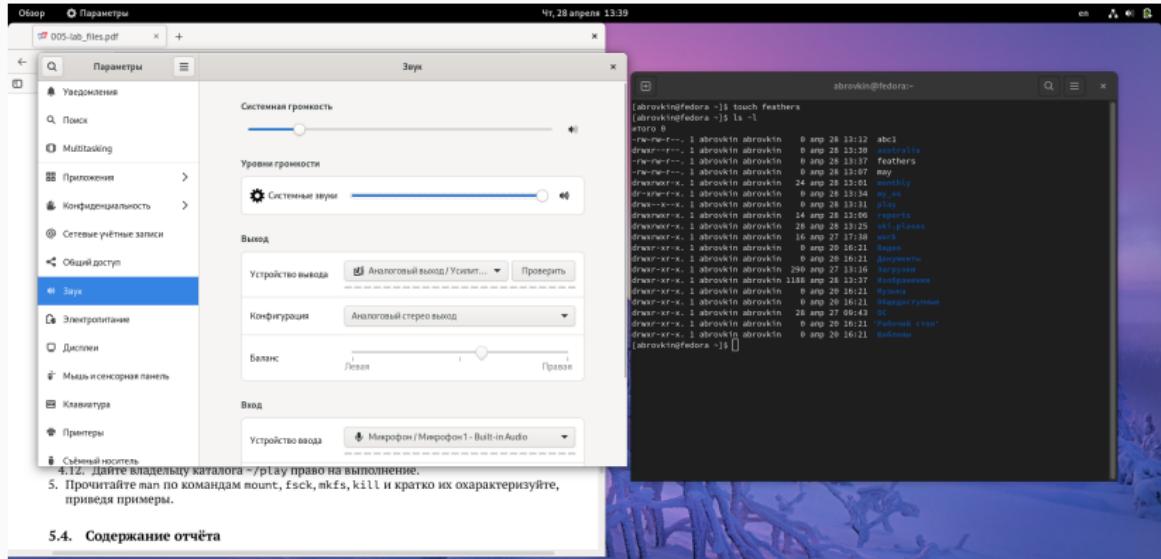
3.2. drwx-x-x ... play



The screenshot shows a terminal window with the following content:

```
абровкин@абровкин: ~$ chmod g+x australia
абровкин@абровкин: ~$ chmod g+x australia
абровкин@абровкин: ~$ chmod g+x australia
абровкин@абровкин: ~$ chmod g+w australia
абровкин@абровкин: ~$ ls -l
total 9
drwxrwxr-x 3 abrovkin abrovkin 0 апр 28 13:12 abc1
drwxrwxr-x 3 abrovkin abrovkin 0 апр 28 13:10 australia
drwxrwxr-x 3 abrovkin abrovkin 0 апр 28 13:07 may
drwxrwxr-x 3 abrovkin abrovkin 0 апр 28 13:05 may
drwxrwxr-x 3 abrovkin abrovkin 14 апр 28 13:06 reports
drwxrwxr-x 3 abrovkin abrovkin 28 апр 28 13:25 скроллз
drwxrwxr-x 3 abrovkin abrovkin 16 апр 27 17:38 work
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 before
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 after
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 before
drwxrwxr-x 3 abrovkin abrovkin 290 апр 27 13:16 загрузки
drwxrwxr-x 3 abrovkin abrovkin 864 апр 28 13:27 подключения
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 Публика
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 Публика
drwxrwxr-x 3 abrovkin abrovkin 28 апр 27 09:45 ОС
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 Рабочий стол
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 веббрауз
абровкин@абровкин: ~$ ls -l
total 9
drwxrwxr-x 3 abrovkin abrovkin 0 апр 28 13:12 abc1
drwxrwxr-x 3 abrovkin abrovkin 0 апр 28 13:10 australia
drwxrwxr-x 3 abrovkin abrovkin 0 апр 28 13:07 may
drwxrwxr-x 3 abrovkin abrovkin 0 апр 28 13:05 may
drwxrwxr-x 3 abrovkin abrovkin 14 апр 28 13:06 reports
drwxrwxr-x 3 abrovkin abrovkin 28 апр 28 13:25 скроллз
drwxrwxr-x 3 abrovkin abrovkin 16 апр 27 17:38 work
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 before
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 after
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 before
drwxrwxr-x 3 abrovkin abrovkin 290 апр 27 13:16 загрузки
drwxrwxr-x 3 abrovkin abrovkin 972 апр 28 13:27 подключения
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 Публика
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 Публика
drwxrwxr-x 3 abrovkin abrovkin 28 апр 27 09:45 ОС
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 Рабочий стол
drwxrwxr-x 3 abrovkin abrovkin 0 апр 20 16:21 веббрауз
абровкин@абровкин: ~$ chmod g+r, g+w australia
абровкин@абровкин: ~$ chmod -hhelp
Чтобы: краткий режим: «g+r, g+w»
Чтобы: полное: «chmod -hhelp» можно получить дополнительную информацию.
абровкин@абровкин: ~$
```

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



4. Проделал приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 - 4.1. Не просмотрел содержимое файла /etc/password, так как у меня его нет.
 - 4.2. Скопировал файл ~/feathers в файл ~/file.old командой cp.

- 4.3. Переместил файл ~/file.old в каталог ~/play командой mv.
- 4.4. Скопировал каталог ~/play в каталог ~/fun командой cp -r.
- 4.5. Переместил каталог ~/fun в каталог ~/play командой mv и назвал его games командой mv.

- 4.6. Лишил владельца файла ~/feathers права на чтение командой chmod u-r.
- 4.7. Если попытаться скопировать файл ~/feathers командой cp, то выведется:
- 4.8. Дал владельцу файла ~/feathers право на чтение командой chmod u+r.

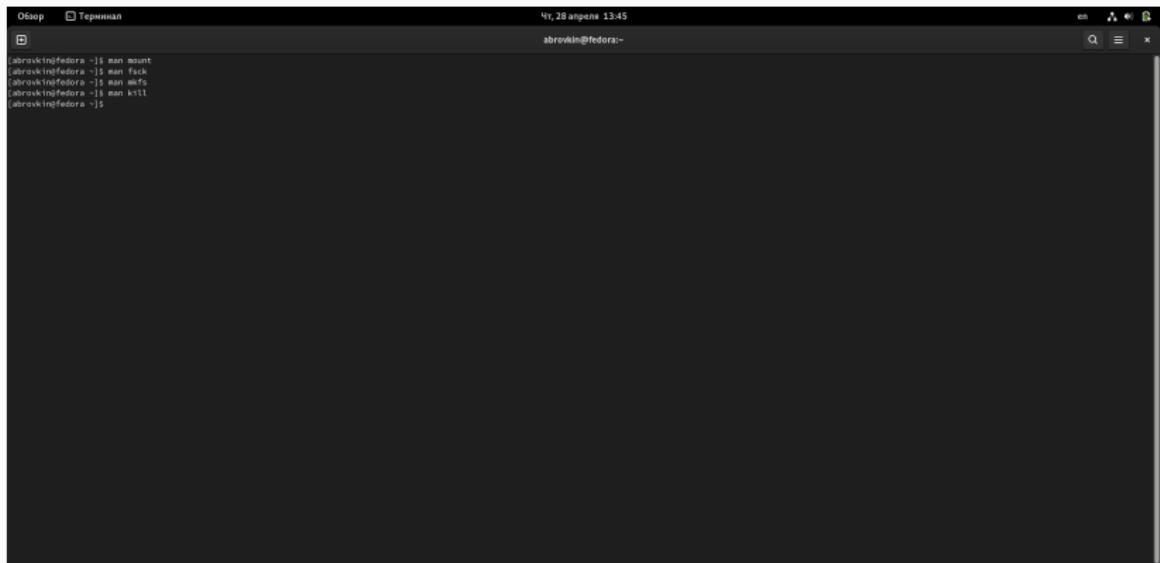
- 4.9. Лишил владельца каталога ~/play права на выполнение командой chmod u-x.
- 4.10. Попытался перейти в каталог ~/play командой cd.
- 4.11. Дал владельцу каталога ~/play право на выполнение командой chmod u+x.

```
Обзор Терминал abrovkin@fedora:~
```

```
abrovkin@fedora ~]$ ls
abc1    may     play    work   Загрузки  Общедоступные  Вебпапки
australia  mostly  reports  Neuro   Избранные  ОС          Папка  "Рабочий стол"
australia  mostly  reports  Neuro   Избранные  ОС          Папка  "Рабочий стол"
abrovkin@fedora ~]$ cp -r feathers ~/file.old
abrovkin@fedora ~]$ mv ~/file.old ~/play
abrovkin@fedora ~]$ cp -r ./fun ./play
abrovkin@fedora ~]$ ex ./fun ./play
abrovkin@fedora ~]$ cp -r ./fun ./play
abrovkin@fedora play]$ chmod u=r feathers
abrovkin@fedora play]$ chmod u=r feathers
chmod: невозможно получить доступ к `feathers': Нет такого файла или каталога
abrovkin@fedora play]$ chmod u=r feathers
abrovkin@fedora play]$ chmod u=r feathers
abrovkin@fedora play]$ chmod u=r play
abrovkin@fedora ~]$ cd ~/play
bash: cd: /home/abrovkin/play: Отказано в доступе
abrovkin@fedora ~]$ chmod u=r play
abrovkin@fedora ~]$
```

Figure 8: На скриншоте все ответы на данные пункты

5. Прочитал man по командам mount, fsck, mkfs, kill.

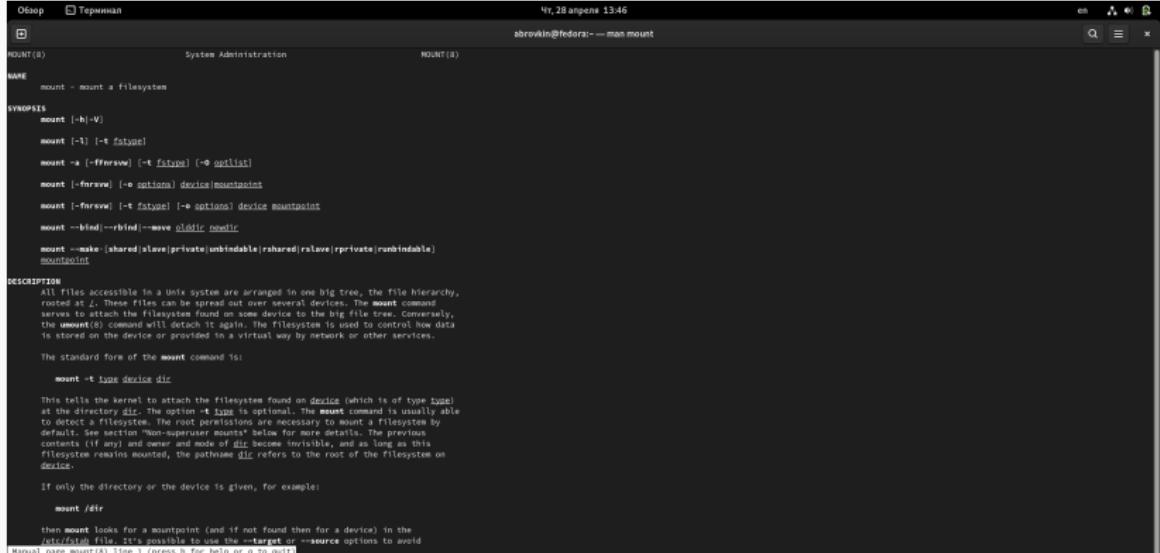


The screenshot shows a terminal window titled 'Терминал' (Terminal) with a dark background. At the top, it displays the date and time: 'Чт, 28 апреля 13:45'. Below that is the user's prompt: 'abrosvin@fedora:~\$'. The terminal window contains the following text:
abrosvin@fedora ~]\$ man mount
abrosvin@fedora ~]\$ man fsck
abrosvin@fedora ~]\$ man mkfs
abrosvin@fedora ~]\$ man kill
abrosvin@fedora ~]\$
The text is white and clearly legible against the dark background.

Figure 9: команда man

mount

mount



The screenshot shows a terminal window with the title 'Онлайн' and 'Терминал'. The command 'man mount' has been run, displaying the manual page for the 'mount' command. The page includes sections for NAME, SYNOPSIS, DESCRIPTION, and EXAMPLES.

NAME
mount - mount a filesystem

SYNOPSIS
mount [-h|-V]
mount [-t] [-t fstype] [device|mountpoint]
mount -a [-ffsraw] [-t fstype] [-o options]
mount [-frsvw] [-o options] device|mountpoint
mount [-frsvw] [-t fstype] [-o options] device mountpoint
mount --bind --rbind |--move olddir newdir
mount --make [shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable]
mountpoint

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 associates each file with its device name relative to the base 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.

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
```

then **mount** looks for a mountpoint (and if not found then for a device) in the **/etc/fstab** file. It's possible to use the **--target** or **--source** options to avoid **Manual page mount(8)**. Line 1 (press h for help or q to quit).

Figure 10: mount

Обзор Терминал

4ч, 28 април 13:46

abrovkin@fedora: ~ — man mount

```
If only the directory or the 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 the --target or --source options to avoid  
ambiguous interpretation of the given argument. For example:  
mount --target /mountpoint  
The same filesystem may be mounted more than once, and in some cases (e.g., network  
filesystems), the same filesystem may be mounted on the same mountpoint multiple times.  
The mount command does not implement any protection against this. The behavior  
is controlled by the kernel and it is usually specific to the filesystem driver. The  
exception is --all, in this case already mounted filesystems are ignored (see --all  
below for more details).  
Listing the mounts  
The listing mode is maintained for backward compatibility only.  
For more robust and customizable output use findmnt(), especially in your scripts.  
Note that control characters in the mountpoint name are replaced with '?'.  
The following command lists all mounted filesystems (of type tmpfs):  
mount [-t] [-t type]  
The option -t adds labels to this listing. See below.  
Indicating the device and filesystem  
The mount command indicates the filenames (or a block special device), like /dev/sda1,  
but there are other possibilities. For example, in the case of an NFS mount, device may  
look like /nfs.org:/dir.  
The device names of disk partitions are unstable; hardware reconfiguration, and adding  
or removing a device can cause changes in names. This is the reason why it's strongly  
recommended to use filesystem or partition identifiers like UUID or LABEL. Currently  
supported identifiers (tags):  
LABEL=label  
Human readable filesystem identifier. See also -L.  
UUID=uuid  
Filesystem universally unique identifier. The format of the UUID is usually a  
series of hex digits separated by hyphens. See also -B.  
Manual page mount(8) line 41 (press h for help or q to quit)
```

```
Обзор Терминал 4ч, 28 април 13:46
abrovkin@fedora:~ -- man mount

Note that mount uses UUIDs as strings. The UUIDs from the command line or from
fstab(5) are not converted to internal binary representation. The string
representation of the UUID should be based on lower case characters.

PARTLABEL=uuid
Human readable partition identifier. This identifier is independent on filesystem
and does not change by mfs or mkswap operations It's supported for example for
GUID Partition Tables (GPT).

PARTUUID=uuid
Partition universally unique identifier. This identifier is independent on
filesystems and does not change by mfs or mkswap operations It's supported for
example for GUID Partition Tables (GPT).

ID=id
Hardware block device ID as generated by udev. This identifier is usually based on
WWN (unique storage identifier) and assigned by the hardware manufacturer. See ls
/dev/disk/by-id for more details, this directory and running udevadm is required.
This identifier is not recommended for generic use as the identifier is not
strictly defined and it depends on udev, udev rules and hardware.

The command blkid --fs provides an overview of filesystems, LABELs and UUIDs on
available block devices. The command blkid -p <device> provides details about a
filesystem on the specified device.

Don't forget that there is no guarantee that UUIDs and labels are really unique,
especially if you move, share or copy the device. Use blkid -w +UUID,PARTUUID to verify
that the UUIDs are really unique in your system.

The recommended setup is to use tags (e.g. UUID=uuid) rather than
/dev/disk/by-label/uuid.id.partUUID/partUUID udev symlinks in the /etc/fstab file.
Tags are more readable, robust and portable. The mount() command internally uses udev
symlinks, so the use of symlinks in /etc/fstab has no advantage over tags. For more
details see blkid(8).

The proc filesystem is not associated with a special device, and when mounting it, an
arbitrary keyword – for example, proc – can be used instead of a device specification.
(The customary choice none is less fortunate: the error message ‘‘none already mounted’’
from mount can be confusing.)

The files /etc/fstab, /etc/mtab and /proc/mounts
The file /etc/fstab (see fstab(5)), may contain lines describing what devices are
usually mounted where, using which options. The default location of the fstab(5) file
can be overridden with the --fstab path command-line option (see below for more
details).

Manual page mount(8) line 86 (press h for help or q to quit)
```

```
Обзор Терминал 4ч, 28 април 13:46
abrovkin@fedora:~ -- man mount

-F, --fork
        (specify in conjunction with -o.) Fork off a new incarnation of mount for each device.
        This will do the mounts on different devices or different NFS servers in parallel.
        This has the advantage that it is faster; also NFS timeouts proceed in parallel. A
        disadvantage is that the order of the mount operations is undefined. Thus, you
        cannot use this option if you want to mount both /usr and /usr/local.

-f, --fake
        Causes everything to be done except for the actual system call; if it's not
        obvious, this "fakes" mounting the Filesystem. This option is useful in conjunction
        with the -v option to determine what the mount command is trying to do. It can also
        be used with mount --fake for doing things like umount -v without actually getting
        The -f option checks for an existing record in /etc/mtab and fails when the record
        already exists (with a regular non-fake mount, this check is done by the kernel).

-i, --internal-only
        Don't call the /sbin/mount.filesystem helper even if it exists.

-L, --label label
        Mount the partition that has the specified label.

-l, --show-labels
        Add the labels in the mount output. mount must have permission to read the disk
        device (e.g. be set-user-ID root) for this to work. One can set such a label for
        ext2, ext3 or ext4 using the e2label() utility, or for XFS using xfs_admin(), or
        for reiserfs using reiserfstune().

-N, --move
        Move a subtree to some other place. See above, the subsection The move operation.

-n, --no-stab
        Mount without writing in /etc/mtab. This is necessary for example when /etc is on a
        read-only filesystem.

-N, --namespace ns
        Perform the mount operation in the mount namespace specified by ns. ns is either
        PID of process running in that namespace or special file representing that
        namespace.

mount switches to the mount namespace when it reads /etc/fstab, writes (etc/mtab);
isc writes to /run/mount and calls the mount(2) system call, otherwise it runs in
the original mount namespace. This means that the target namespace does not have to
contain any libraries or other requirements necessary to execute the mount(2) call.

See mount_namespaces(7) for more information.
Manual page mount(8) line 366 (press H for help or q to quit).
```

```
Обзор Терминал 4ч, 28 април 13:46
abrovkin@fedora:~ — man mount

mount LABEL=/dev/sda1 -o noatime,nodev,noatid
For more details, see the FILESYSTEM-INDEPENDENT MOUNT OPTIONS and
FILESYSTEM-SPECIFIC MOUNT OPTIONS sections.

--options-mode mode
Controls how to combine options from /etc/fstab with options from the command line.
Options are of type ignore, append, prepend or replace. For example, append means
that options from /etc/fstab are appended to options from the command line. The default
value is prepend – it means command line options are evaluated after /etc/fstab options.
Note that the last option wins if there are conflicting ones.

--options-source source
Source of default options. source is a comma-separated list of /etc/fstab, /etc/mtab and
disabled. disabled disables /etc/fstab and /etc/mtab and disables --options-source-force. The
default value is /etc/fstab,/etc/mtab.

--options-source-force
Use options from /etc/fstab/mtab even if both device and dir are specified.

-R, --rbind
Remounts a subtree and all possible submounts somewhere else (so that its contents
are available in both places). See above, the subsection Bind mounts.

-r, --read-only
Mount the filesystem read-only. A synonym is -o ro.

Note that, depending on the filesystem type, state and kernel behavior, the system
may still write to the device. For example, ext3 and ext4 will replay the journal
if the filesystem is dirty. To prevent this kind of write access, you may want to
mount an ext3 or ext4 filesystem with the noatime mount option or set the block
device itself to read-only mode, see the blockdev(8) command.

-s
Tolerate sloppy mount options rather than failing. This will ignore mount options
not supported by a filesystem type. Not all filesystems support this option.
Currently it's supported by the mount.efs mount helper only.

--source device
If only one argument for the mount command is given, then the argument might be
interpreted as the target (mountpoint) or source (device). This option allows you
to explicitly define that the argument is the mount source.

--target directory
If only one argument for the mount command is given, then the argument might be
interpreted as the target (mountpoint) or source (device). This option allows you

Manual page mount(8) line 436 (press h for help or q to quit)
```

```
Обзор Терминал 4ч, 28 април 13:46
abrovkin@fedora:— man mount

SYNOPSIS
It is possible for a corrupted filesystem to cause a crash.

Some Linux filesystems don't support -o sync and -o dirsync (the ext2, ext3, ext4, fat and vfat filesystems do support synchronous updates (a la BSD) when mounted with the sync option).

The -e remount may not be able to change mount parameters (all ext2fs-specific parameters, except sb, are changeable with a remount, for example, but you can't change gid or umask for the fatfs).

It is possible that the files /etc/mtab and /proc/mounts don't match on systems with a regular mtab file. The first file is based only on the mount command options, but the content of the second file also depends on the kernel and others settings (e.g. on a remote NFS server – in certain cases the mount command may report unreliable information about the mount point and the /proc/mounts file usually contains more reliable information). This is another reason to replace the mtab file with a symlink to the /proc/mounts file.

Checking files on NFS filesystems referenced by file descriptors (i.e. the fstat and fcntl families of functions) may lead to inconsistent results due to the lack of a consistency check in the kernel even if the nodev mount option is used.

The loop option with the offset or sizefrom options used may fail when using older kernels. It is possible that the size of the block device has been configured as requested. This situation can be worked around by using the losetup(8) command manually before calling mount with the configured loop device.

AUTHORS
Kernel Lin chagrin@redhat.com

SEE ALSO
mount(2), umount(2), filesystems(5), fstab(5), nfs(5), xfs(5), mount_namespaces(7),
ext(7), extlabel(8), findmnt(8), losetup(8), lsblk(8), mkfs(8), mount(8), nfcd(8),
swapon(8), unshare(8), umount(8), xfs_admin(8)

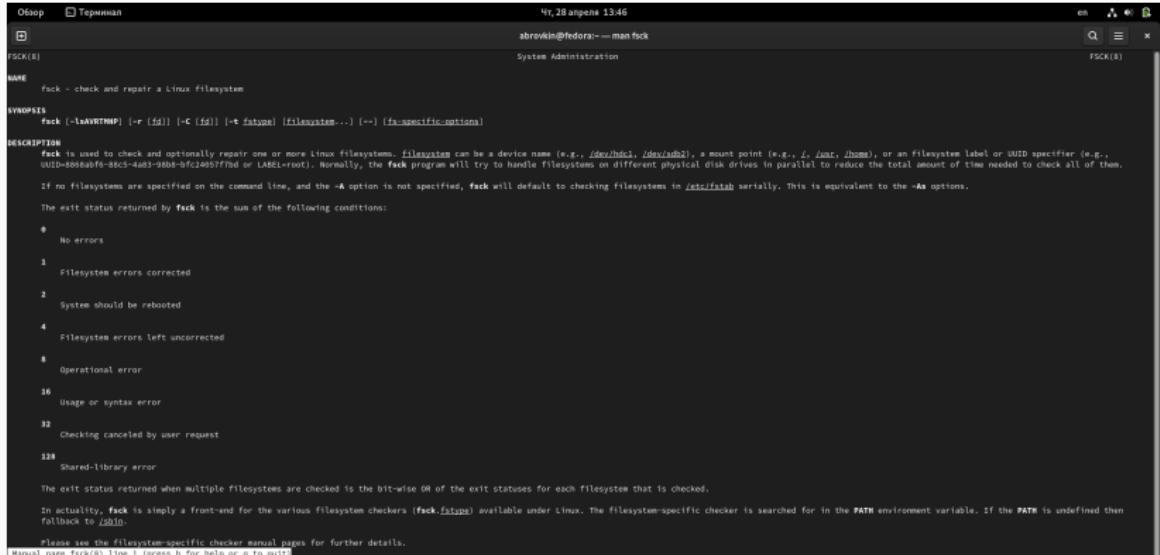
REPORTING BUGS
For bug reports, use the issue tracker at
https://github.com/torvalds/linux/issues.

AVAILABILITY
The mount command is part of the util-linux package which can be downloaded from https://www.kernel.org/pub/linux/utils/util-linux/.
```

util-linux 2.37.2 2021-08-16 MOUNT(8)
Manual page mount(8) line 1045/2090 (END) (press h for help or q to quit)

fsck

fsck



The screenshot shows a terminal window titled "fsck(8)" with the command "man fsck" entered. The window title bar also displays "abrovkin@fedora:~" and the date and time "4/1, 28 amern 13:46". The terminal content is the man page for fsck, which includes sections for NAME, SYNOPSIS, DESCRIPTION, and EXIT STATUS. It describes the fsck command as a tool for checking and repairing Linux filesystems, mentioning options like -A, -C, -t, and -v, and detailing the exit status codes (0 to 128) and their meanings.

```
NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lqvVRTMP] [-e (fd)] [-C (fd)] [-t filetype] [filesystem...] (-e) [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/sda2), a mount point (e.g., /, /var, /home), or an filesystem label or UUID specifier (e.g., UUID=868babf6-98c3-4083-98d8-0fc240577bd4 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 -Aa options.

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

    In actuality, fsck is simply a front-end for the various filesystem checkers (fsck, fsuges) 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.
Manual page fsck(8) line 1 (press h for help or q to quit)
```

Figure 11: fsck

```
Обзор Терминал 4ч, 28 апенди 13:46
abrovkin@fedora:~-- man fsck
prefixed by a negation operator, then only those listed filesystems will be checked.

Options specifiers may be included in the comma-separated fslist. They must have the format optspec=option. If an options specifier is present, then only filesystems which contain fs=option in their mount options field of /etc/fstab will be checked. If the options specifier is prefixed by a negation operator, then only those filesystems that do not have fs=option in their mount options field of /etc/fstab will be checked.

For example, if optstr appears in fslist, then only filesystems listed in /etc/fstab with the re option will be checked.

For compatibility with Mandrake distributions whose boot scripts depend upon an unauthorized UI change to the fsck program, if a filesystem type of loop is found in fslist, it is treated as if opts=loop were specified as an argument to the -t option.

Normally, the filesystem type is deduced by searching for filesystem in the /etc/fstab file and using the corresponding entry. If the type cannot be deduced, and there is only a single filesystem given as an argument to the -t option, fsck will use the specified filesystem type. If this type is not available, then the default filesystem type (currently ext2) is used.

-A Walk through the /etc/fstab file and try to check all filesystems in one run. This option is typically used from the /etc/rc system initialization file, instead of multiple commands for checking a single filesystem.

The root filesystem will be checked first unless the -P option is specified (see below). After that, filesystems will be checked in the order specified by the fs_passno (the sixth) field in the /etc/fstab file. Filesystems with a fs_passno value of 0 are skipped and are not checked at all. Filesystems with a fs_passno value of greater than zero will be checked in order, with filesystems with the lowest fs_passno number being checked first. If there are multiple filesystems with the same pass number, fsck will attempt to check them in parallel, although it will avoid running multiple filesystem checks on the same physical disk.

fsck does not check stacked devices (RAIDs, de-crypt, ...) in parallel with any other device. See below for FSCK_FORCE_ALL_PARALLEL setting. The /vss filesystem is used to determine dependencies between devices.

Hence, a very common configuration in /etc/fstab files is to set the root filesystem to have a fs_passno value of 1 and to set all other filesystems to have a fs_passno value of 2. This will allow fsck to automatically run filesystem checkers in parallel if it is advantageous to do so. System administrators might choose not to use this configuration if they need to avoid multiple filesystem checks running in parallel for some reason - for example, if the machine in question is short on memory so that excessive paging is a concern.

fsck normally does not check whether the device actually exists before calling a filesystem specific checker. Therefore non-existing devices may cause the system to enter filesystem repair mode during boot if the filesystem specific checker returns a fatal error. The /etc/fstab mount option noauto may be used to have fsck skip non-existing devices. fsck also skips non-existing devices that have the special filesystem type auto.

-C [file] Display completion/progress bars for those filesystem checkers (currently only for ext[234]) which support them. fsck will manage the filesystem checkers so that only one of them will display a progress bar at a time. GUI front-ends may specify a file descriptor fd, in which case the progress bar information will be sent to that file descriptor.

-H Do not check mounted filesystems and return an exit status of 0 for mounted filesystems.

-N Don't execute, just show what would be done.

-P When the -A flag is set, check the root filesystem in parallel with the other filesystems. This is not the safest thing in the world to do, since if the root filesystem is in doubt things like the #!/fsck() executable might be corrupted! This option is mainly provided for those sysadmins who don't want to repartition the root filesystem to be small and compact (which is really the right solution).

-R When checking all filesystems with the -A flag, skip the root filesystem. (This is useful in case the root filesystem has already been mounted read-write.)
```

Manual page **fsck(8)** (line 7) (press H for help or q to quit)

```
Обзор Терминал 4ч, 28 април 13:46
abrovkin@fedora:— man fsck

Options and arguments which follow the -- are treated as filesystem-specific options to be passed to the filesystem-specific checker.

Please note that fsck is not designed to pass arbitrarily complicated options to filesystem-specific checkers. If you're doing something complicated, please just execute the filesystem-specific checker directly. If you pass fsck some horribly complicated options and arguments, and it doesn't do what you expect, don't bother reporting it as a bug. You're almost certainly doing something that you shouldn't be doing with fsck. Options to different filesystem-specific fsck's are not standardized.

ENVIRONMENT
The fsck program's behavior is affected by the following environment variables:

FSCK_FORCE_ALL_PARALLEL
If this environment variable is set, fsck will attempt to check all of the specified filesystems in parallel, regardless of whether the filesystems appear to be on the same device. (This is useful for RAID systems or high-end storage systems such as those sold by companies such as IBM or EMC.) Note that the fs_passes value is still used.

FSCK_MAX_INST
This environment variable will limit the maximum number of filesystem checkers that can be running at one time. This allows configurations which have a large number of disks to avoid fsck starting too many filesystem checkers at once, which might overload CPU and memory resources available on the system. If this value is zero, then an unlimited number of processes can be spawned. This is currently the default, but future versions of fsck may attempt to automatically determine how many filesystem checks can be run based on gathering accounting data from the operating system.

PATH
The PATH environment variable is used to find filesystem checkers.

FSTAB_FILE
This environment variable allows the system administrator to override the standard location of the /etc/fstab file. It is also useful for developers who are testing fsck.

LIBBLKID_DEBUG=all
enables libblkid debug output.

LIBMOUNT_DEBUG=all
enables libmount debug output.

FILES
/etc/fstab

AUTHORS
 Theodore Ts'o <tytso@mit.edu>, Karel Zak <kzak@redhat.com>

SEE ALSO
 fstab(5), mke2fs(8), fsck.ext2(8) or fsck.ext3(8) or e2fsck(8), fsck.cramfs(8), fsck.jfs(8), fsck.hfs(8), fsck.minix(8), fsck.esdss(8), fsck.vfat(8), fsck.xfs(8), reiserfsck(8)

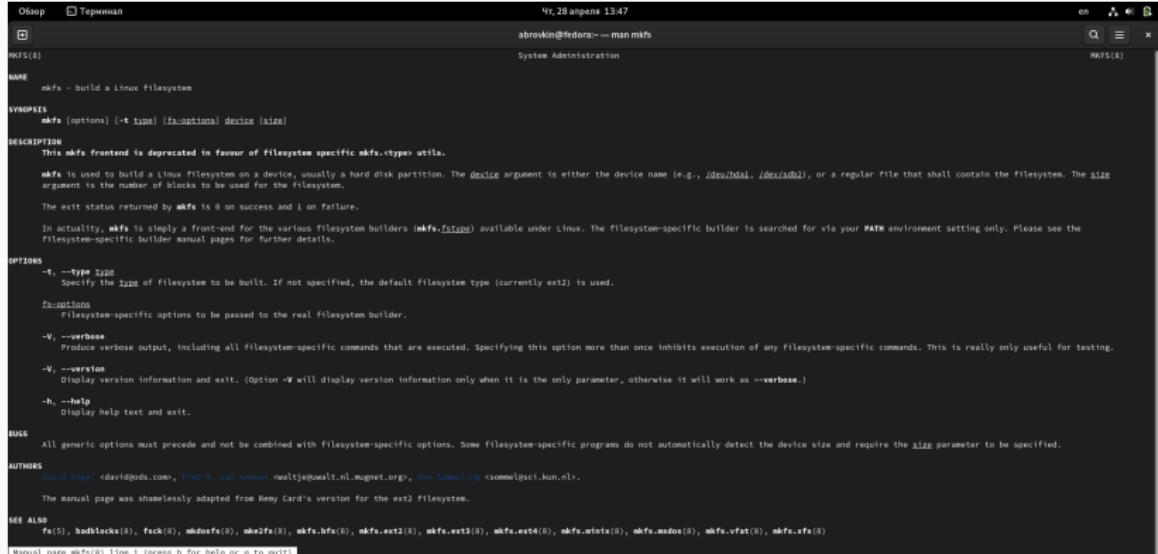
REPORTING BUGS
For bug reports, use the issue tracker at https://github.com/karelzak/util-linux/issues.

AVAILABILITY
The fsck command is part of the util-linux package which can be downloaded from Linux Kernel Archive https://www.kernel.org/pub/linux/utils/util-linux/,

util-linux 2.37.2 2021-07-20 FSCK(8)
Manual page fsck(8) line 134/170 (END) (press h for help or q to quit)
```


mkfs

mkfs



mkfs(8) Terminus 44, 28 април 13:47
abrovin@fedora: ~ man mkfs
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/hd1**, **/dev/sd02**), 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.

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.

BUGS
All generic options must precede and not be combined with filesystem-specific options. Some filesystem-specific programs do not automatically detect the device size and require the **size** parameter to be specified.

AUTHORS
David Engel <david@ods.com>, Fred N. van Kempen <waltje@walt.nl>, Ron Sommeling <sommelgsci.kun.nl>.
The manual page was shamelessly adapted from Remy Card's version for the ext2 filesystem.

SEE ALSO
fs(5), **badblocks(8)**, **fsck(8)**, **mkdosfs(8)**, **mke2fs(8)**, **mkfs.btrfs(8)**, **mkfs.ext2(8)**, **mkfs.ext3(8)**, **mkfs.ext4(8)**, **mkfs.minix(8)**, **mkfs.msdos(8)**, **mkfs.vfat(8)**, **mkfs.xfs(8)**

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

Figure 12: mkfs

kill

kill

```
Обзор Терминал 4ч, 28 април 13:47
abrovkin@fedora: ~ man kill
User Commands
KILL(1)                                         User Commands                                         KILL(1)

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds] signal [-c] pid|name...
    kill -l [named] [-L]

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.

    Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all, --pid, and --queue options, and the possibility to specify processes by command name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is still performed.

ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.

pid
    Each pid can be expressed in one of the following ways:
        0
            where 0 is larger than 0. The process with PID 0 is signaled.
        -
            All processes in the current process group are signaled.
        -1
            All processes with a PID larger than 1 are signaled.
        -g
            where g is larger than 1. All processes in process group g are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either a signal must be specified first, or the argument must be preceded by a '--' option, otherwise it will be taken as the signal to send.

name
    All processes invoked using this name will be signaled.

OPTIONS
    -s, --signal signal
Manual page kill(1) line 1 (press h for help or q to quit).
```

Figure 13: kill

Обзор Терминал

4ч, 28 апреля 13:47
abrovkin@fedora: ~ man kill

```
NAME
    All processes invoked using this name will be signaled.

OPTIONS
    -s, --signal signal
        The signal to send. It may be given as a name or a number.

    -l, --list [number]
        Print a list of signal names, or convert the given signal number to a name. The signals can be found in /usr/include/linux/signal.h.

    -L, --table
        Similar to -l, but it will print signal names and their corresponding numbers.

    -a, --all
        Do not restrict the command-name-to-PID conversion to processes with the same UID as the present process.

    -P, --pid
        Only print the process ID (PID) of the named processes, do not send any signals.

    --verbose
        Print PID(s) that will be signaled with kill along with the signal.

    -q, --queue value
        Send the signal using sigqueue(3) rather than kill(2). The value argument is an integer that is sent along with the signal. If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigaction(3), then it can obtain this data via the si_value field of the siginfo_t structure.

    --timeout milliseconds signal
        Send a signal defined in the usual way to a process, followed by an additional signal after a specified delay. The --timeout option causes kill to wait for a period defined in milliseconds before sending a follow-up signal to the process. This feature is implemented using the Linux kernel FD file descriptor feature in order to guarantee that the follow-up signal is sent to the same process or not sent if the process no longer exists.

        Note that the operating system may reuse PIDs and implementing an equivalent feature in a shell using kill and sleep would be subject to races whereby the follow-up signal might be sent to a different process that used a recycled PID.

        The --timeout option can be specified multiple times: the signals are sent sequentially with the specified timeouts. The --timeout option can be combined with the --queue option.

        As an example, the following command sends the signals QUIT, TERM and KILL in sequence and waits for 1000 milliseconds between sending the signals:
            kill --verbose --timeout 1000 TERM --timeout 1000 KILL \
                --signal QUIT 12345

EXIT STATUS
    kill has the following exit status values:
        0
Manual page kill(1) line 41 (press h for help or q to quit)
```

Обзор Терминал 4ч, 28 април 13:47 abrovkin@fedora:~ -- man kill

Note that the operating system may re-use PIDs and implementing an equivalent feature in a shell using `kill` and `sleep` would be subject to races whereby the follow-up signal might be sent to a different process that used a recycled PID.

The `--timeout` option can be specified multiple times: the signals are sent sequentially with the specified timeouts. The `--timeout` option can be combined with the `--queue` option.

As an example, the following command sends the signals QUIT, TERM and KILL in sequence and waits for 1000 milliseconds between sending the signals:

```
kill --verbose --timeout 1000 TERM --timeout 1000 KILL --signal QUIT 12345
```

EXIT STATUS

`KILL` has the following exit status values:

0	success
1	failure
64	partial success (when more than one process specified)

NOTES

Although it is possible to specify the TID (thread ID, see `gettid(2)`) of one of the threads in a multithreaded process as the argument of `KILL`, the signal is nevertheless directed to the process (i.e., the entire thread group). In other words, it is not possible to send a signal to an explicitly selected thread in a multithreaded process. The signal will be delivered to an arbitrarily selected thread in the target process that is not blocking the signal. For more details, see `signal(7)` and the description of `CLONE_THREAD` in `clone(2)`.

Various shells provide a builtin `kill` command that is preferred in relation to the `KILL()` executable described by this manual. The easiest way to ensure one is executing the command described in this page is to use the full path when calling the command, for example: `/bin/kill --version`

AUTHORS

Salvatore Valtorta <valtorta@mit.edu>, Karel Zak <kzak@redhat.com>

The original version was taken from BSD 4.4.

SEE ALSO

`hash(1)`, `tcsh(1)`, `sigaction(2)`, `KILL(2)`, `sigqueue(3)`, `signal(7)`

REPORTING BUGS

For bug reports, use the issue tracker at <https://github.com/karelzak/util-linux/issues>.

AVAILABILITY

The `KILL` command is part of the `util-linux` package which can be downloaded from [Linux Kernel Archive](https://www.kernel.org/pub/linux/utils/util-linux/) (<https://www.kernel.org/pub/linux/utils/util-linux/>).

util-linux 2.37.2 2021-06-02 KILL(1)
Manual page `KILL(1)` line 73/117 (END) (press h for help or q to quit)

Краткая характеристика:

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

Ознакомился с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрел практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.