

Лабораторная работа №5. Анализ файловой системы Linux. Команды для работы с файлами и каталогами

НПИбд-01-22

Коннова Татьяна Алексеевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	8
4	Контрольные вопросы	19
5	Выводы	25

Список иллюстраций

3.1	Копирование файла io.h под названием equipment.	8
3.2	Копирование	8
3.3	Создание директории ski.places	9
3.4	Перемещение файла equipment в каталог ski.places	9
3.5	Создание файла abc1 и копирование его в каталог ski.places под названием equiplist2	9
3.6	Создание каталога equipment в каталоге ski.places	9
3.7	Перемещение файлов в подкаталог equipment	10
3.8	Создание каталога newdir и перемещение его в каталог ski.places под названием plans	10
3.9	Создание каталогов и файлов	11
3.10	Присваивание прав доступа	12
3.11	Просмотр содержимого файла passwd	12
3.12	Копирование файла feathers в file.old	13
3.13	Перемещение файла file.old в каталог play	13
3.14	Копирование каталога play в каталог fun	13
3.15	Перемещение каталога fun с изменением название на games	14
3.16	Лишение права на чтение. Попытки чтения и копирования файла. Возвращение права на чтение	14
3.17	Лишение права на выполнение. Попытка перехода в каталог. Возвращение права на выполнение	15
3.18	man mount	15
3.19	man fsck	16
3.20	man mkfs	17
3.21	man kill	18

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:
 - 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.

3 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. [3.1]), (рис. [3.2]).

```
[takonnova@fedora sys]$ cd  
[takonnova@fedora ~]$ cp /usr/include/sys/io.h equipment
```

Рис. 3.1: Копирование файла `io.h` под названием `equipment`.

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

Рис. 3.2: Копирование

В домашнем каталоге командой `mkdir` создаём директорию `ski.plases`. Выполняем проверку. Перемещаем файл `equipment` в каталог `ski.plases` командой `mv`. Выполняем проверку. Переименоуем файл `equipment`, находящийся в каталоге `ski.plases` в `equiplist` с помощью команды `mv`. Выполняем проверку (рис. [3.3]).


```
[takonnova@fedora ~]$ mkdir ski.places
[takonnova@fedora ~]$ mv equipment ski.places
[takonnova@fedora ~]$ cd ski.places
[takonnova@fedora ski.places]$ ls
equipment
[takonnova@fedora ski.places]$
```

Рис. 3.3: Создание директории ski.places

```
[takonnova@fedora ~]$ mkdir ski.places
[takonnova@fedora ~]$ mv equipment ski.places
[takonnova@fedora ~]$ cd ski.places
[takonnova@fedora ski.places]$ ls
equipment
[takonnova@fedora ski.places]$
```

Рис. 3.4: Перемещение файла equipment в каталог ski.places

Создаём в домашнем каталоге файл abc1 командой touch и копируем его в каталог ski.places под названием equiplist2. Выполняем проверку (рис. [3.5]).

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

Рис. 3.5: Создание файла abc1 и копирование его в каталог ski.places под названием equiplist2

Создаём каталог с именем equipment в каталоге ski.places (рис. [3.6]).

```
[takonnova@fedora ski.places]$ mkdir equiplist
[takonnova@fedora ski.places]$
```

Рис. 3.6: Создание каталога equipment в каталоге ski.places

Перемещаем файлы `equiplist` и `equiplist2` из каталога `ski.places` в подкаталог `equipment` (рис. [3.7]).

```
[takonnova@fedora ski.places]$ cd
[takonnova@fedora ~]$ mv ski.places/equipment equiplist
[takonnova@fedora ~]$ mv ski.places/equipment2 equiplist
[takonnova@fedora ~]$ ls
```

Рис. 3.7: Перемещение файлов в подкаталог `equipment`

[part 2] (image/7_2.png){ #fig:008 width=70% }

Создаём каталог `newdir`. Далее перемещаем его в каталог `ski.places` под названием `plans` (рис. [3.8]).

```
шаблоны
[takonnova@fedora ~]$ cd
[takonnova@fedora ~]$ mkdir newdir_1
[takonnova@fedora ~]$ mv newdir_1 ski.places
[takonnova@fedora ~]$ mv ski.places/newdir_1 ski.places/plans
[takonnova@fedora ~]$ cd ski.places/
[takonnova@fedora ski.places]$ ls
equiplist plans
[takonnova@fedora ski.places]$
```

Рис. 3.8: Создание каталога `newdir` и перемещение его в каталог `ski.places` под названием `plans`

Создаём 2 каталога (`australia` и `play`) и 2 файла (`my_os` и `feathers`) (рис. [3.9]).

```

[takonnova@fedora ski.plases]$ cd
[takonnova@fedora ~]$ mkdir australia
[takonnova@fedora ~]$ mkdir play
[takonnova@fedora ~]$ mkdir my_os
[takonnova@fedora ~]$ mc

[takonnova@fedora ~]$ touch my_os
[takonnova@fedora ~]$ touch feathers
[takonnova@fedora ~]$ ls
abcl
australia
bin
'$'\033'[D'$'\033'[D'$'\033'[D'$'\033'[
'\033'[D'$'\033'[C'$'\033'[C'$'\033'[C'$
33'[C'$'\033'[C'$'\033'[C'$'\033'[C'$'\0
'[C'$'\033'[C'$'\033'[C'$'\033'[D'$'\033'
'$'\033'[D'$'\033'[D'$'\033'[D'$'\033'[D
'$'\033'[D'$'\033'[D'$'\033'[D'$'\033'[
'\033'[D'$'\033'[C'$'\033'[C'$'\033'[C'$
33'[C'$'\033'[C'$'\033'[C'$'\033'[C'$'\0
'[C'$'\033'[C'$'\033'[C'$'\033'[D'$'\033'
'$'\033'[D'$'\033'[D'$'\033'[D'$'\033'[D
equiplist
feathers
my_os
newdir
newdir2
pandoc-2.19.2
pandoc-2.19.2-linux-amd64.tar.gz
pandoc-2.19-linux-amd64.tar.gz
pandoc-crossref
pandoc-crossref.1
pandoc-crossref-Linux.tar.xz
pandoc-crossref-Linux.tar.xz.1
play
ski.plases
work

```

Рис. 3.9: Создание каталогов и файлов

Далее присвоим каждому из каталогов и файлов определённые права доступа (рис. [3.10]).

[illegible]

Рис. 3.10: Присваивание прав доступа

Просматриваем содержимое файла `passwd` с помощью команды `cat` (рис. [3.11]).

```

takonnaova@fedora ~]$ cd
[takonnaova@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
dbus:x:81:81:System message bus:/:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
tss:x:59:59:Account used for TPM access:/dev/null:/sbin/nologin

```

Рис. 3.11: Просмотр содержимого файла `passwd`

Скопируем файл feathers в file.old (рис. [3.12]).

```
[takonnova@fedora ~]$ cp feathers file.old
[takonnova@fedora ~]$ ls
abc1
australia
bin
'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D
33''[D'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'$'\03
'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'
3''[C'$'\033''[C'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033
'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'
'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D
33''[D'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'$'\03
'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'$'\033''[C'
3''[C'$'\033''[C'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033
'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D'$'\033''[D
equiplist
feathers
file.old
my_os
newdir
ti
```

Рис. 3.12: Копирование файла feathers в file.old

Переместим файл file.old в каталог play (рис. [3.13]).

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

Рис. 3.13: Перемещение файла file.old в каталог play

Скопируем каталог play в каталог fun (рис. [3.14]).

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

Рис. 3.14: Копирование каталога play в каталог fun

Переместим каталог fun в каталог play и назовём его games (рис. [3.15]).

```

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

```

Рис. 3.15: 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. [3.16]).

```

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

```

Рис. 3.16: 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. [3.17]).

```

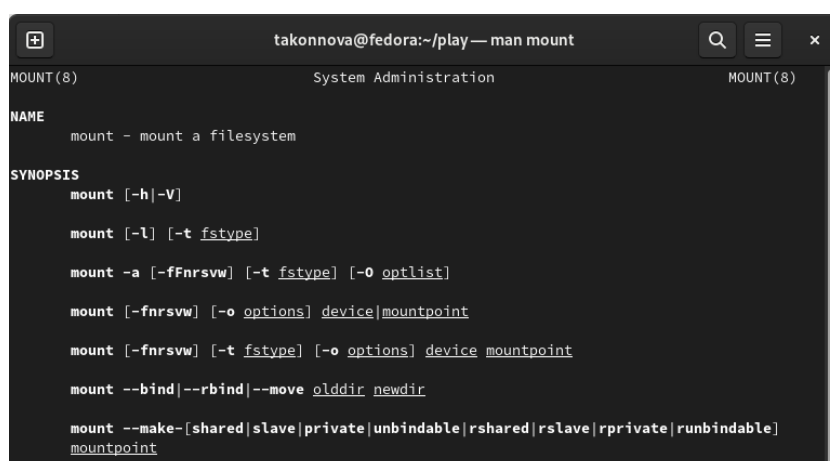
[takonnova@fedora ~]$ chmod u-x play
[takonnova@fedora ~]$ cd play
bash: cd: play: Отказано в доступе
[takonnova@fedora ~]$ chmod u+x play
[takonnova@fedora ~]$ cd play/
[takonnova@fedora play]$ ls
file.old  games
[takonnova@fedora play]$

```

Рис. 3.17: 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. [3.18]).



```

takonnova@fedora:~/play — 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

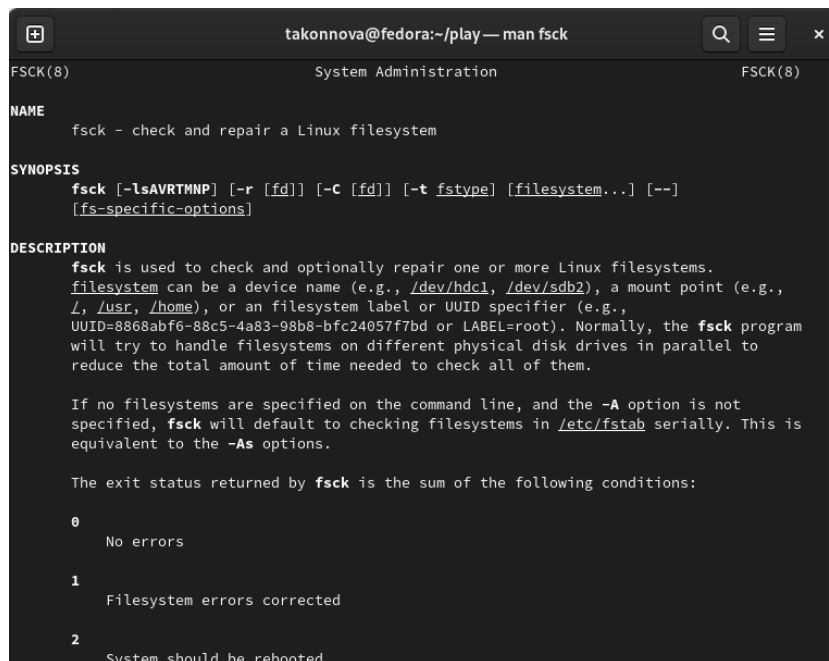
    mount --bind|--rbind|--move olddir newdir

    mount --make-[shared|slave|private|unbindable|rshared|rsave|rprivate|runbindable]
    mountpoint

```

Рис. 3.18: `man mount`

С помощью команды `fsck` можно проверить (а в ряде случаев восстановить) целостность файловой системы (рис. [3.19]).



```
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:

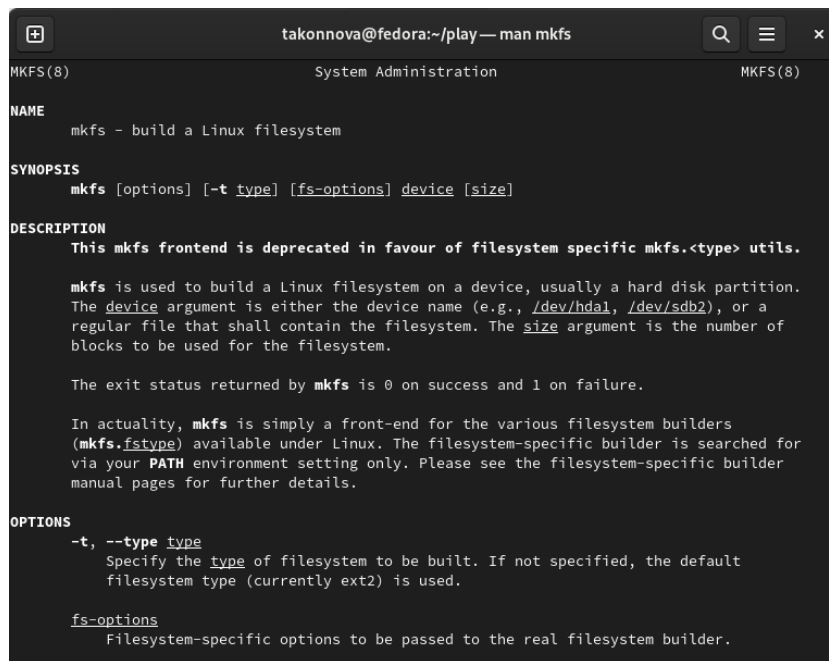
    0      No errors

    1      Filesystem errors corrected

    2      System should be rebooted
```

Рис. 3.19: 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. [3.20]).



```

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.
```

Рис. 3.20: 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. [3.21]).

```

+ takonnova@fedora:~/play — 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
```

Рис. 3.21: man kill

4 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.

5 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.