

Лабораторная работа №7

Операционные системы

Замула Егор Сергеевич

Содержание

Цель работы	5
Задание	6
Выполнение лабораторной работы	8
Контрольные вопросы	17
Выводы	23

Список иллюстраций

1	Копирование файла io.h под названием equipment. Проверка. . .	8
2	Создание директории ski.places	8
3	Перемещение файла equipment в каталог ski.places	9
4	Переименование файла equipment в equiplist	9
5	Создание файла abc1 и копирование его в каталог ski.places под названием equiplist2	9
6	Создание каталога equipment в каталоге ski.places	9
7	Перемещение файлов в подкаталог equipment	10
8	Создание каталога newdir и перемещение его в каталог ski.places под названием plans	10
9	Создание каталогов и файлов	10
10	Присваивание прав доступа	11
11	Просмотр содержимого файла passwd	11
12	Перемещение файла file.old в каталог play	12
13	Копирование каталога play в каталог fun	12
14	Лишение права на чтение. Попытки чтения и копирования файла. Возвращение права на чтение	13
15	Лишение права на выполнение. Попытка перехода в каталог. Возвращение права на выполнение	13
16	man mount	14
17	man fsck	14
18	man mkfs	15

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.

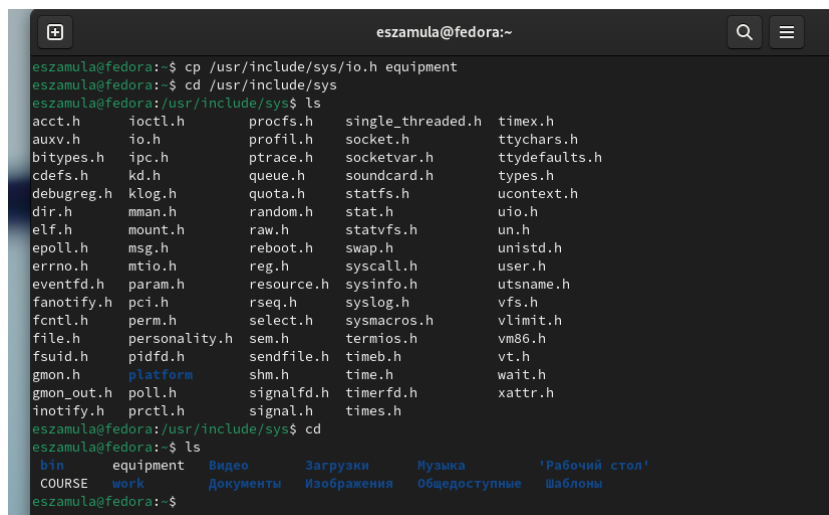
Задание

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

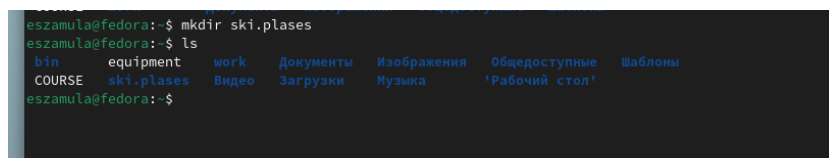
1. Скопируем файл `io.h` в домашний каталог и назовём его `equipment`. Для этого воспользуемся командой `cp` и укажем путь к нашему файлу. Выполним проверку командой `ls`



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

Рис. 1: Копирование файла `io.h` под названием `equipment`. Проверка.

2. В домашнем каталоге командой `mkdir` создаём директорию `ski.places`. Выполняем проверку



```
eszamula@fedora:~$ mkdir ski.places
eszamula@fedora:~$ ls
bin      equipment  work      Документы  Изображения  Общедоступные  Шаблоны
COURSE  ski.places Видео      Загрузки    Музыка      'Рабочий стол'
eszamula@fedora:~$
```

Рис. 2: Создание директории `ski.places`

3. Перемещаем файл `equipment` в каталог `ski.places` командой `mv`. Выполняем проверку

```
eszamula@fedora:~/ski.places$ mv equipment ski.places
eszamula@fedora:~/ski.places$ ls
bin      ski.places  Видео      Загрузки  Музыка      'Рабочий стол'
COURSE   work        Документы  Изображения  Общедоступные  Шаблоны
eszamula@fedora:~/ski.places$
```

Рис. 3: Перемещение файла `equipment` в каталог `ski.places`

4. Переименуем файл `equipment`, находящийся в каталоге `ski.places` в `equiplist` с помощью команды `mv`. Выполняем проверку

```
eszamula@fedora:~/ski.places$ cd
eszamula@fedora:~$ mv ski.places/equipment ski.places/equiplist
eszamula@fedora:~$ cd ski.places
eszamula@fedora:~/ski.places$ ls
equiplist
eszamula@fedora:~/ski.places$
```

Рис. 4: Переименование файла `equipment` в `equiplist`

5. Создаём в домашнем каталоге файл `abc1` командой `touch` и копируем его в каталог `ski.places` под названием `equiplist2`. Выполняем проверку

```
eszamula@fedora:~/ski.places$ cd
eszamula@fedora:~$ touch abc1
eszamula@fedora:~$ cp abc1 ski.places
eszamula@fedora:~$ mv ski.places/abc1 ski.places/equiplist2
eszamula@fedora:~$ cd ski.places
eszamula@fedora:~/ski.places$ ls
equiplist  equiplist2
eszamula@fedora:~/ski.places$
```

Рис. 5: Создание файла `abc1` и копирование его в каталог `ski.places` под названием `equiplist2`

6. Создаём каталог с именем `equipment` в каталоге `ski.places`

```
eszamula@fedora:~/ski.places$ cd
eszamula@fedora:~$ mv ski.places/equiplist equipment
eszamula@fedora:~$ mv ski.places/equiplist2 equipment
eszamula@fedora:~$ ls
abc1  COURSE  ski.places  Видео  Загрузки  Музыка  'Рабочий стол'
bin   equipment  work        Документы  Изображения  Общедоступные  Шаблоны
eszamula@fedora:~$ cd ski.places
eszamula@fedora:~/ski.places$ ls
equipment
eszamula@fedora:~/ski.places$
```

Рис. 6: Создание каталога `equipment` в каталоге `ski.places`

7. Перемещаем файлы `equiplist` и `equiplist2` из каталога `ski.plases` в подкаталог `equipment`

```
eszmula@fedora:~/ski.plases$ cd
eszmula@fedora:~/ski.plases$ mv ski.plases/equiplist equipment
eszmula@fedora:~/ski.plases$ mv ski.plases/equiplist2 equipment
eszmula@fedora:~/ski.plases$ ls
abcl  COURSE  ski.plases  Видео  Загрузки  Музыка  'Рабочий стол'
bin   equipment work        Документы  Изображения  Общедоступные  Шаблоны
eszmula@fedora:~/ski.plases$ cd ski.plases
eszmula@fedora:~/ski.plases$ ls
equipment
eszmula@fedora:~/ski.plases$
```

Рис. 7: Перемещение файлов в подкаталог `equipment`

8. Создаём каталог `newdir`. Далее перемещаем его в каталог `ski.plases` под названием `plans`

```
eszmula@fedora:~/ski.plases$ cd
eszmula@fedora:~/ski.plases$ mkdir newdir
eszmula@fedora:~/ski.plases$ ls
abcl  COURSE  newdir  work  Документы  Изображения  Общедоступные  Шаблоны
bin   equipment ski.plases  Видео  Загрузки  Музыка  'Рабочий стол'
eszmula@fedora:~/ski.plases$ mv newdir ski.plases
eszmula@fedora:~/ski.plases$ mv ski.plases/newdir ski.plases/plans
eszmula@fedora:~/ski.plases$ cd ski.plases
eszmula@fedora:~/ski.plases$ ls
equipment plans
eszmula@fedora:~/ski.plases$
```

Рис. 8: Создание каталога `newdir` и перемещение его в каталог `ski.plases` под названием `plans`

9. Создаём 2 каталога (`australia` и `play`) и 2 файла (`my_os` и `feathers`)

```
equipment plans
eszmula@fedora:~/ski.plases$ cd
eszmula@fedora:~/ski.plases$ mkdir australia
eszmula@fedora:~/ski.plases$ mkdir play
eszmula@fedora:~/ski.plases$ touch my_os
eszmula@fedora:~/ski.plases$ touch feathers
eszmula@fedora:~/ski.plases$ ls
abcl  COURSE  my_os  work  Загрузки  Общедоступные
australia  equipment  play  Видео  Изображения  'Рабочий стол'
bin   feathers  ski.plases  Документы  Музыка  Шаблоны
eszmula@fedora:~/ski.plases$
```

Рис. 9: Создание каталогов и файлов

10. Далее присвоим каждому из каталогов и файлов определённые права доступа

```
eszamula@fedora:~$ chmod 744 australia
eszamula@fedora:~$ chmod 711 play
eszamula@fedora:~$ chmod 544 my_os
eszamula@fedora:~$ chmod 664 feathers
eszamula@fedora:~$ ls -l
итого 4
-rw-r--r--. 1 eszamula eszamula 0 мар 23 11:26 abc1
drwxr--r--. 1 eszamula eszamula 0 мар 23 11:31 australia
drwxr-xr-x. 1 eszamula eszamula 8 мар 14 22:51 bin
-rw-r--r--. 1 eszamula eszamula 9 фев 24 14:44 COURSE
-rw-r--r--. 1 eszamula eszamula 0 мар 23 11:26 equipment
-rw-rw-r--. 1 eszamula eszamula 0 мар 23 11:31 feathers
-r-xr--r--. 1 eszamula eszamula 0 мар 23 11:31 my_os
drwx--x--x. 1 eszamula eszamula 0 мар 23 11:31 play
drwxr-xr-x. 1 eszamula eszamula 28 мар 23 11:30 ski.places
drwxr-xr-x. 1 eszamula eszamula 56 мар 14 23:42 work
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Видео
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Документы
drwxr-xr-x. 1 eszamula eszamula 688 мар 15 12:11 Загрузки
drwxr-xr-x. 1 eszamula eszamula 172 мар 15 12:40 Изображения
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Музыка
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Общедоступные
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 'Рабочий стол'
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Шаблоны
eszamula@fedora:~$
```

Рис. 10: Присваивание прав доступа

11. Просматриваем содержимое файла passwd с помощью команды cat

```
eszamula@fedora:~$ chmod 744 australia
eszamula@fedora:~$ chmod 711 play
eszamula@fedora:~$ chmod 544 my_os
eszamula@fedora:~$ chmod 664 feathers
eszamula@fedora:~$ ls -l
итого 4
-rw-r--r--. 1 eszamula eszamula 0 мар 23 11:26 abc1
drwxr--r--. 1 eszamula eszamula 0 мар 23 11:31 australia
drwxr-xr-x. 1 eszamula eszamula 8 мар 14 22:51 bin
-rw-r--r--. 1 eszamula eszamula 9 фев 24 14:44 COURSE
-rw-r--r--. 1 eszamula eszamula 0 мар 23 11:26 equipment
-rw-rw-r--. 1 eszamula eszamula 0 мар 23 11:31 feathers
-r-xr--r--. 1 eszamula eszamula 0 мар 23 11:31 my_os
drwx--x--x. 1 eszamula eszamula 0 мар 23 11:31 play
drwxr-xr-x. 1 eszamula eszamula 28 мар 23 11:30 ski.places
drwxr-xr-x. 1 eszamula eszamula 56 мар 14 23:42 work
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Видео
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Документы
drwxr-xr-x. 1 eszamula eszamula 688 мар 15 12:11 Загрузки
drwxr-xr-x. 1 eszamula eszamula 172 мар 15 12:40 Изображения
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Музыка
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Общедоступные
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 'Рабочий стол'
drwxr-xr-x. 1 eszamula eszamula 0 фев 18 23:11 Шаблоны
eszamula@fedora:~$
```

Рис. 11: Просмотр содержимого файла passwd

```

eszamula@fedora:~$ cd
eszamula@fedora:~$ cat /etc/passwd
root:x:0:0:Super User:/root:/bin/bash
bin:x:1:1:bin:/bin:/usr/sbin/nologin
daemon:x:2:2:daemon:/sbin:/usr/sbin/nologin
adm:x:3:4:adm:/var/adm:/usr/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/usr/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:/usr/sbin/nologin
operator:x:11:0:operator:/root:/usr/sbin/nologin
games:x:12:100:games:/usr/games:/usr/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/usr/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/usr/sbin/nologin
dbus:x:81:81:System Message Bus:/usr/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/usr/sbin/nologin
systemd-coredump:x:998:998:systemd Core Dumper:/usr/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/usr/sbin/nologin
systemd-oom:x:997:997:systemd Userspace OOM Killer:/usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization:/usr/sbin/nologin
qemu:x:107:107:qemu user:/usr/sbin/nologin
polkitd:x:114:114:User for polkitd:/usr/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin

```

12. Скопируем файл feathers в file.old

13. Переместим файл file.old в каталог play

```

eszamula@fedora:~$ cp feathers file.old
eszamula@fedora:~$ ls
abcl      COURSE    file.old  ski.places  Документы  Музыка  Шаблоны
australia equipment my_os      work        Загрузки   Общедоступные
bin       feathers  play      Видео       Изображения  'Рабочий стол'
eszamula@fedora:~$

```

Рис. 12: Перемещение файла file.old в каталог play

14. Скопируем каталог play в каталог fun

```

eszamula@fedora:~/play$ cd
eszamula@fedora:~$ cp -r play fun
eszamula@fedora:~$ cd fun
eszamula@fedora:~/fun$ ls
file.old
eszamula@fedora:~/fun$

```

Рис. 13: Копирование каталога play в каталог fun

15. Переместим каталог fun в каталог play и назовём его games

```

eszamula@fedora:~/fun$ cd
eszamula@fedora:~$ mv fun play
eszamula@fedora:~$ cd play
eszamula@fedora:~/play$ ls
file.old  fun
eszamula@fedora:~/play$ cd
eszamula@fedora:~$ mv play/fun play/games
eszamula@fedora:~$ cd play
eszamula@fedora:~/play$ ls
file.old  games
eszamula@fedora:~/play$

```

16. 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

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

Рис. 14: Лишение права на чтение. Попытки чтения и копирования файла. Возвращение права на чтение

17. Лишаем владельца каталога play права на выполнение. При попытке перейти в этот каталог мы получаем отказ в доступе. Возвращаем владельцу каталога право на выполнение

```
eszmula@fedora:~$ chmod u-x play
eszmula@fedora:~$ cd play
bash: cd: play: Отказано в доступе
eszmula@fedora:~$ chmod u+x play
eszmula@fedora:~$ cd play
eszmula@fedora:~/play$ ls
file.oid  games
eszmula@fedora:~/play$ cd
eszmula@fedora:~$
```

Рис. 15: Лишение права на выполнение. Попытка перехода в каталог. Возвращение права на выполнение

18. 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

```

eszamula@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|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

Manual page mount(8) line 1 (press h for help or q to quit)

```

Рис. 16: man mount

19. С помощью команды **fsck** можно проверить (а в ряде случаев восстановить) целостность файловой системы

```

eszamula@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 a
    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:

    0
        No errors

    1
        Filesystem errors corrected

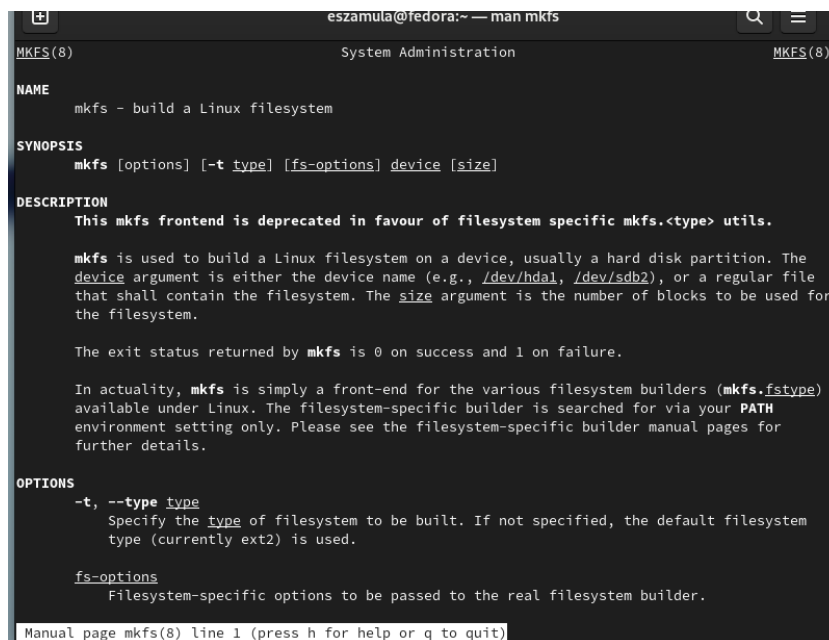
    2
        System should be rebooted

Manual page fsck(8) line 1 (press h for help or q to quit)

```

Рис. 17: man fsck

20. `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`)



```
eszamula@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.

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

Рис. 18: `man mkfs`

21. 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

```
eszamula@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.

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

Manual page kill(1) line 1 (press h for help or q to quit)
```


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 экзбайта.

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 суперпользователя;

/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.

Cp – 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.