

# Отчёт по лабораторной работе №5

дисциплина: Операционные системы

Студент: Махорин Иван Сергеевич

# Содержание

Цель работы	5
Задание	6
Выполнение лабораторной работы	8
Контрольные вопросы	18
Выводы	23

## Список иллюстраций

0.1	Копирование файла <code>io.h</code> под названием <code>equipment</code> . Проверка. . . . .	8
0.2	Создание директории <code>ski.places</code> . . . . .	9
0.3	Перемещение файла <code>equipment</code> в каталог <code>ski.places</code> . . . . .	9
0.4	Переименование файла <code>equipment</code> в <code>equiplist</code> . . . . .	9
0.5	Создание файла <code>abc1</code> и копирование его в каталог <code>ski.places</code> под названием <code>equiplist2</code> . . . . .	9
0.6	Создание каталога <code>equipment</code> в каталоге <code>ski.places</code> . . . . .	10
0.7	Перемещение файлов в подкаталог <code>equipment</code> . . . . .	10
0.8	Создание каталога <code>newdir</code> и перемещение его в каталог <code>ski.places</code> под названием <code>plans</code> . . . . .	10
0.9	Создание каталогов и файлов . . . . .	11
0.10	Присваивание прав доступа . . . . .	11
0.11	Просмотр содержимого файла <code>passwd</code> . . . . .	12
0.12	Копирование файла <code>feathers</code> в <code>file.old</code> . . . . .	12
0.13	Перемещение файла <code>file.old</code> в каталог <code>play</code> . . . . .	12
0.14	Копирование каталога <code>play</code> в каталог <code>fun</code> . . . . .	13
0.15	Перемещение каталога <code>fun</code> с изменением название на <code>games</code> . . . . .	13
0.16	Лишение права на чтение. Попытки чтения и копирования файла. Возвращение права на чтение . . . . .	13
0.17	Лишение права на выполнение. Попытка перехода в каталог. Возвращение права на выполнение . . . . .	14
0.18	<code>man mount</code> . . . . .	14
0.19	<code>man fsck</code> . . . . .	15
0.20	<code>man mkfs</code> . . . . .	16
0.21	<code>man kill</code> . . . . .	17

## Список таблиц

## 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.

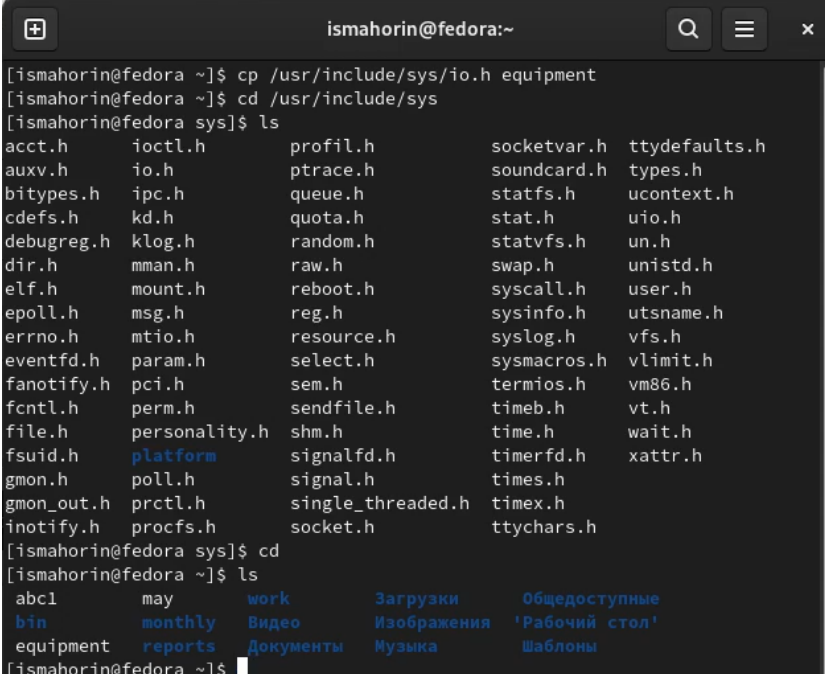
# Задание

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:
  - 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.
  - В домашнем каталоге создайте директорию `~/ski.places`.
  - Переместите файл `equipment` в каталог `~/ski.places`.
  - Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
  - Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
  - Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
  - Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
  - 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: `drwxr-r- australia drwx-x-x play -r-xr-r- my_os -rw-rw-r- feathers` При необходимости создайте нужные файлы.
4. Прodelайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:

- Просмотрите содержимое файла `/etc/password`.
  - Скопируйте файл `~/feathers` в файл `~/file.old`.
  - Переместите файл `~/file.old` в каталог `~/play`.
  - Скопируйте каталог `~/play` в каталог `~/fun`.
  - Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`.
  - Лишите владельца файла `~/feathers` права на чтение.
  - Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`?
  - Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`?
  - Дайте владельцу файла `~/feathers` право на чтение.
  - Лишите владельца каталога `~/play` права на выполнение.
  - Перейдите в каталог `~/play`. Что произошло?
  - Дайте владельцу каталога `~/play` право на выполнение.
5. Прочитайте ман по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.

# Выполнение лабораторной работы

Скопируем файл `io.h` в домашний каталог и назовём его `equipment`. Для этого воспользуемся командой `cp` и укажем путь к нашему файлу. Выполним проверку командой `ls` (рис. [-@fig:001]).



```
ismahorin@fedora:~  
[ismahorin@fedora ~]$ cp /usr/include/sys/io.h equipment  
[ismahorin@fedora ~]$ cd /usr/include/sys  
[ismahorin@fedora sys]$ ls  
acct.h      ioctl.h      profil.h      socketvar.h   ttydefaults.h  
auxv.h      io.h         ptrace.h      soundcard.h   types.h  
bitypes.h   ipc.h        queue.h       statfs.h      ucontext.h  
cdefs.h     kd.h         quota.h       stat.h         uio.h  
debugreg.h  klog.h       random.h      statvfs.h     un.h  
dir.h       mman.h       raw.h         swap.h        unistd.h  
elf.h       mount.h      reboot.h      syscall.h     user.h  
epoll.h     msg.h        reg.h         sysinfo.h     utsname.h  
errno.h     mtio.h       resource.h    syslog.h      vfs.h  
eventfd.h   param.h      select.h      sysmacros.h   vlimit.h  
fanotify.h  pci.h        sem.h         termios.h     vm86.h  
fcntl.h     perm.h       sendfile.h    timeb.h       vt.h  
file.h      personality.h shm.h          time.h        wait.h  
fsuid.h     platform     signalfd.h    timerfd.h     xattr.h  
gmon.h      poll.h       signal.h      times.h  
gmon_out.h  prctl.h     single_threaded.h timex.h  
inotify.h   procfs.h     socket.h      ttychars.h  
[ismahorin@fedora sys]$ cd  
[ismahorin@fedora ~]$ ls  
abc1      may      work      Загрузки      Общедоступные  
bin        monthly  Видео      Изображения   'Рабочий стол'  
equipment  reports  Документы  Музыка         Шаблоны  
[ismahorin@fedora ~]$
```

Рис. 0.1: Копирование файла `io.h` под названием `equipment`. Проверка.

В домашнем каталоге командой `mkdir` создаём директорию `ski.places`. Выполняем проверку (рис. [-@fig:002]).



```
[ismahorin@fedora ~]$ mkdir ski.places
[ismahorin@fedora ~]$ ls
abc1      may      ski.places  Документы  Музыка      Шаблоны
bin       monthly  work       Загрузки   Общедоступные
equipment reports  Видео      Изображения 'Рабочий стол'
```

Рис. 0.2: Создание директории ski.places

Перемещаем файл equipment в каталог ski.places командой mv. Выполняем проверку (рис. [-@fig:003]).

```
[ismahorin@fedora ~]$ mv equipment ski.places
[ismahorin@fedora ~]$ ls
abc1  monthly  work      Загрузки  Общедоступные
bin   reports  Видео     Изображения 'Рабочий стол'
may   ski.places  Документы Музыка     Шаблоны
[ismahorin@fedora ~]$ cd ski.places
[ismahorin@fedora ski.places]$ ls
equipment
```

Рис. 0.3: Перемещение файла equipment в каталог ski.places

Переименуем файл equipment, находящийся в каталоге ski.places в equiplist с помощью команды mv. Выполним проверку (рис. [-@fig:004]).

```
[ismahorin@fedora ski.places]$ cd
[ismahorin@fedora ~]$ mv ski.places/equipment ski.places/equiplist
[ismahorin@fedora ~]$ cd ski.places
[ismahorin@fedora ski.places]$ ls
equiplist
```

Рис. 0.4: Переименование файла equipment в equiplist

Создаём в домашнем каталоге файл abc1 командой touch и копируем его в каталог ski.places под названием equiplist2. Выполняем проверку (рис. [-@fig:005]).

```
[ismahorin@fedora ski.places]$ cd
[ismahorin@fedora ~]$ touch abc1
[ismahorin@fedora ~]$ cp abc1 ski.places
[ismahorin@fedora ~]$ mv ski.places/abc1 ski.places/equiplist2
[ismahorin@fedora ~]$ cd ski.places
[ismahorin@fedora ski.places]$ ls
equiplist equiplist2
```

Рис. 0.5: Создание файла abc1 и копирование его в каталог ski.places под названием equiplist2

Создаём каталог с именем equipment в каталоге ski.plases (рис. [-@fig:006]).

```
[ismahorin@fedora ski.plases]$ cd
[ismahorin@fedora ~]$ cd ski.plases
[ismahorin@fedora ski.plases]$ mkdir equipment
[ismahorin@fedora ski.plases]$ ls
equiplist  equiplist2  equipment
```

Рис. 0.6: Создание каталога equipment в каталоге ski.plases

Перемещаем файлы equiplist и equiplist2 из каталога ski.plases в подкаталог equipment (рис. [-@fig:007]).

```
[ismahorin@fedora ski.plases]$ cd
[ismahorin@fedora ~]$ mv ski.plases/equiplist equipment
[ismahorin@fedora ~]$ mv ski.plases/equiplist2 equipment
[ismahorin@fedora ~]$ ls
abcl    may    ski.plases  Документы  Музыка  Шаблоны
bin     monthly work        Загрузки  Общедоступные
equipment reports Видео      Изображения 'Рабочий стол'
[ismahorin@fedora ~]$ cd
[ismahorin@fedora ~]$ cd ski.plases
[ismahorin@fedora ski.plases]$ ls
equipment
```

Рис. 0.7: Перемещение файлов в подкаталог equipment

Создаём каталог newdir. Далее перемещаем его в каталог ski.plases под названием plans (рис. [-@fig:008]).

```
[ismahorin@fedora equipment]$ cd
[ismahorin@fedora ~]$ mkdir newdir
[ismahorin@fedora ~]$ ls
abcl    monthly    work    Изображения  Шаблоны
bin     newdir     Видео   Музыка
equipment reports    Документы  Общедоступные
may     ski.plases Загрузки  'Рабочий стол'
[ismahorin@fedora ~]$ mv newdir ski.plases
[ismahorin@fedora ~]$ mv ski.plases/newdir ski.plases/plans
[ismahorin@fedora ~]$ cd ski.plases
[ismahorin@fedora ski.plases]$ ls
equipment  plans
```

Рис. 0.8: Создание каталога newdir и перемещение его в каталог ski.plases под названием plans

Создаём 2 каталога (australia и play) и 2 файла (my\_os и feathers) (рис. [-@fig:009]).

```
[ismahorin@fedora ski.plases]$ cd
[ismahorin@fedora ~]$ mkdir australia
[ismahorin@fedora ~]$ mkdir play
[ismahorin@fedora ~]$ touch my_os
[ismahorin@fedora ~]$ touch feathers
[ismahorin@fedora ~]$ ls
abc1      feathers  play      Видео      Музыка
australia may       reports   Документы  Общедоступные
bin       monthly  ski.plases Загрузки   'Рабочий стол'
equipment my_os     work      Изображения Шаблоны
```

Рис. 0.9: Создание каталогов и файлов

Далее присвоим каждому из каталогов и файлов определённые права доступа (рис. [-@fig:010]).

```
[ismahorin@fedora ~]$ chmod 744 australia
[ismahorin@fedora ~]$ chmod 711 play
[ismahorin@fedora ~]$ chmod 544 my_os
[ismahorin@fedora ~]$ chmod 664 feathers
[ismahorin@fedora ~]$ ls -l
итого 4
-rw-rw-r--. 1 ismahorin ismahorin 0 мая 1 13:22 abc1
drwxr--r--. 1 ismahorin ismahorin 0 мая 1 13:41 australia
drwxr-xr-x. 1 ismahorin ismahorin 8 апр 28 21:01 bin
-rw-rw-r--. 1 ismahorin ismahorin 0 мая 1 13:23 equipment
-rw-rw-r--. 1 ismahorin ismahorin 0 мая 1 13:41 feathers
-rw-rw-r--. 1 ismahorin ismahorin 0 апр 30 17:18 may
drwxrwxr-x. 1 ismahorin ismahorin 24 апр 30 17:11 monthly
-r-xr--r--. 1 ismahorin ismahorin 0 мая 1 13:41 my_os
drwx--x--x. 1 ismahorin ismahorin 0 мая 1 13:41 play
drwxrwxr-x. 1 ismahorin ismahorin 14 апр 30 17:18 reports
drwxrwxr-x. 1 ismahorin ismahorin 28 мая 1 13:33 ski.plases
drwxrwxr-x. 1 ismahorin ismahorin 62 апр 28 21:55 work
drwxr-xr-x. 1 ismahorin ismahorin 0 апр 21 00:22 Видео
drwxr-xr-x. 1 ismahorin ismahorin 0 апр 21 00:22 Документы
drwxr-xr-x. 1 ismahorin ismahorin 392 апр 28 21:00 Загрузки
drwxr-xr-x. 1 ismahorin ismahorin 0 апр 27 16:51 Изображения
drwxr-xr-x. 1 ismahorin ismahorin 0 апр 21 00:22 Музыка
drwxr-xr-x. 1 ismahorin ismahorin 0 апр 21 00:22 Общедоступные
```

Рис. 0.10: Присваивание прав доступа

Просматриваем содержимое файла passwd с помощью команды cat (рис. [-@fig:011]).

```
[ismahorin@fedora etc]$ cd
[ismahorin@fedora ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:usr/sbin/nologin
systemd-oom:x:999:999:systemd Userspace OOM Killer:/:usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:usr/sbin/nologin
systemd-timesync:x:998:998:systemd Time Synchronization:/:usr/sbin/nologin
systemd-coredump:x:997:997:systemd Core Dumper:/:usr/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin
```

Рис. 0.11: Просмотр содержимого файла passwd

Скопируем файл feathers в file.old (рис. [-@fig:012]).

```
[ismahorin@fedora ~]$ cp feathers file.old
[ismahorin@fedora ~]$ ls
abcl feathers my_os work Изображения Шаблоны
australia file.old play Видео Музыка
bin may reports Документы Общедоступные
equipment monthly ski.plases Загрузки 'Рабочий стол'
```

Рис. 0.12: Копирование файла feathers в file.old

Переместим файл file.old в каталог play (рис. [-@fig:013]).

```
[ismahorin@fedora ~]$ mv file.old play
[ismahorin@fedora ~]$ cd play
[ismahorin@fedora play]$ ls
file.old
```

Рис. 0.13: Перемещение файла file.old в каталог play

Скопируем каталог play в каталог fun (рис. [-@fig:014]).

```
[ismahorin@fedora ~]$ cp -r play fun
[ismahorin@fedora ~]$ cd fun
[ismahorin@fedora fun]$ ls
file.old
```

Рис. 0.14: Копирование каталога play в каталог fun

Переместим каталог fun в каталог play и назовём его games (рис. [-@fig:015]).

```
[ismahorin@fedora fun]$ cd
[ismahorin@fedora ~]$ mv fun play
[ismahorin@fedora ~]$ cd play
[ismahorin@fedora play]$ ls
file.old fun
[ismahorin@fedora play]$ cd
[ismahorin@fedora ~]$ mv play/fun play/games
[ismahorin@fedora ~]$ cd play
[ismahorin@fedora play]$ ls
file.old games
```

Рис. 0.15: Перемещение каталога fun с изменением название на games

Лишаем владельца файла права на чтение. При попытке просмотреть файл мы получаем отказ в доступе, такой же отказ мы получаем при попытке скопировать этот файл. В конце возвращаем владельцу файла право на чтение (рис. [-@fig:016]).

```
[ismahorin@fedora play]$ cd
[ismahorin@fedora ~]$ chmod u-r feathers
[ismahorin@fedora ~]$ cat feathers
cat: feathers: Отказано в доступе
[ismahorin@fedora ~]$ cp feathers play
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
[ismahorin@fedora ~]$ cp feathers equipment
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
[ismahorin@fedora ~]$ chmod u+r feathers
```

Рис. 0.16: Лишение права на чтение. Попытки чтения и копирования файла. Возвращение права на чтение

Лишаем владельца каталога play права на выполнение. При попытке перейти в этот каталог мы получаем отказ в доступе. Возвращаем владельцу каталога право на выполнение (рис. [-@fig:017]).

```

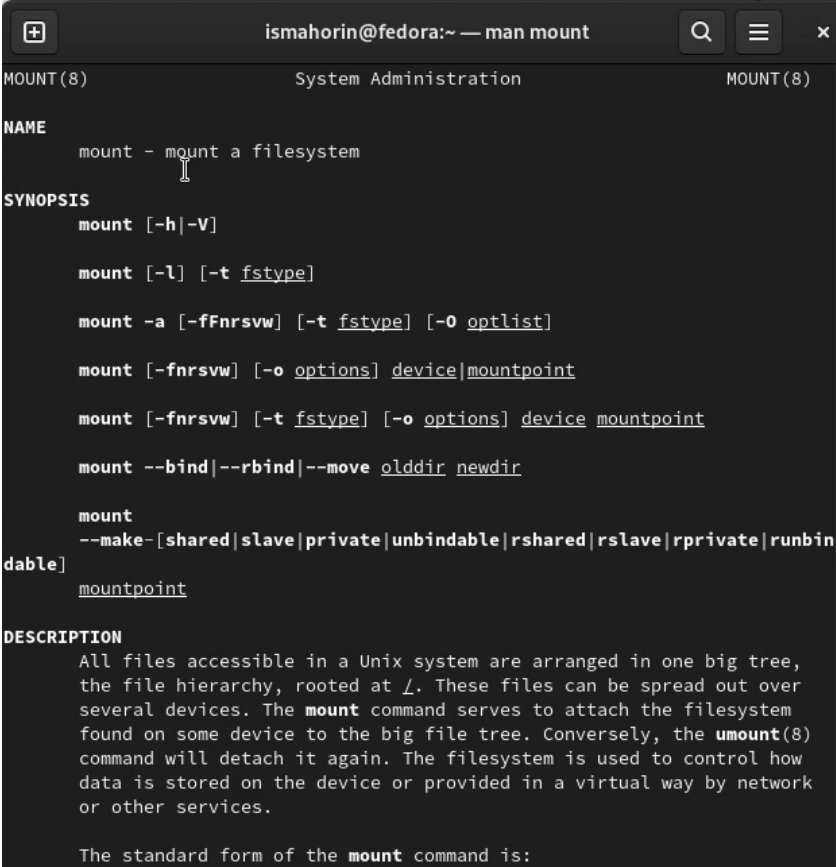
[ismahorin@fedora ~]$ chmod u-x play
[ismahorin@fedora ~]$ cd play
bash: cd: play: Отказано в доступе
[ismahorin@fedora ~]$ chmod u+x play
[ismahorin@fedora ~]$ cd play
[ismahorin@fedora play]$ ls
file.old  games
[ismahorin@fedora play]$ cd
[ismahorin@fedora ~]$

```

Рис. 0.17: Лишение права на выполнение. Попытка перехода в каталог. Возвращение права на выполнение

Прочитаем с помощью команды `man` следующие команды: `mount`, `fsck`, `mkfs`, `kill`. Кратко охарактеризуем эти команды.

Для просмотра используемых в операционной системе файловых систем используется команда `mount` (рис. [-@fig:018]).



```

ismahorin@fedora:~ — man mount
MOUNT(8)                                System Administration                                MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-fFnrsvw] [-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|rshared|rslave|rprivate|runbin-
dable]
    mountpoint

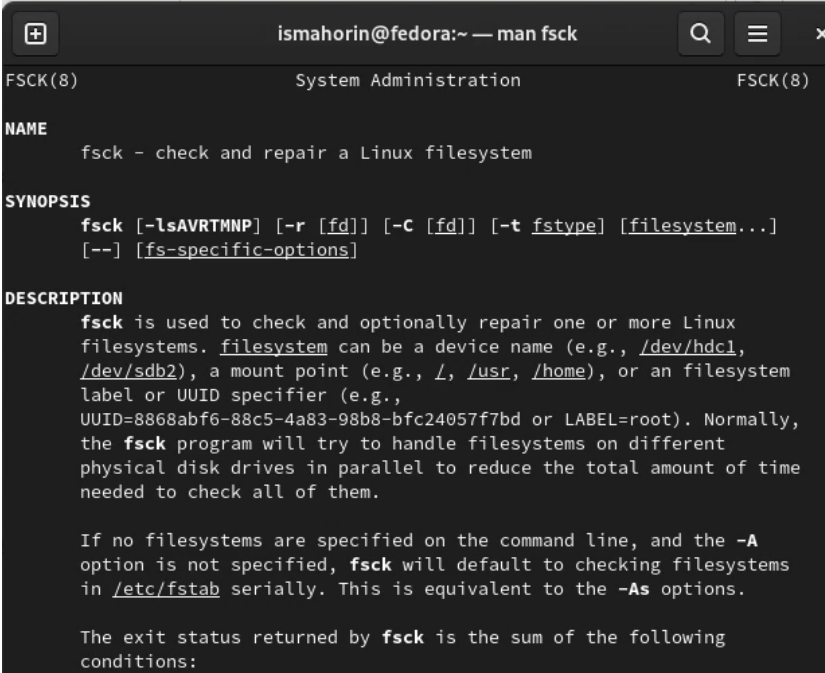
DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree,
    the file hierarchy, rooted at /. These files can be spread out over
    several devices. The mount command serves to attach the filesystem
    found on some device to the big file tree. Conversely, the umount(8)
    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:

```

Рис. 0.18: `man mount`

С помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы (рис. [-@fig:019]).



```
ismahorin@fedora:~ — man fsck
FSCK(8) System Administration FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

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

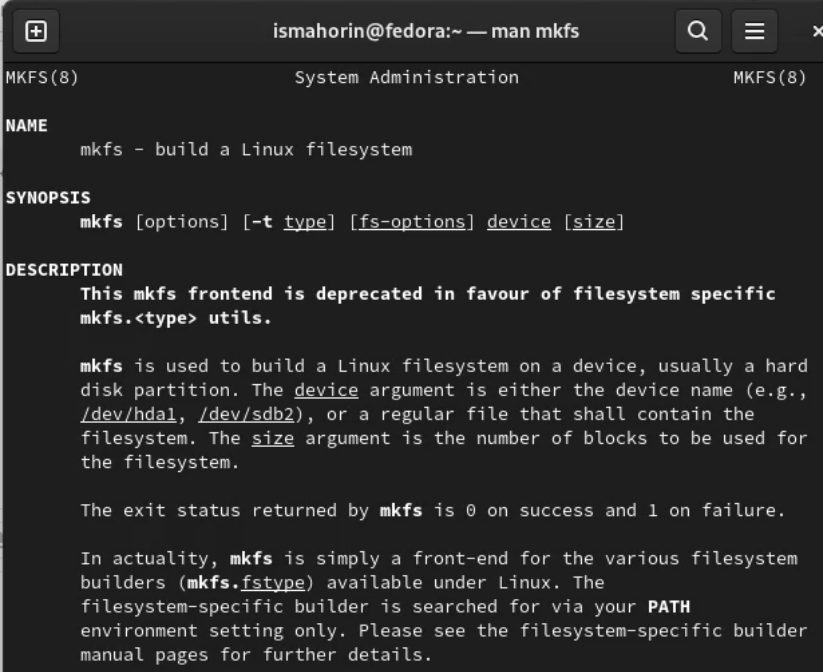
DESCRIPTION
    fsck is used to check and optionally repair one or more Linux
    filesystems. filesystem can be a device name (e.g., /dev/hdc1,
    /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem
    label or UUID specifier (e.g.,
    UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root).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

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

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

Рис. 0.19: `man fsck`

`mkfs` используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента `filesystem` для файловой системы может выступать или название устройства (например, `/dev/hda1`, `/dev/sdb2`) или точка монтирования (например, `/`, `/usr`, `/home`) (рис. [-@fig:020]).



```
ismahorin@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.
```

Рис. 0.20: man mkfs

Утилита `kill` отправляет сигнал процессу(-ам), указанному с помощью каждого из операндов идентификатор\_процесса. По умолчанию утилита `kill` отправляет сигнал `SIGTERM`, но эту настройку по умолчанию можно переопределить путем определения имени сигнала для отправки (рис. [-@fig:021]).



```
ismahorin@fedora:~ — man kill
KILL(1) User Commands KILL(1)

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.
```

Рис. 0.21: man kill

# Контрольные вопросы

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

Ext2, Ext3, Ext4 или Extended Filesystem - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений. В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзабайта.

JFS или Journaled File System была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопроцессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.

ReiserFS - была разработана намного позже, в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она была разработана под руководством Ганса Райзера и поддерживает только Linux. Из особенностей можно отметить динамический размер блока, что позволяет упаковывать несколько

небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами. Еще одно преимущество - в возможности изменять размеры разделов на лету. Но минус в некоторой нестабильности и риске потери данных при отключении энергии. Раньше ReiserFS применялась по умолчанию в SUSE Linux, но сейчас разработчики перешли на Btrfs.

XFS - это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы еще в 2001 году. Она изначально была рассчитана на файлы большого размера, и поддерживала диски до 2 Терабайт. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету и незначительный размер служебной информации.

XFS - журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по умолчанию в дистрибутивах на основе Red Hat. Из недостатков - это невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находится в памяти.

Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера на лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

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

/ — root каталог. Содержит в себе всю иерархию системы;

`/bin` — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: `pwd`, `ls`, `cat`, `ps`);

`/boot` — тут расположены файлы, используемые для загрузки системы (образ `initrd`, ядро `vmlinuz`);

`/dev` — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;

`/etc` — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;

`/home` — каталог, аналогичный каталогу `Users` в `Windows`. Содержит домашние каталоги учетных записей пользователей (кроме `root`). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;

`/lib` — содержит системные библиотеки, с которыми работают программы и модули ядра;

`/lost+found` — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;

`/media` — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию `/media/cdrom`;

`/mnt` — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;

`/opt` — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);

`/proc` — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;

`/root` — директория, которая содержит файлы и личные настройки суперпользо-

вателя;

/run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;

/sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;

/srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);

/sys — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;

/tmp — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;

/usr — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой;

/var — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.

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

Монтирование тома.

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

Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:

- Один блок адресуется несколькими mode (принадлежит нескольким файлам).

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

5. Как создаётся файловая система?

`mkfs` - позволяет создать файловую систему Linux.

6. Дайте характеристику командам для просмотра текстовых файлов.

`Cat` - выводит содержимое файла на стандартное устройство вывода

7. Приведите основные возможности команды `cp` в Linux.

`Ср` – копирует или перемещает директорию, файлы.

8. Приведите основные возможности команды `mv` в Linux.

`Mv` - переименовать или переместить файл или директорию

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

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

## Выводы

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