

Анализ файловой системы Linux. Команды для работы с файлами и каталогами

Ыбырай Роза

Содержание

Цель работы	1
Выполнение лабораторной работы.....	1
Выполнение всех пример, приведённых в первой части описания лабораторной работы	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.

Выполнение лабораторной работы

Выполнение всех пример, приведённых в первой части описания лабораторной работы

1. Копирование файлов и каталогов

```
rozaybyrai@rihbihrayj:~  
[rozaybyrai@rihbihrayj ~]$ cd  
[rozaybyrai@rihbihrayj ~]$ touch abc1  
[rozaybyrai@rihbihrayj ~]$ cp abc1 april  
[rozaybyrai@rihbihrayj ~]$ cp abc1 may  
[rozaybyrai@rihbihrayj ~]$ mkdir monthly  
[rozaybyrai@rihbihrayj ~]$ cp april may monthly  
[rozaybyrai@rihbihrayj ~]$ cp monthly/may monthly/june  
[rozaybyrai@rihbihrayj ~]$ ls monthly  
april  june  may  
[rozaybyrai@rihbihrayj ~]$ mkdir monthly.00  
[rozaybyrai@rihbihrayj ~]$ cp -r monthly.00 /tmp  
[rozaybyrai@rihbihrayj ~]$ cp -r monthly monthly.00  
[rozaybyrai@rihbihrayj ~]$ cp -r monthly.00 /tmp  
[rozaybyrai@rihbihrayj ~]$
```

Рис.1:

2. Перемещение и переименование файлов и каталогов

```
[rozaybyrai@rihbihrayj ~]$ cd  
[rozaybyrai@rihbihrayj ~]$ mv april july  
[rozaybyrai@rihbihrayj ~]$ mv july monthly.00  
[rozaybyrai@rihbihrayj ~]$ ls monthly.00  
july  monthly  
[rozaybyrai@rihbihrayj ~]$ mv monthly.00 monthly.01  
[rozaybyrai@rihbihrayj ~]$ mkdir reports  
[rozaybyrai@rihbihrayj ~]$ mv monthly.01 reports  
[rozaybyrai@rihbihrayj ~]$ mv reports/monthly.01 reports/monthly  
[rozaybyrai@rihbihrayj ~]$
```

Рис.2:

3. Права доступа

```
rozaybyrai@rihbihrayj:~  
[rozaybyrai@rihbihrayj ~]$ cd  
[rozaybyrai@rihbihrayj ~]$ touch may  
[rozaybyrai@rihbihrayj ~]$ ls -l may  
-rw-r--r--. 1 rozaybyrai rozaybyrai 0 сен  4 01:15 may  
[rozaybyrai@rihbihrayj ~]$ chmod u+x may  
[rozaybyrai@rihbihrayj ~]$ ls -l may  
-rwxr--r--. 1 rozaybyrai rozaybyrai 0 сен  4 01:15 may  
[rozaybyrai@rihbihrayj ~]$ chmod u-x may  
[rozaybyrai@rihbihrayj ~]$ ls -l may  
-rw-r--r--. 1 rozaybyrai rozaybyrai 0 сен  4 01:15 may  
[rozaybyrai@rihbihrayj ~]$ cd  
[rozaybyrai@rihbihrayj ~]$ mkdir monthly  
mkdir: невозможно создать каталог «monthly»: Файл существует  
[rozaybyrai@rihbihrayj ~]$ chmod g-r, o-r monthly  
chmod: неверный режим: «g-r,»  
По команде «chmod --help» можно получить дополнительную информацию.  
[rozaybyrai@rihbihrayj ~]$ chmod g-r, o-r monthly/  
chmod: неверный режим: «g-r,»  
По команде «chmod --help» можно получить дополнительную информацию.  
[rozaybyrai@rihbihrayj ~]$ chmod g-r, o-r monthly  
chmod: неверный режим: «g-r,»  
По команде «chmod --help» можно получить дополнительную информацию.  
[rozaybyrai@rihbihrayj ~]$ chmod g-r monthly  
[rozaybyrai@rihbihrayj ~]$ chmod o-r monthly  
[rozaybyrai@rihbihrayj ~]$ cd  
[rozaybyrai@rihbihrayj ~]$ touch abc1  
[rozaybyrai@rihbihrayj ~]$ chmod g+w abc1
```

##

Выполнение следующих действий

1. Скопирование файла `/usr/include/sys/io.h` в домашний каталог и назовим его `equipment`. Если файла `io.h` нет, то используем любой другой файл в каталоге `/usr/include/sys/` вместо него. В домашнем каталоге создаем директорию `~/ski.places`. Переместим файл `equipment` в каталог `~/ski.places`. Переименуем файл `~/ski.places/equipment` в `~/ski.places/equiplist`. Создаем в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовем его `equiplist2`. Создаем каталог с именем `equipment` в каталоге `~/ski.places`. Переместим файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`. Создаем и переместите каталог `~/newdir` в каталог `~/ski.places` и назовите его `plans`.

```
rozaybyrai@rihbihrayj:~  
[rozaybyrai@rihbihrayj ~]$ cp /usr/include/sys/io.h/  
cp: после '/usr/include/sys/io.h/' пропущен операнд, задающий целевой файл  
По команде «cp --help» можно получить дополнительную информацию.  
[rozaybyrai@rihbihrayj ~]$ cp /usr/include/sys/io.h~  
cp: после '/usr/include/sys/io.h~' пропущен операнд, задающий целевой файл  
По команде «cp --help» можно получить дополнительную информацию.  
[rozaybyrai@rihbihrayj ~]$ cp /usr/include/sys/  
cp: после '/usr/include/sys/' пропущен операнд, задающий целевой файл  
По команде «cp --help» можно получить дополнительную информацию.  
[rozaybyrai@rihbihrayj ~]$ cp /usr/include/sys ~  
cp: не указан -r; пропускается каталог '/usr/include/sys'  
[rozaybyrai@rihbihrayj ~]$ cp -r /usr/include/sys ~  
[rozaybyrai@rihbihrayj ~]$ mv sys equipment  
[rozaybyrai@rihbihrayj ~]$ mkdir ski.plases  
[rozaybyrai@rihbihrayj ~]$ mv equipment ski.plases  
[rozaybyrai@rihbihrayj ~]$ mv ski.plases/equipment ski.plases/equiplist  
[rozaybyrai@rihbihrayj ~]$ touch abc1  
[rozaybyrai@rihbihrayj ~]$ cp abc1 ski.plases/equiplist2  
[rozaybyrai@rihbihrayj ~]$ cd ski.plases  
[rozaybyrai@rihbihrayj ski.plases]$ mkdir equipment  
[rozaybyrai@rihbihrayj ski.plases]$ mv equiplist equipment  
[rozaybyrai@rihbihrayj ski.plases]$ mv equiplist2 equipment  
[rozaybyrai@rihbihrayj ski.plases]$ cd  
[rozaybyrai@rihbihrayj ~]$ mkdir newdir  
[rozaybyrai@rihbihrayj ~]$ mv newdir ski.plases  
[rozaybyrai@rihbihrayj ~]$ mv ski.plases/newdir ski.plases/plans  
[rozaybyrai@rihbihrayj ~]$
```

Рис.4:

2. Определим 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

```
rozaybyrai@rihbihrayj:~  
[rozaybyrai@rihbihrayj ~]$ touch australia  
[rozaybyrai@rihbihrayj ~]$ touch play  
[rozaybyrai@rihbihrayj ~]$ touch my_os  
[rozaybyrai@rihbihrayj ~]$ touch feathers  
[rozaybyrai@rihbihrayj ~]$ chmod 744 australia  
[rozaybyrai@rihbihrayj ~]$ chmod 711 play  
[rozaybyrai@rihbihrayj ~]$ chmod 544 my_os  
[rozaybyrai@rihbihrayj ~]$ chmod 664 feathers  
[rozaybyrai@rihbihrayj ~]$ ls -l  
итого 0  
-rw-rw-r--. 1 rozaybyrai rozaybyrai  0 сен  4 01:29 abc1  
-rwxr--r--. 1 rozaybyrai rozaybyrai  0 сен  4 01:38 australia  
-rw-rw-r--. 1 rozaybyrai rozaybyrai  0 сен  4 01:38 feathers  
-rw-r--r--. 1 rozaybyrai rozaybyrai  0 сен  4 01:15 may  
drwx--x--x. 1 rozaybyrai rozaybyrai 24 сен  4 01:06 monthly  
-r-xr--r--. 1 rozaybyrai rozaybyrai  0 сен  4 01:38 my_os  
-rwx--x--x. 1 rozaybyrai rozaybyrai  0 сен  4 01:38 play  
drwxr-xr-x. 1 rozaybyrai rozaybyrai 14 сен  4 01:12 reports  
drwxr-xr-x. 1 rozaybyrai rozaybyrai 28 сен  4 01:34 ski_places  
drwxr-xr-x. 1 rozaybyrai rozaybyrai  0 сен  2 21:35 Видео  
drwxr-xr-x. 1 rozaybyrai rozaybyrai  0 сен  2 21:35 Документы  
drwxr-xr-x. 1 rozaybyrai rozaybyrai 434 сен  3 12:57 Загрузки  
drwxr-xr-x. 1 rozaybyrai rozaybyrai  50 сен  2 21:38 Изображения  
drwxr-xr-x. 1 rozaybyrai rozaybyrai  0 сен  2 21:35 Музыка  
drwxr-xr-x. 1 rozaybyrai rozaybyrai  0 сен  2 21:35 Общедоступные  
drwxr-xr-x. 1 rozaybyrai rozaybyrai  0 сен  2 21:35 'Рабочий стол'  
drwxr-xr-x. 1 rozaybyrai rozaybyrai  0 сен  2 21:35 Шаблоны
```

Рис.5:

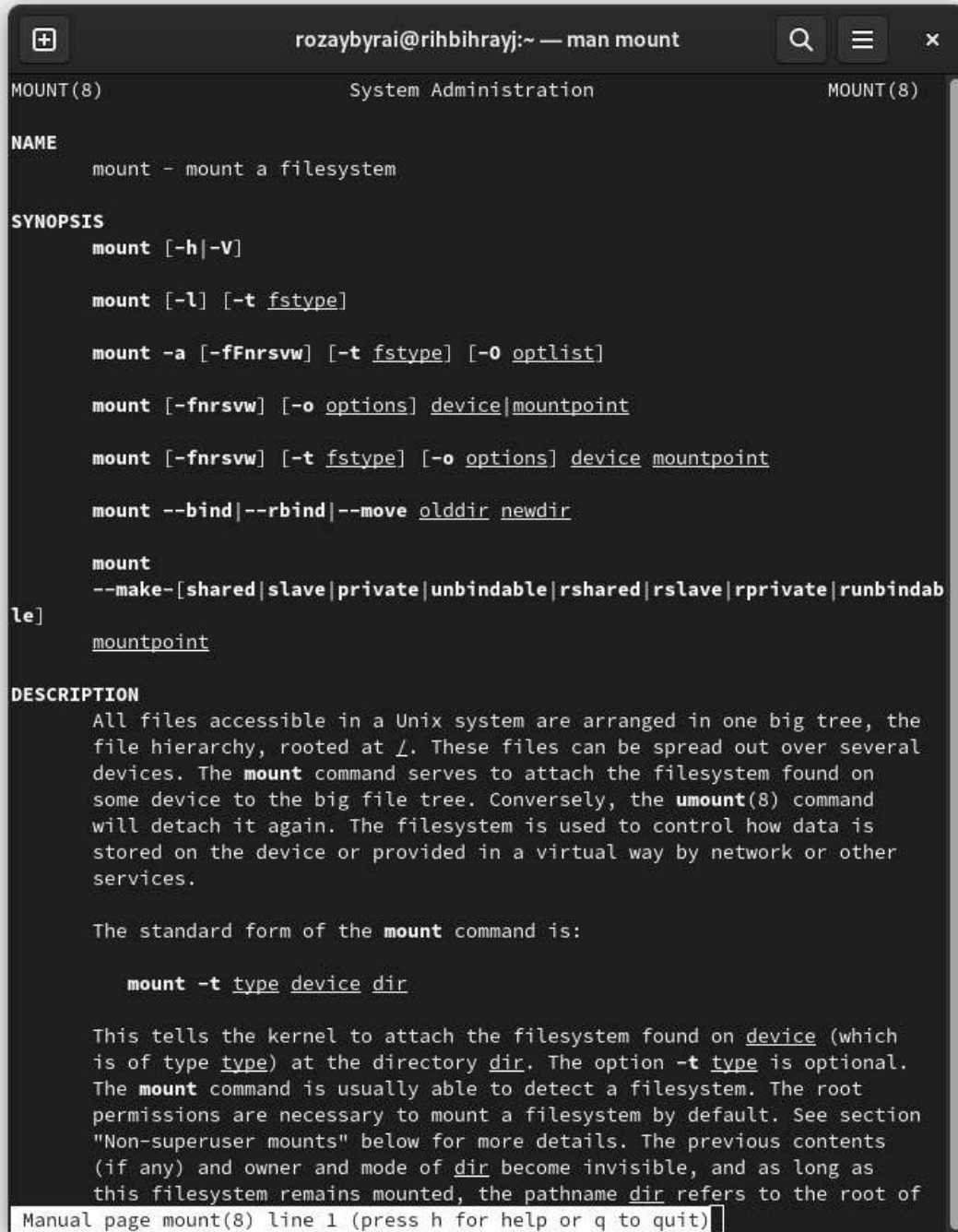
3.

- Просмотрим содержимое файла /etc/password.
- Скопируем файл ~/feathers в файл ~/file.old.
- Переместим файл ~/file.old в каталог ~/play.
- Скопируем каталог ~/play в каталог ~/fun.
- Переместим каталог ~/fun в каталог ~/play и назовите его games.
- Лишим владельца файла ~/feathers права на чтение.
- Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat? Отказано в доступе.
- Что произойдёт, если вы попытаетесь скопировать файл ~/feathers? Отказано в доступе.
- Даем владельцу файла ~/feathers право на чтение.
- Лишим владельца каталога ~/play права на выполнение.
- Перейдем в каталог ~/play. Что произошло?
- Даем владельцу каталога ~/play право на выполнение.

```
rozaybyrai@rihbihrayj:~  
[rozaybyrai@rihbihrayj ~]$ 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-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  
qemu:x:107:107:qemu user:/:sbin/nologin  
polkitd:x:998:997:User for polkitd:/:sbin/nologin  
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin  
geoclue:x:997:996:User for geoclue:/var/lib/geoclue:/sbin/nologin  
nm-openconnect:x:996:995:NetworkManager user for OpenConnect:/:sbin/nologin  
colord:x:995:994:User for colord:/var/lib/colord:/sbin/nologin  
usbmuxd:x:113:113:usbmuxd user:/:sbin/nologin
```

```
rozaybyrai@rihbihrayj:~  
[rozaybyrai@rihbihrayj] ~]$ cp feathers file.old.  
[rozaybyrai@rihbihrayj] ~]$ mv file.old play  
mv: не удалось выполнить stat для 'file.old': Нет такого файла или каталога  
[rozaybyrai@rihbihrayj] ~]$ mv file.old. play  
[rozaybyrai@rihbihrayj] ~]$ cp play fun  
[rozaybyrai@rihbihrayj] ~]$ mv fun play/games  
mv: не удалось выполнить stat для 'play/games': Это не каталог  
[rozaybyrai@rihbihrayj] ~]$ mv fun/ play/ games  
mv: цель 'games': Нет такого файла или каталога  
[rozaybyrai@rihbihrayj] ~]$ mv fun/ play/games  
mv: не удалось выполнить stat для 'fun/': Это не каталог  
[rozaybyrai@rihbihrayj] ~]$ mv fun play  
[rozaybyrai@rihbihrayj] ~]$ mv play games  
[rozaybyrai@rihbihrayj] ~]$ chmod u-r feathers  
[rozaybyrai@rihbihrayj] ~]$ cat feathers  
cat: feathers: Отказано в доступе  
[rozaybyrai@rihbihrayj] ~]$ cp feathers  
cp: после 'feathers' пропущен операнд, задающий целевой файл  
По команде «cp --help» можно получить дополнительную информацию.  
[rozaybyrai@rihbihrayj] ~]$ cp feathers birds  
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе  
[rozaybyrai@rihbihrayj] ~]$ chmod u+r feathers  
[rozaybyrai@rihbihrayj] ~]$ chmod u-x play  
chmod: невозможно получить доступ к 'play': Нет такого файла или каталога  
[rozaybyrai@rihbihrayj] ~]$ chmod u-x play/  
chmod: невозможно получить доступ к 'play/': Нет такого файла или каталога  
[rozaybyrai@rihbihrayj] ~]$ mv games play  
[rozaybyrai@rihbihrayj] ~]$ mv fun games  
mv: не удалось выполнить stat для 'fun': Нет такого файла или каталога  
[rozaybyrai@rihbihrayj] ~]$ chmod u-x play  
[rozaybyrai@rihbihrayj] ~]$ cd play  
bash: cd: play: Это не каталог  
[rozaybyrai@rihbihrayj] ~]$ chmod u+x play
```


4. Прочтение man по командам mount, fsck, mkfs, kill



```
rozaybyrai@rihbihrayj:~ — 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

    This tells the kernel to attach the filesystem found on device (which
    is of type type) at the directory dir. The option -t type is optional.
    The mount command is usually able to detect a filesystem. The root
    permissions are necessary to mount a filesystem by default. See section
    "Non-superuser mounts" below for more details. The previous contents
    (if any) and owner and mode of dir become invisible, and as long as
    this filesystem remains mounted, the pathname dir refers to the root of
    Manual page mount(8) line 1 (press h for help or q to quit)
```




le]

mountpoint**DESCRIPTION**

All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at `/`. These files can be spread out over several devices. The **mount** command serves to attach the filesystem found on some device to the big file tree. Conversely, the **umount**(8) command will detach it again. The filesystem is used to control how data is stored on the device or provided in a virtual way by network or other services.

The standard form of the **mount** command is:

```
mount -t type device dir
```

This tells the kernel to attach the filesystem found on device (which is of type type) at the directory dir. The option **-t type** is optional. The **mount** command is usually able to detect a filesystem. The root permissions are necessary to mount a filesystem by default. See section "Non-superuser mounts" below for more details. The previous contents (if any) and owner and mode of dir become invisible, and as long as this filesystem remains mounted, the pathname dir refers to the root of the filesystem on device.

If only the directory or the device is given, for example:

```
mount /dir
```

then **mount** looks for a mountpoint (and if not found then for a device) in the `/etc/fstab` file. It's possible to use the **--target** or **--source** options to avoid ambiguous interpretation of the given argument. For example:

```
mount --target /mountpoint
```

The same filesystem may be mounted more than once, and in some cases (e.g., network filesystems) the same filesystem may be mounted on the same mountpoint multiple times. The **mount** command does not implement any policy to control this behavior. All behavior is controlled by the kernel and it is usually specific to the filesystem driver. The exception is **--all**, in this case already mounted filesystems are ignored (see **--all** below for more details).

Manual page mount(8) line 20 (press h for help or q to quit)



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
- 32
Checking canceled by user request
- 128
Shared-library error

The exit status returned when multiple filesystems are checked is the bit-wise OR of the exit statuses for each filesystem that is checked.

```
rozaybyrai@rihbihrayj:~ — man mkfs
MKFS(8) System Administration MKFS(8)

NAME
    mkfs - build a Linux filesystem

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

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

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

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

    In actuality, mkfs is simply a front-end for the various filesystem
    builders (mkfs.fstype) available under Linux. The filesystem-specific
    builder is searched for via your PATH environment setting only. Please
    see the filesystem-specific builder manual pages for further details.

OPTIONS
    -t, --type type
        Specify the type of filesystem to be built. If not specified, the
        default filesystem type (currently ext2) is used.

    fs-options
        Filesystem-specific options to be passed to the real filesystem
        builder.

    -V, --verbose
        Produce verbose output, including all filesystem-specific commands
        that are executed. Specifying this option more than once inhibits
        execution of any filesystem-specific commands. This is really only
        useful for testing.

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

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

```
rozaybyrai@rihbihrayj:~ — 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.

    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.

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

#

Выводы

В ходе выполнения данной 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.

...