

Лабораторная работа №5

Дисциплина: Операционные системы

Алиева Милена Арифовна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	8
4	Выполнение лабораторной работы	9
5	Ответы на контрольные вопросы	18
6	Вывод	22

Список иллюстраций

4.1	Выполнение примеров - копирование файлов и каталогов	9
4.2	Выполнение примеров - перемещение и переименовывание фай- лов и каталогов	10
4.3	Выполнение примеров - права доступа	10
4.4	Копирование файла io.h	10
4.5	Работа с директорией ~/ski.plases	11
4.6	Продолжение работы с директорией ~/ski.plases	11
4.7	Создание 2 каталогов и 2 файлов	12
4.8	Изначальные права доступа	12
4.9	Права доступа после изменений	13
4.10	Просмотр файла /etc/password	13
4.11	Работа с файлами и каталогами	14
4.12	Работа с правами доступа файла features	14
4.13	Работа с правами доступа каталога ~/play	15
4.14	man mount	15
4.15	man fsck	16
4.16	man mkfs	16
4.17	man kill	17

Список таблиц

1 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

2 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:
 - 2.1. 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.
 - 2.2. В домашнем каталоге создайте директорию `~/ski.places`.
 - 2.3. Переместите файл `equipment` в каталог `~/ski.places`.
 - 2.4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 - 2.5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
 - 2.6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
 - 2.7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 - 2.8. 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:
 - 3.1. `drwxr-r- ... australia`
 - 3.2. `drwx-x-x ... play`
 - 3.3. `-r-xr-r- ... my_os`
 - 3.4. `-rw-rw-r- ... feathers`При необходимости создайте нужные файлы.
4. Прделайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 - 4.1. Просмотрите содержи-

- мое файла `/etc/password`. 4.2. Скопируйте файл `~/feathers` в файл `~/file.old`. 4.3. Переместите файл `~/file.old` в каталог `~/play`. 4.4. Скопируйте каталог `~/play` в каталог `~/fun`. 4.5. Переместите каталог `~/fun` в каталог `~/play` и назовите его `games`. 4.6. Лишите владельца файла `~/feathers` права на чтение. 4.7. Что произойдёт, если вы попытаетесь просмотреть файл `~/feathers` командой `cat`? 4.8. Что произойдёт, если вы попытаетесь скопировать файл `~/feathers`? 4.9. Дайте владельцу файла `~/feathers` право на чтение. 4.10. Лишите владельца каталога `~/play` права на выполнение. 4.11. Перейдите в каталог `~/play`. Что произошло? 4.12. Дайте владельцу каталога `~/play` право на выполнение.
5. Прочитайте `man` по командам `mount`, `fsck`, `mkfs`, `kill`.

3 Теоретическое введение

Файловая система (ФС) — архитектура хранения данных, которые могут находиться в разделах жесткого диска и ОП. Выдает пользователю доступ к конфигурации ядра. Определяет, какую структуру принимают файлы в каждом из разделов, создает правила для их генерации, а также управляет файлами в соответствии с особенностями каждой конкретной ФС. Основные файловые системы, используемые в дистрибутивах Linux: Ext2; Ext3; Ext4; JFS; ReiserFS; XFS; Btrfs; ZFS. Ext2, Ext3, Ext4 или Extended Filesystem – стандартная файловая система, первоначально разработанная еще для Minix.

4 Выполнение лабораторной работы

1. Выполним примеры, приведенные в первой части лабораторной работы.

Копирование файлов и каталогов (рис. [4.1])

```
[maalieva@fedora ~]$ touch april
[maalieva@fedora ~]$ cp abc1 april
[maalieva@fedora ~]$ touch may
[maalieva@fedora ~]$ cp abc1 may
[maalieva@fedora ~]$ mkdir monthly
[maalieva@fedora ~]$ cp april may monthly
[maalieva@fedora ~]$ cd ~/monthly
[maalieva@fedora monthly]$ touch june
[maalieva@fedora monthly]$ cd
[maalieva@fedora ~]$ cp monthly/may monthly/june
[maalieva@fedora ~]$ ls monthly

april  june  may
[maalieva@fedora ~]$ mkdir monthly.00
[maalieva@fedora ~]$ cp -r monthly monthly.00
[maalieva@fedora ~]$ cp -r monthly.00 /tmp
[maalieva@fedora ~]$ ls tmp
ls: невозможно получить доступ к 'tmp': Нет такого файла или каталога
[maalieva@fedora ~]$ ls ~/tmp
ls: невозможно получить доступ к '/home/maalieva/tmp': Нет такого файла или каталога
[maalieva@fedora ~]$ ls /tmp
hugo_cache
monthly.00
systemd-private-b9dc25f2cd9141cbb356475152f4715c-chronyd.service-588Y0H
systemd-private-b9dc25f2cd9141cbb356475152f4715c-colord.service-whU3IV
systemd-private-b9dc25f2cd9141cbb356475152f4715c-dbus-broker.service-fQmxhE
systemd-private-b9dc25f2cd9141cbb356475152f4715c-gaoclus.service-f710pP
systemd-private-b9dc25f2cd9141cbb356475152f4715c-lve-memory-monitor.service-wVY7iC
systemd-private-b9dc25f2cd9141cbb356475152f4715c-ModemManager.service-0u1IVn
systemd-private-b9dc25f2cd9141cbb356475152f4715c-power-profiles-daemon.service-2PdIYW
systemd-private-b9dc25f2cd9141cbb356475152f4715c-rtkit-daemon.service-Kg0csb
systemd-private-b9dc25f2cd9141cbb356475152f4715c-switcheroo-control.service-Yesu8g
systemd-private-b9dc25f2cd9141cbb356475152f4715c-systemd-logind.service-uh6yz9
systemd-private-b9dc25f2cd9141cbb356475152f4715c-systemd-oomd.service-8rlkM2
systemd-private-b9dc25f2cd9141cbb356475152f4715c-systemd-resolved.service-Q5JvR9
systemd-private-b9dc25f2cd9141cbb356475152f4715c-upower.service-2KPt6z
```

Рис. 4.1: Выполнение примеров - копирование файлов и каталогов

2. Выполним примеры, приведенные в первой части лабораторной работы.

Перемещение и переименовывание файлов и каталогов (рис. [4.2])

```

[maalieva@fedora ~]$ mv april july
[maalieva@fedora ~]$ mv july monthly.00
[maalieva@fedora ~]$ ls monthly.00
july  monthly
[maalieva@fedora ~]$ mv monthly.00 monthly.01
[maalieva@fedora ~]$ mkdir reports
[maalieva@fedora ~]$ mv monthly.01 reports
[maalieva@fedora ~]$ mv reports/monthly.01 reports/monthly

```

Рис. 4.2: Выполнение примеров - перемещение и переименовывание файлов и каталогов

3. Выполним примеры, приведенные в первой части лабораторной работы. Перемещение и переименовывание файлов и каталогов. Права доступа (рис. [4.3])

```

[maalieva@fedora ~]$ mkdir monthly
[maalieva@fedora ~]$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
[maalieva@fedora ~]$ chmod g-r,o-r monthly
[maalieva@fedora ~]$ touch abc1
[maalieva@fedora ~]$ chmod g+w abc1
[maalieva@fedora ~]$

```

Рис. 4.3: Выполнение примеров - права доступа

4. Скопируем файл /usr/include/sys/io.h в домашний каталог и назовём его equipment (рис. [4.4])

```

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

```

Рис. 4.4: Копирование файла io.h

5. В домашнем каталоге создадим директорию `~/ski.plases` и переместим файл `equipment` в каталог `~/ski.plases`. Далее переименуем файл `~/ski.plases/equipment` в `~/ski.plases/equiplist`. Затем создадим в домашнем каталоге файл `abc1` и скопируем его в каталог `~/ski.plases`, назовём его `equiplist2` (рис. [4.5])

```
[maalieva@fedora ~]$ mkdir ski.plases
[maalieva@fedora ~]$ cp equipment ski.plase
[maalieva@fedora ~]$ cp equipment ski.plases
[maalieva@fedora ~]$ rm ski.plase
[maalieva@fedora ~]$ cp abc1 ski.plases
[maalieva@fedora ~]$ cd ~/ski.plases
[maalieva@fedora ski.plases]$ mv abc1 equiplist2
[maalieva@fedora ski.plases]$ mv equipment equiplist
[maalieva@fedora ski.plases]$
```

Рис. 4.5: Работа с директорией `~/ski.plases`

6. Создадим каталог с именем `equipment` в каталоге `~/ski.plases` и переместим файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment`. Затем создадим и переместим каталог `~/newdir` в каталог `~/ski.plases` и назовем его `plans` (рис. [4.6])

```
[maalieva@fedora ski.plases]$ mv equipment equiplist
[maalieva@fedora ski.plases]$ mv equiplist2 equipment
[maalieva@fedora ski.plases]$ cd
[maalieva@fedora ~]$ mkdir newdir
[maalieva@fedora ~]$ mv newdir ski.plases
[maalieva@fedora ~]$ cd ~/ski.plases
[maalieva@fedora ski.plases]$ mv newdir plans
[maalieva@fedora ski.plases]$
```

Рис. 4.6: Продолжение работы с директорией `~/ski.plases`

7. Создадим 2 новых каталога и 2 новых файла в домашнем каталоге для выполнения задания 3 (рис. [4.7])

```
[maalieva@fedora ~]$ mkdir australia
[maalieva@fedora ~]$ mkdir play
[maalieva@fedora ~]$ touch my_os
[maalieva@fedora ~]$ touch features
```

Рис. 4.7: Создание 2 каталогов и 2 файлов

7. Изначальные права доступа ранее созданных файлов (рис. [4.8])

```
[maalieva@fedora ~]$ ls -l
итого 8
-rw-rw-r--. 1 maalieva maalieva 0 мар 2 17:53 abc1
drwxr-xr-x. 1 maalieva maalieva 0 мар 2 18:24 australia
drwxr-xr-x. 1 maalieva maalieva 8 мар 2 16:56 bin
-rw-r--r--. 1 maalieva maalieva 5086 мар 2 18:01 equipment
-rw-r--r--. 1 maalieva maalieva 0 мар 2 18:24 features
drwx--x--x. 1 maalieva maalieva 0 мар 2 17:52 monthly
-rw-r--r--. 1 maalieva maalieva 0 мар 2 18:24 my_os
drwxr-xr-x. 1 maalieva maalieva 0 мар 2 18:24 play
drwxr-xr-x. 1 maalieva maalieva 28 мар 2 18:12 ski.places
drwxr-xr-x. 1 maalieva maalieva 112 мар 2 17:00 work
drwxr-xr-x. 1 maalieva maalieva 0 мар 1 20:20 Видео
drwxr-xr-x. 1 maalieva maalieva 0 мар 1 20:20 Документы
drwxr-xr-x. 1 maalieva maalieva 146 мар 2 16:56 Загрузки
drwxr-xr-x. 1 maalieva maalieva 0 мар 1 20:20 Изображения
drwxr-xr-x. 1 maalieva maalieva 0 мар 1 20:20 Музыка
drwxr-xr-x. 1 maalieva maalieva 0 мар 1 20:20 Общедоступные
drwxr-xr-x. 1 maalieva maalieva 0 мар 1 20:20 'Рабочий стол'
drwxr-xr-x. 1 maalieva maalieva 0 мар 1 20:20 Шаблоны
[maalieva@fedora ~]$
```

Рис. 4.8: Изначальные права доступа

Права доступа после наших изменений. При проверке они полностью соответствуют заданию (рис. [4.9])

```
[maalieva@fedora ~]$ ls -l
итого 8
-rw-rw-r--. 1 maalieva maalieva 0 map 2 17:53 abc1
drwxr--r--. 1 maalieva maalieva 0 map 2 18:24 australia
drwxr-xr-x. 1 maalieva maalieva 8 map 2 16:56 bin
-rw-r--r--. 1 maalieva maalieva 5086 map 2 18:01 equipment
-rw-rw-r--. 1 maalieva maalieva 0 map 2 18:24 features
drwx--x--x. 1 maalieva maalieva 0 map 2 17:52 monthly
-r-xr--r--. 1 maalieva maalieva 0 map 2 18:24 my_os
drwx--x--x. 1 maalieva maalieva 0 map 2 18:24 play
drwxr-xr-x. 1 maalieva maalieva 28 map 2 18:12 ski.places
drwxr-xr-x. 1 maalieva maalieva 112 map 2 17:00 work
drwxr-xr-x. 1 maalieva maalieva 0 map 1 20:20 Видео
drwxr-xr-x. 1 maalieva maalieva 0 map 1 20:20 Документы
drwxr-xr-x. 1 maalieva maalieva 146 map 2 16:56 Загрузки
drwxr-xr-x. 1 maalieva maalieva 0 map 1 20:20 Изображения
drwxr-xr-x. 1 maalieva maalieva 0 map 1 20:20 Музыка
drwxr-xr-x. 1 maalieva maalieva 0 map 1 20:20 Общедоступные
drwxr-xr-x. 1 maalieva maalieva 0 map 1 20:20 'Рабочий стол'
drwxr-xr-x. 1 maalieva maalieva 0 map 1 20:20 Шаблоны
```

Рис. 4.9: Права доступа после изменений

8. По заданию необходимо просмотреть содержимое файла /etc/passwd. Но при попытке сделать это, обнаруживаем, что такого файла у нас нет (рис. [4.10])

```
[maalieva@fedora ~]$ cd /etc
[maalieva@fedora etc]$ cat passwd
cat: passwd: Нет такого файла или каталога
[maalieva@fedora etc]$ ls
abrt                                gccrypt                            ndepctl.d                         rmtab.d
adjtime                             gdbinit                           mncurial                          rygel.conf
aliases                             gdbinit.d                        mime.types                        samba
aliases.conf                       gdm                               mke2fs.conf                      sasl2
anaconda                           geoclue                          modprobe.d                       security
anthy-unicode.conf                 glvd                             modules-load.d                   selinux
appstream.conf                    GREP_COLORS                      motd                              services
asound.conf                        groff                             mtd.d                            sestatus.conf
audit                              group                             mtools.conf                      sgml
authselect                        group-                            my.cnf                           shadow
avahi                             grub2.cfg                       nanorc                           shadow-
bash_completion.d                 grub2-efi.cfg                   ndctl                             shells
bashrc                             grub.d                           ndctl.conf.d                     skel
bindresvport.blacklist             gshadow                          networkmanager                   sos
binfmt.d                           gshadow-                        netconfig                        speech-dispatcher
bluetooth                          gas                               nfs.conf                         ssh
brlapi.key                         gasproxy                        nfsmount.conf                    ssl
brlapi.conf                       host.conf                       nftables                         statetab.d
brlapi.conf                       hostname                        nftables.conf                   subgid
brlapi.conf                       hosts                           nsswitch.conf                   subgid-
brlapi.conf                       httpd                           opensc.conf                     subuid
brlapi.conf                       idmapd.conf                    opensc-x86_64.conf              subuid-
brlapi.conf                       inittab                        openvpn                          sudo.conf
brlapi.conf                       inputrc                         opt                              sudoers
brlapi.conf                       iproute2                       os-release                      sudoers.d
```

Рис. 4.10: Просмотр файла /etc/passwd

9. Скопируем файл ~/features в файл ~/file.old и переместим файл ~/file.old в каталог ~/play. Далее скопируем каталог ~/play в каталог ~/fun и переместим

каталог ~/fun в каталог ~/play и назовем его games (рис. [4.11])

```
[maalievafedora ~]$ cp features file.old
[maalievafedora ~]$ mv file.old play
[maalievafedora ~]$ cd /play
bash: cd: /play: Нет такого файла или каталога
[maalievafedora ~]$ cp play fun
cp: не указан -r; пропускается каталог 'play'
[maalievafedora ~]$ mkdir fun
[maalievafedora ~]$ cp play fun
cp: не указан -r; пропускается каталог 'play'
[maalievafedora ~]$ cp -r play fun
[maalievafedora ~]$ mv fun play
[maalievafedora ~]$ mv play games
[maalievafedora ~]$
```

Рис. 4.11: Работа с файлами и каталогами

10. Лишим владельца файла ~/features права на чтение. Проверим, что при попытке просмотреть файл ~/features командой cat нам отказано в доступе, также убедимся, что при попытке скопировать файл ~/features нам тоже отказано в доступе. Вернём обратно владельцу файла ~/features право на чтение (рис. [4.12])

```
[maalievafedora ~]$ chmod u-r features
[maalievafedora ~]$ cat ~/features
cat: /home/maalieva/features: Нет такого файла или каталога
[maalievafedora ~]$ cat ~/features
cat: /home/maalieva/features: Отказано в доступе
[maalievafedora ~]$ cp ~/features
cp: после '/home/maalieva/features' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
[maalievafedora ~]$ cp features fun
cp: невозможно открыть 'features' для чтения: Отказано в доступе
[maalievafedora ~]$ chmod u+r features
[maalievafedora ~]$
```

Рис. 4.12: Работа с правами доступа файла features

11. Лишим владельца каталога ~/play права на выполнение. Попробуем перейти в каталог ~/play и увидим, что нам отказано в доступе. Вернём обратно владельцу каталога ~/play право на выполнение (рис. [4.13])

```

[maalieva@fedora ~]$ chmod u-x play
[maalieva@fedora ~]$ cd ~/play
bash: cd: /home/maalieva/play: Отказано в доступе
[maalieva@fedora ~]$ chmod u+x play

```

Рис. 4.13: Работа с правами доступа каталога ~/play

12. Прочитаем ман по командам mount: (рис. [4.14])

```

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

```

Рис. 4.14: man mount

fsck: (рис. [4.15])


```

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:

    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

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

```

Рис. 4.15: man fsck

mkfs: (рис. [4.16])

```

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.

    -h, --help
        Display help text and exit.

```

Рис. 4.16: man mkfs

kill: (рис. [4.17])

```
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.

    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.
```

Рис. 4.17: man kill

5 Ответы на 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 - стандартная файловая система для Linux. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана для Linux и получила много улучшений. В 2001 году появилась ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня.

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;

`/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 суперполь-

зователя;

/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.

Причины:

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 inode).

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?

Команда `mkfs` позволяет создать файловую систему Linux

6. Дайте характеристику командам для просмотра текстовых файлов.

Команда `cat` выводит содержимое файла на стандартное устройство вывода

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

Команда `cp` копирует директории или файлы

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

Команда `mv` переименовывает или перемещает файлы или директории

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.

6 Вывод

Я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов, также приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами