

# Цель работы

---

Ознакомление с файловой системой Linux, её структурой, именами и содержимым 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

## Задание

---

1. Выполните все примеры, приведённые в первой части описания лабораторной работы.
2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
  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.
  2. В домашнем каталоге создайте директорию `~/ski.places`.
  3. Переместите файл `equipment` в каталог `~/ski.places`.
  4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
  5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
  6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
  7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
  8. Создайте и переместите каталог `~/newdir` в каталог `~/ski.places` 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:
  1. `drwxr--r— ... australia`
  2. `drwx--x--x ... play`
  3. `-r-xr--r— ... my_os`
  4. `-rw-rw-r— ... feathers`При необходимости создайте нужные файлы.
4. Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
  1. Просмотрите содержимое файла `/etc/password`.
  2. Скопируйте файл `~/feathers` в файл `~/file.old`.
  3. Переместите файл `~/file.old` в каталог `~/play`.
  4. Скопируйте каталог `~/play` в каталог `~/fun`.
  5. Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`.
  6. Лишите владельца файла `~/feathers` права на чтение.
  7. Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`?
  8. Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`?
  9. Дайте владельцу файла `~/feathers` право на чтение.
  10. Лишите владельца каталога `~/play` права на выполнение.
  11. Перейдите в каталог `~/play`. Что произошло?
  12. Дайте владельцу каталога `~/play` право на выполнение.

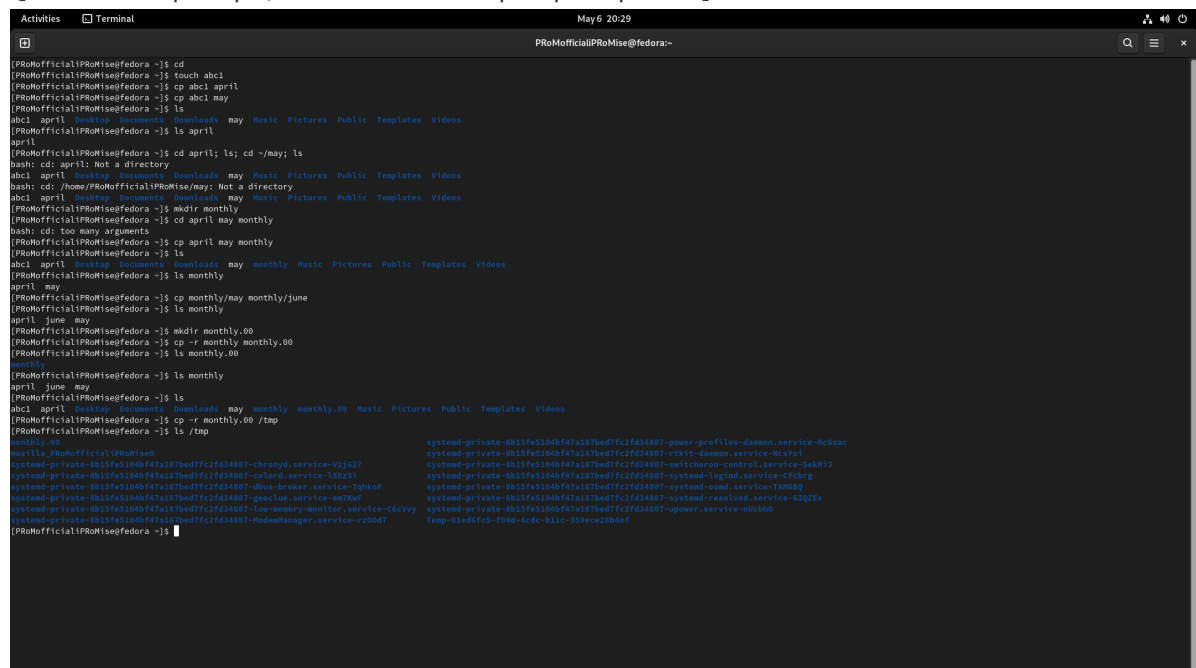
5. Прочитайте man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, приведя примеры.

## Выполнение лабораторной работы

1. Для начала выполним примеры, описанные в первой части описания лабораторной работы (рис. -@fig:001):

1. Скопируем файл ~/abc1 в файл april и в файл may. Для этого создадим файл abc1, используя команду «touch abc1», далее осуществим копирование с помощью команд «cp abc1 april» и «cp abc1 may».
2. Скопируем файлы april и may в каталог monthly, используя команды «mkdir monthly» – для создания каталога monthly и «cp april may monthly» – для копирования.
3. Скопируем файл monthly/may в файл с именем june. Выполним команды «cp monthly/may monthly/june» и «ls monthly» (для просмотра содержимого каталога).
4. Скопируем каталог monthly в каталог monthly.00. Для этого создадим каталог monthly.00 командой «mkdir monthly.00» и осуществим копирование, используя команду «cp -r monthly monthly.00» (команда cp с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами).
5. Скопируем каталог monthly.00 в каталог /tmp, используя команду «cp -r monthly.00 /tmp».

![Выполним примеры, описанные в лабораторной работе]

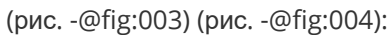


```
ProMoOfficial@ProMise@fedora:~$ cd
ProMoOfficial@ProMise@fedora:~$ touch abc1
ProMoOfficial@ProMise@fedora:~$ cp abc1 april
ProMoOfficial@ProMise@fedora:~$ cp abc1 may
ProMoOfficial@ProMise@fedora:~$ ls
abc1 april desktop Downloads may Music Pictures Public Templates Videos
ProMoOfficial@ProMise@fedora:~$ ls april
ProMoOfficial@ProMise@fedora:~$ cd april; ls; cd -/may; ls
bash: cd: april: not a directory
ProMoOfficial@ProMise@fedora:~$ cd /home/ProMoOfficial@ProMise/may; ls
ProMoOfficial@ProMise@fedora:~$ mkdir monthly
ProMoOfficial@ProMise@fedora:~$ cd april may monthly
bash: cd: too many arguments
ProMoOfficial@ProMise@fedora:~$ cp april may monthly
ProMoOfficial@ProMise@fedora:~$ ls
abc1 april desktop Downloads may monthly Music Pictures Public Templates Videos
ProMoOfficial@ProMise@fedora:~$ ls monthly
april may
ProMoOfficial@ProMise@fedora:~$ cp monthly/may monthly/june
ProMoOfficial@ProMise@fedora:~$ ls monthly
april june may
ProMoOfficial@ProMise@fedora:~$ mkdir monthly.00
ProMoOfficial@ProMise@fedora:~$ cp -r monthly monthly.00
ProMoOfficial@ProMise@fedora:~$ ls monthly.00
monthly
ProMoOfficial@ProMise@fedora:~$ ls monthly
april june may
ProMoOfficial@ProMise@fedora:~$ ls
abc1 april desktop Downloads may monthly monthly.00 Music Pictures Public Templates Videos
ProMoOfficial@ProMise@fedora:~$ cp -r monthly.00 /tmp
ProMoOfficial@ProMise@fedora:~$ ls /tmp
monthly.00
ProMoOfficial@ProMise@fedora:~$
```

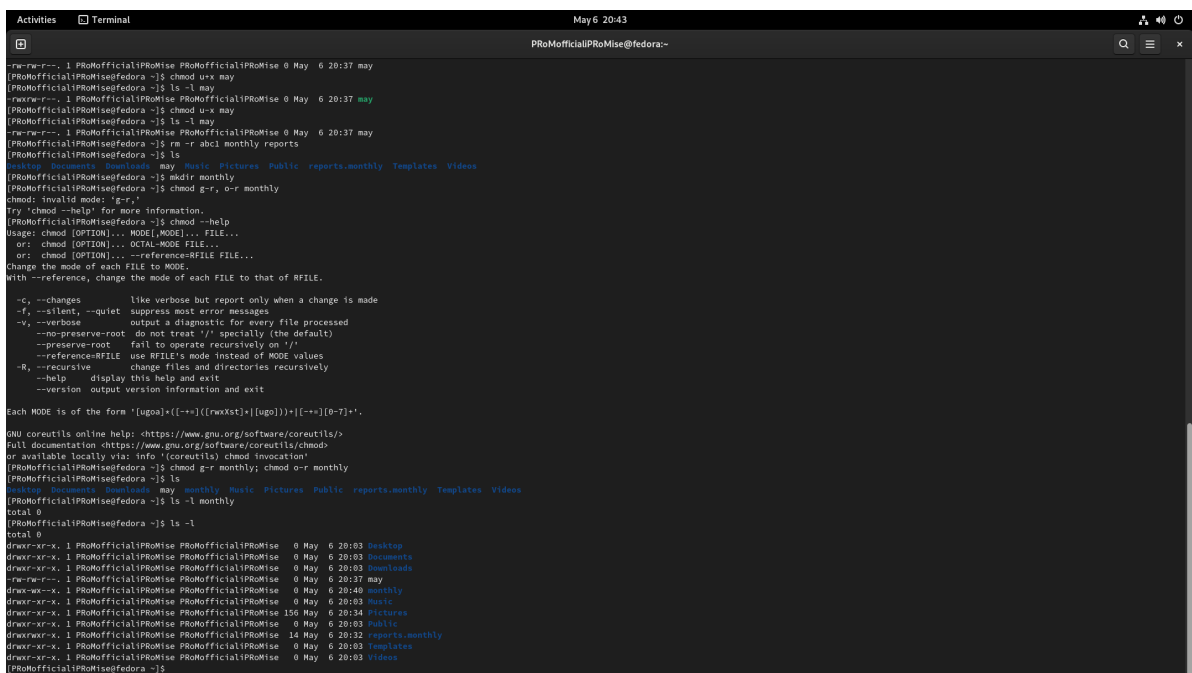
(рис. -@fig:002):

1. Изменим название файла april на july в домашнем каталоге, используя команду «mv april july».
2. Переместим файл july в каталог monthly.00 с помощью команды «mv july monthly.00».
3. Переименуем каталог monthly.00 в monthly.01, используя команду «mv monthly.00 monthly.01».
4. Переместим каталог monthly.01 в каталог reports. Для этого создадим каталог reports с помощью команды «mkdir reports» и выполним перемещение командой «mv monthly.01

Переименуем каталог `reports/monthly.01` в `reports/monthly` командой «`mv reports/monthly.01 reports/monthly`».



1. Создадим файл ~/may с правом выполнения для владельца. Для этого выполним следующие команды: «touch may» (создание файла), «ls -l may» (просмотр сведений о файле), «chmod u+x may» (изменение прав), «ls -l may».
2. Лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may» (изменение прав), «ls -l may» (просмотр сведений о файле).
3. Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly» (создание каталога), «chmod go-r monthly» (изменение прав).
4. Создаем файл ~/abc1 с правом записи для членов группы, используя команды: «touch abc1» (создание файла), «chmod g+w abc1» (изменение прав).



```
Activities Terminal May 6 20:44 PRoMofficialPRoMise@fedora:~  
[PRoMofficialPRoMise@fedora ~]$ chmod u-x may  
[PRoMofficialPRoMise@fedora ~]$ ls -l may  
-rwxrwxr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:37 may  
[PRoMofficialPRoMise@fedora ~]$ rm -r abc1 monthly reports  
[PRoMofficialPRoMise@fedora ~]$ ls  
Desktop Documents Downloads may Music Pictures Public reports.monthly Templates Videos  
[PRoMofficialPRoMise@fedora ~]$ mkdir monthly  
[PRoMofficialPRoMise@fedora ~]$ chmod g-r, o-r monthly  
chmod: invalid mode: 'g-r, o-r'  
Try 'chmod --help' for more information.  
[PRoMofficialPRoMise@fedora ~]$ chmod --help  
usage: chmod [OPTION]... MODE[MODE]... FILE...  
or: chmod [OPTION]... OCTAL-MODE FILE...  
or: chmod [OPTION]... --reference=FILE FILE...  
change the mode of each FILE to MODE  
with --reference, change the mode of each FILE to that of RFILE.  
  
-C, --changes like verbose but report only when a change is made  
-f, --silent, --quiet suppress most error messages  
-v, --verbose output a diagnostic for every file processed  
--no-preserve-root do not treat '/' specially (the default)  
--preserve-root fail to operate recursively on '/'  
--reference=RFILE use RFILE's mode instead of MODE values  
-R, --recursive change files and directories recursively  
--help display this help and exit  
--version output version information and exit  
  
Each MODE is of the form '[ugoa]*([+-]([rwxSt+]*|[ugo]))+([+-]0-7)*'.  
  
GNU coreutils online help: <https://www.gnu.org/software/coreutils/>  
full documentation <https://www.gnu.org/software/coreutils/chmod>  
or available locally via: info '(coreutils) chmod invocation'  
[PRoMofficialPRoMise@fedora ~]$ chmod g-r monthly; chmod o-r monthly  
[PRoMofficialPRoMise@fedora ~]$ ls  
Desktop Documents Downloads may monthly Music Pictures Public reports.monthly Templates Videos  
[PRoMofficialPRoMise@fedora ~]$ ls -l monthly  
total 0  
[PRoMofficialPRoMise@fedora ~]$ ls -l  
total 0  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:03 Desktop  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:03 Documents  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:03 Downloads  
-rwxrwxr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:37 may  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:40 monthly  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:03 Music  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 150 May 6 20:34 Pictures  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:03 Public  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 14 May 6 20:32 reports.monthly  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:03 Templates  
drwxr-xr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:03 Videos  
[PRoMofficialPRoMise@fedora ~]$ touch abc1  
[PRoMofficialPRoMise@fedora ~]$ chmod g+w abc1  
[PRoMofficialPRoMise@fedora ~]$ ls -l abc1  
-rwxrwxr-x. 1 PRoMofficialPRoMise PRoMofficialPRoMise 0 May 6 20:44 abc1  
[PRoMofficialPRoMise@fedora ~]$
```

2. Выполняем следующие действия, отображенные на (рис. -@fig:005) (рис. -@fig:006) (рис. -@fig:007):

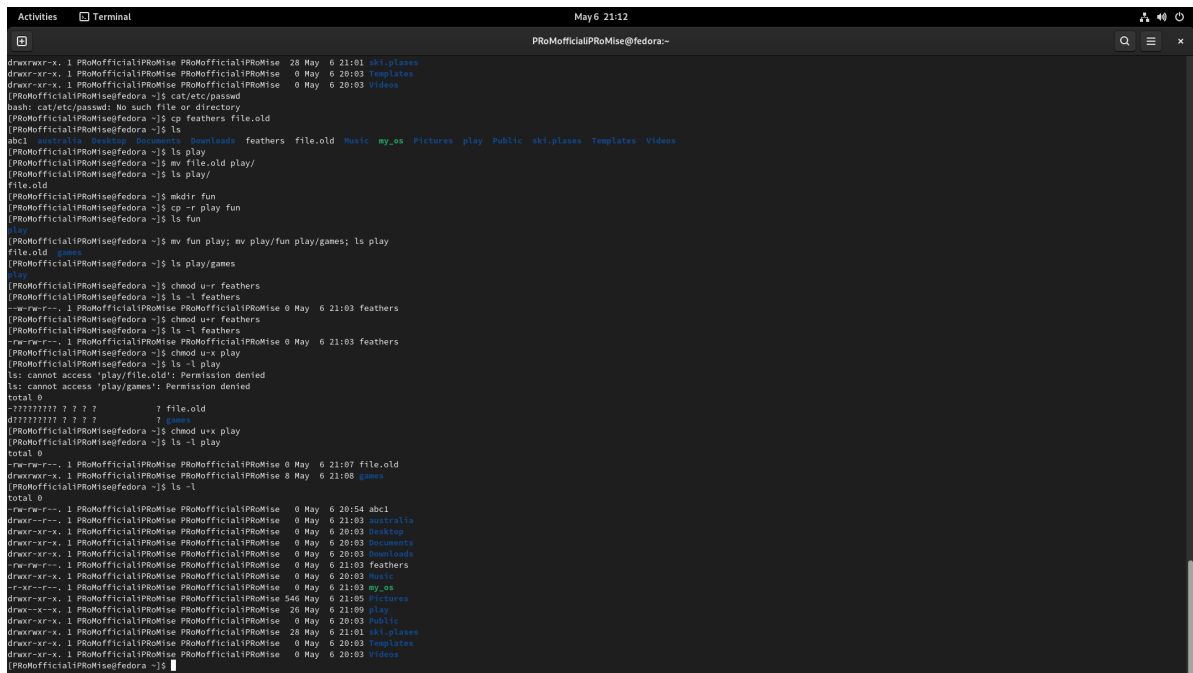
1. Копируем файл /usr/include/aio.h (т.к. у меня нет каталога /usr/include/sys/, то беру произвольный файл из каталога /usr/include/) в домашний каталог (команда «cp /usr/include/aio.h ~») и называем его equipment (команда «mv aio.h equipment»).
2. В домашнем каталоге создаем директорию ~/ski.places (команда «mkdir ski.places»).
3. Перемещаем файл equipment в каталог ~/ski.places (команда «mv equipment ski.places»).
4. Переименовываем файл ~/ski.places/equipment в ~/ski.places/equiplist (команда «mv ski.places/equipment ski.places/equiplist»).
5. Создаем в домашнем каталоге файл abc1 (команда «touch abc1») и копируем его в каталог ~/ski.places (команда «cp abc1 ski.places»), называем его equiplist2 (команда «mv ski.places/abc1 ski.places/equiplist2»).
6. Создаем каталог с именем equipment в каталоге ~/ski.places (команда «mkdir ski.places/equipment»).
7. Перемещаем файлы ~/ski.places/equiplist и equiplist2 в каталог ~/ski.places/equipment (команда «mv ski.places/equiplist ski.places/equipment»).
8. Создаем (команда «mkdir newdir») и перемещаем каталог ~/newdir в каталог ~/ski.places (команда «mv newdir ski.places») и называем его plans (команда «mv ski.places/newdir ski.places/plans»).



3. Определяем опции команды `chmod`, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем необходимые файлы, используя команды: «`mkdir australia play`», «`touch my_os feathers`».

- `drwxr--r-- ... australia`: команда «`chmod 744 australia`» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение)
- `drwx--x--x ... play`: команда «`chmod 711 play`» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение)
- `-r-xr--r-- ... my_os`: команда «`chmod 544 my_os`» (это файл, владелец имеет право на чтение и выполнение, группа владельца и остальные – только чтение)
- `-rw-rw-r-- ... feathers`: команда «`chmod 664 feathers`» (это файл, владелец и группа владельца имеют право на чтение и запись, остальные – только чтение)

Командой «`ls -l`» проверяем правильность выполненных действий (рис. -@fig:008).



```
drwxrwxr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 28 May 6 21:01 ski_places
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 Templates
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 Videos
[PROmoOfficialPRoMise@fedora ~]$ cat /etc/passwd
bash: cat/etc/passwd: No such file or directory
[PROmoOfficialPRoMise@fedora ~]$ cp feathers file.old
[PROmoOfficialPRoMise@fedora ~]$ ls
abcl  australia  desktop  documents  downloads  feathers  file.old  Music  my_os  Pictures  play  Public  ski_places  Templates  Videos
[PROmoOfficialPRoMise@fedora ~]$ ls play
[PROmoOfficialPRoMise@fedora ~]$ mv file.old play/
[PROmoOfficialPRoMise@fedora ~]$ ls play/
file.old
[PROmoOfficialPRoMise@fedora ~]$ mkdir fun
[PROmoOfficialPRoMise@fedora ~]$ cp -r play fun
[PROmoOfficialPRoMise@fedora ~]$ ls fun
play
[PROmoOfficialPRoMise@fedora ~]$ mv fun play; mv play/fun play/games; ls play
file.old  games
[PROmoOfficialPRoMise@fedora ~]$ ls play/games
play
[PROmoOfficialPRoMise@fedora ~]$ chmod u-r feathers
[PROmoOfficialPRoMise@fedora ~]$ ls -l feathers
-rw-rw-r--. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 0 May 6 21:03 feathers
[PROmoOfficialPRoMise@fedora ~]$ chmod u-r feathers
[PROmoOfficialPRoMise@fedora ~]$ ls -l feathers
-rw-rw-r--. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 0 May 6 21:03 feathers
[PROmoOfficialPRoMise@fedora ~]$ chmod u-x play
[PROmoOfficialPRoMise@fedora ~]$ ls -l play
ls: cannot access 'play/file.old': Permission denied
ls: cannot access 'play/games': Permission denied
total 0
????????? ? ? ? ? ? file.old
????????? ? ? ? ? ? games
[PROmoOfficialPRoMise@fedora ~]$ chmod u-x play
[PROmoOfficialPRoMise@fedora ~]$ ls -l play
total 0
-rw-rw-r--. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 0 May 6 21:07 file.old
drwxrwxr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 8 May 6 21:08 games
[PROmoOfficialPRoMise@fedora ~]$ ls -l
total 0
-rw-rw-r--. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 0 May 6 20:54 abcl
drwxr--r--. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 21:03 australia
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 desktop
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 documents
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 downloads
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 21:03 feathers
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 Music
-r-xr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 21:03 my_os
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 544 May 6 21:03 Pictures
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 20 May 6 21:09 play
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 Public
drwxrwxr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise 28 May 6 21:03 ski_places
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 Templates
drwxr-xr-x. 1 PROmoOfficialPRoMise PROmoOfficialPRoMise  6 May 6 20:03 Videos
[PROmoOfficialPRoMise@fedora ~]$
```

4. Выполняем следующие действия, отображенные на (рис. -@fig:009) (рис. -@fig:010) (рис. -@fig:011):

1. Просмотрим содержимое файла `/etc/passwd` (команда «`cat /etc/passwd`»).
2. Копируем файл `~/feathers` в файл `~/file.old` (команда «`cp feathers file.old`»).
3. Переместим файл `~/file.old` в каталог `~/play` (команда «`mv file.ord play`»).
4. Скопируем каталог `~/play` в каталог `~/fun` (команда «`cp -r play fun`»).
5. Переместим каталог `~/fun` в каталог `~/play` (команда «`mv fun play`») и назовем его `games` (команда «`mv play/fun play/games`»).
6. Лишим владельца файла `~/feathers` права на чтение (команда «`chmod u-r feathers`»).
7. Если мы попытаемся просмотреть файл `~/feathers` командой `cat`, то получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на чтение данного файла.
8. Если мы попытаемся скопировать файл `~/feathers`, например, в каталог `monthly`, то получим отказ в доступе, по причине, описанной в предыдущем пункте.
9. Дадим владельцу файла `~/feathers` право на чтение (команда «`chmod u+r feathers`»).
10. Лишим владельца каталога `~/play` права на выполнение (команда «`chmod u-x play`»).
11. Перейдем в каталог `~/play` (команда «`cd play`»). Получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на выполнение данного каталога.
12. Дадим владельцу каталога `~/play` право на выполнение (команда «`chmod u+x play`»).

```
Activities Terminal May 6 21:13
PRoMofficialPRoMise@fedora:~$ man mount
NAME
    mount - mount a filesystem
SYNOPSIS
    mount [-h] [-V]
    mount [-l] [-t fstype]
    mount -a [-fnrsvw] [-t fstype] [-O optlist]
    mount [-fnrsvw] [-o options] device[mountpoint]
    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint
    mount --bind|--rbind|--move olddir newdir
    mount --make-shared|slave|private|unbindable|shared|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 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.
    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 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 policy to control this behavior. All 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(8), especially in your scripts. Note that control characters in the mountpoint name are replaced with '?'.
    The following command lists all mounted filesystems (of type type):
        mount [-l] [-t type]
    The option -l adds labels to this listing. See below.
Manual page mount(8) line 1 (press h for help or q to quit)
```

5. Используя команды «man mount», «man fsck», «man mkfs», «man kill», получим информацию о соответствующих командах.

## Команда mount

Предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого дерева обозначается как /. Физически файлы могут располагаться на различных устройствах. Команда mount служит для подключения файловых систем разных устройств к этому большому дереву.

Наиболее часто встречающаяся форма команды mount выглядит следующим образом:

«mount -t vfstype device dir»

Такая команда предлагает ядру смонтировать (подключить) файловую систему указанного типа vfstype, расположенную на устройстве device, к заданному каталогу dir, который часто называют точкой монтирования.

```
Activities Terminal May 6 21:14
PRoMofficialPRoMise@fedora:~$ man fsck
NAME
    fsck - check and repair a Linux filesystem
SYNOPSIS
    fsck [-lsAVTnmp] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-options]
DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystem can be a device name (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=8868abf6-88c5-4a83-98b8-bfc240577bd 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 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.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.
OPTIONS
    -l      Create an exclusive flock(2) lock file (/run/fsck/cdiskname.lock) for whole-disk device. This option can be used with one device only (this means that -A and -l are mutually exclusive). This option is recommended when more fsck instances are executed in the same time. The option is ignored when used for multiple devices or for non-rotating disks. fsck does not lock underlying devices when executed to check stacked devices (e.g. MD or DM) - this feature is not implemented yet.
    -r [fd]
Manual page fsck(8) line 1 (press h for help or q to quit)
```

## Команда fsck

Это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Он использует программы, специфичные для типа файловой системы, которую он проверяет.

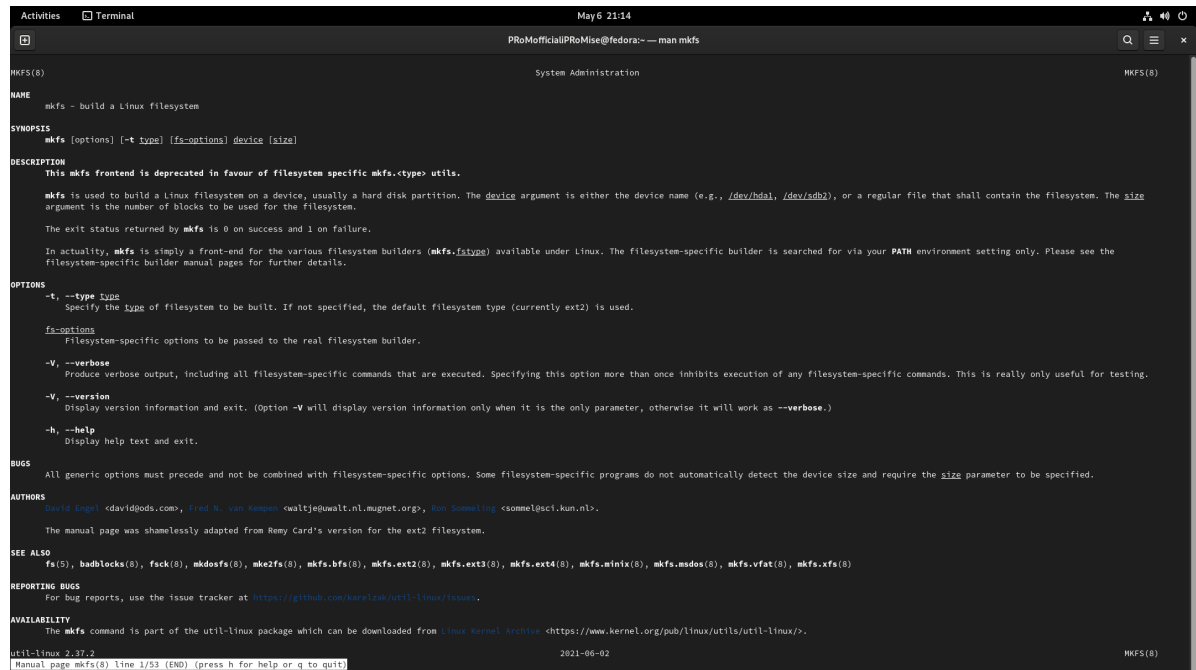
У команды fsck следующий синтаксис:

fsck параметр -- параметры ФС <файловая система> . . .

Например, если нужно восстановить («починить») файловую систему на некотором устройстве /dev/sdb2, следует воспользоваться командой:

«sudo fsck -y /dev/sdb2»

Опция -y необходима, т. к. при её отсутствии придётся слишком часто давать подтверждение.



```
Activities Terminal May6 21:14
PRoMofficialPRoMise@fedora:~$ man mkfs

mkfs(8) System Administration mkfs(8)

NAME
mkfs - build a Linux filesystem

SYNOPSIS
mkfs [options] [-t type] [fs-options] device [size]

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

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

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

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@dods.com>, Fred N. van Kempen <waltje@u.walt.nl.mugnet.org>, Ron Sommeling <sonnel@sci.kun.nl>.

The manual page was shamelessly adapted from Remy Card's version for the ext2 filesystem.

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

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

AVAILABILITY
The mkfs 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 2021-06-02
Manual page mkfs(8) line 1/53 (END) (press h for help or q to quit)
```

## Команда mkfs:

Создаёт новую файловую систему Linux.

Имеет следующий синтаксис:

mkfs -V -t fstype fs-options filesys blocks

mkfs используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента filesys для файловой системы может выступать или название устройства (например, /dev/hda1, /dev/sdb2) или точка монтирования (например, /, /usr, /home).

Аргументом blocks указывается количество блоков, которые выделяются для использования этой файловой системой.

По окончании работы mkfs возвращает 0 - в случае успеха, а 1 - при неудачной операции.

Например, команда «mkfs -t ext2 /dev/hdb1» создаёт файловую систему типа ext2 в разделе /dev/hdb1 (второй жёсткий диск).



```
Activities Terminal May 6 21:15
ProMofficialProMise@fedora:~$ man kill
NAME
kill - terminate a process
SYNOPSIS
kill [-signal=-s signal][-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|name...
kill -l [number] | -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:
n
where n is larger than 0. The process with PID n is signaled.
0
All processes in the current process group are signaled.
-1
All processes with a PID larger than 1 are signaled.
-n
where n is larger than 1. All processes in process group n 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
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.
Manual page kill(1) line 1 (press h for help or q to quit)
```

Команда kill (рис. -@fig:015):

Посылает сигнал процессу или выводит список допустимых сигналов.

Имеет следующий синтаксис:

kill опции PID, где PID – это PID (числовой идентификатор) процесса или несколько PID процессов, если требуется послать сигнал сразу нескольким процессам.

Например, команда «kill -KILL 3121» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс.

```
Activities Terminal May 6 21:15
ProMofficialProMise@fedora:~$
[ProMofficialProMise@fedora ~]$ cp feathers file.old
[ProMofficialProMise@fedora ~]$ ls
abc1  Downloads  file.old  Music  my_as  Pictures  play  Public  ski-places  Templates  Videos
[ProMofficialProMise@fedora ~]$ ls play
[ProMofficialProMise@fedora ~]$ mv file.old play/
[ProMofficialProMise@fedora ~]$ ls play/
file.old
[ProMofficialProMise@fedora ~]$ mkdir fun
[ProMofficialProMise@fedora ~]$ cp -r play fun
[ProMofficialProMise@fedora ~]$ ls fun
play
[ProMofficialProMise@fedora ~]$ mv fun play; mv play/fun play/games; ls play
file.old  games
[ProMofficialProMise@fedora ~]$ ls play/games
play
[ProMofficialProMise@fedora ~]$ chmod u-r feathers
[ProMofficialProMise@fedora ~]$ ls -l feathers
-rw-rw-r--. 1 ProMofficialProMise ProMofficialProMise 0 May 6 21:03 feathers
[ProMofficialProMise@fedora ~]$ chmod u+r feathers
[ProMofficialProMise@fedora ~]$ ls -l feathers
-rw-rw-r--. 1 ProMofficialProMise ProMofficialProMise 0 May 6 21:03 feathers
[ProMofficialProMise@fedora ~]$ chmod u-x play
[ProMofficialProMise@fedora ~]$ ls -l play
ls: cannot access 'play/file.old': Permission denied
ls: cannot access 'play/games': Permission denied
total 0
-rw-rw-r--. 1 ProMofficialProMise ProMofficialProMise 0 May 6 21:07 file.old
drwxrwxr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 21:08 games
[ProMofficialProMise@fedora ~]$ ls -l
total 0
-rw-rw-r--. 1 ProMofficialProMise ProMofficialProMise 0 May 6 20:54 abc1
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 21:03 australia
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 20:03 desktop
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 20:03 documents
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 20:03 Downloads
-rw-rw-r--. 1 ProMofficialProMise ProMofficialProMise 0 May 6 21:03 feathers
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 20:03 home
-rwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 21:03 my_as
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 540 May 6 21:05 Pictures
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 20 May 6 21:09 play
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 20:03 public
drwxrwxr-x. 1 ProMofficialProMise ProMofficialProMise 28 May 6 21:01 ski-places
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 20:03 templates
drwxr-xr-x. 1 ProMofficialProMise ProMofficialProMise 0 May 6 20:03 Videos
[ProMofficialProMise@fedora ~]$ man mount
[ProMofficialProMise@fedora ~]$ man fsck
[ProMofficialProMise@fedora ~]$ man mkfs
[ProMofficialProMise@fedora ~]$ man kill
[ProMofficialProMise@fedora ~]$
[ProMofficialProMise@fedora ~]$
```

## Контрольные вопросы

1. Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th». На моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. devtmpfs позволяет ядру создать экземпляр tmpfs с именем devtmpfs при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором / минором будет предоставлять узел устройства в devtmpfs. devtmpfs монтируется на /dev и содержит специальные файлы устройств для всех устройств. tmpfs – временное файловое хранилище во многих Unix-подобных ОС.

Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначена для быстрого и ненадёжного хранения временных данных. Хорошо подходит для /tmp и массовой сборки пакетов/образов. Предполагает наличие достаточного объёма виртуальной памяти. Файловая система tmpfs предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска. ext4 – имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «extent file system», который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (delayed allocation – delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных.

Характеристики:

- максимальный размер файла: 16 TB;
- максимальный размер раздела: 16 TB;
- максимальный размер имени файла: 255 символов.

Рекомендации по использованию:

- наилучший выбор для SSD;
- наилучшая производительность по сравнению с предыдущими Ext-системами;
- она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3.

ISO 9660 – стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

2. Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел – / (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, "примонтированных" к определенным каталогам.

- / – корень

Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. 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.

- /BIN – бинарные файлы пользователя

Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления.

- /sbin – 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.

- /ETC – 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.
- /DEV – 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/. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов.
- /PROC – информация о процессах  
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.
- /VAR – 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.
- /TMP – временные файлы  
В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию.
- /USR – программы пользователя  
Это самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию.
- /HOME – домашняя папка  
В этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т.д.
- /BOOT – файлы загрузчика  
Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящиеся в каталоге /boot/grub.
- /LIB – системные библиотеки  
Содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах /bin и /sbin.
- /OPT – дополнительные программы  
В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями.
- /MNT – монтирование  
В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы.
- /MEDIA – съемные носители  
В этот каталог система монтирует все подключаемые внешние накопители –USB флешки, оптические диски и другие носители информации.
- /SRV – сервер  
В этом каталоге содержатся файлы серверов и сервисов.
- /RUN - процессы  
Каталог, содержащий PID файлы процессов, похожий на /var/run, но в отличие от него, он размещен в TMPFS, а поэтому после перезагрузки все файлы теряются.

3. Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount.
4. Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду fsck.
5. Файловую систему можно создать, используя команду mkfs. Ее краткое описание дано в пункте 5 в ходе выполнения заданий лабораторной работы.
6. Для просмотра текстовых файлов существуют следующие команды:

- cat

Задача команды cat очень проста – она читает данные из файла или стандартного ввода и выводит их на экран.

Синтаксис утилиты:

cat опции файл1 файл2 ...

Основные опции:

- b – нумеровать только непустые строки
- E – показывать символ \$ в конце каждой строки
- n – нумеровать все строки
- s – удалять пустые повторяющиеся строки
- T – отображать табуляции в виде ^I
- h – отобразить справку
- v – версия утилиты

- nl

Команда nl действует аналогично команде cat, но выводит еще и номера строк в столбце слева.

- less

Существенно более развитая команда для пролистывания текста.

При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста.

Синтаксис аналогичный синтаксису команды cat.

Некоторые опции:

- g – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения)
- N – показывать номера строк

- head

Команда head выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита.

Синтаксис аналогичный синтаксису команды cat.

Основные опции:

- c (--bytes) – позволяет задавать количество текста не в строках, а в байтах
- n (--lines) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию
- q (--quiet, --silent) – выводит только текст, не добавляя к нему название файла
- v (--verbose) – перед текстом выводит название файла
- z (--zero-terminated) – символы перехода на новую строку заменяет символами завершения строк

- tail

Эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме.

Синтаксис аналогичный синтаксису команды cat.

Основные опции:

- c – выводить указанное количество байт с конца файла
- f – обновлять информацию по мере появления новых строк в файле
- n – выводить указанное количество строк из конца файла
- pid – используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс
- q – не выводить имена файлов
- retry – повторять попытки открыть файл, если он недоступен
- v – выводить подробную информацию о файле

## 7. Утилита `cp` позволяет полностью копировать файлы и директории.

Синтаксис:

`cp` опции файл-источник файл-приемник

После выполнения команды файл-источник будет полностью перенесен в файл-приемник.

Если в конце указан слэш, файл будет записан в заданную директорию с оригинальным именем.

Основные опции:

- attributes-only – не копировать содержимое файла, а только флаги доступа и владельца
- f, --force – перезаписывать существующие файлы
- i, --interactive – спрашивать, нужно ли перезаписывать существующие файлы
- L – копировать не символические ссылки, а то, на что они указывают -P – не перезаписывать существующие файлы
- P – не следовать символическим ссылкам
- r – копировать папку Linux рекурсивно
- s – не выполнять копирование файлов в Linux, а создавать символические ссылки
- u – скопировать файл, только если он был изменён
- x – не выходить за пределы этой файловой системы
- p – сохранять владельца, временные метки и флаги доступа при копировании
- t – считать файл-приемник директорией и копировать файл-источник в эту директорию

## 8. Команда `mv` используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий.

Синтаксис:

`mv` -опции старый*файл* новый*файл*

Основные опции:

- help – выводит на экран официальную документацию об утилите
- version – отображает версию `mv`
- b – создает копию файлов, которые были перемещены или перезаписаны
- f – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла
- i – наоборот, будет спрашивать разрешение у владельца
- n – отключает перезапись уже существующих объектов
- strip-trailing-slashes — удаляет завершающий символ / у файла при его наличии
- t директория — перемещает все файлы в указанную директорию
- u – осуществляет перемещение только в том случае, если исходный файл новее объекта назначения
- v – отображает сведения о каждом элементе во время обработки команды

Команда `rename` также предназначена, чтобы переименовать файл.

Синтаксис:

`rename` опции старое*имя* новое*имя* файлы

Основные опции:

-v – вывести список обработанных файлов

-n – тестовый режим, на самом деле никакие действия выполнены не будут

-f – принудительно перезаписывать существующие файлы

9. Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации.

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

Синтаксис команды:

`chmod режим имя_файла`

Режим имеет следующие компоненты структуры и способ записи:

- = установить право
- o лишить права
- o дать право
- r чтение
- w запись
- x выполнение
- u (user) владелец файла
- g (group) группа, к которой принадлежит владелец файла
- o (others) все остальные

## Выводы

---

В ходе выполнения данной лабораторной работы я ознакомился с файловой системой Linux, её структурой, именами и содержанием каталогов, получил навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.