

Отчет по лабораторной работе №5

Операционные системы

Лукина Ирина Олеговна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12
	Список литературы	13

Список иллюстраций

2.1	Выполнение примеров	6
2.2	equipment	6
2.3	ski.plases/	6
2.4	Перемещение	7
2.5	abc1	7
2.6	equipment	7
2.7	Перемещение	7
2.8	Каталог newdir	8
2.9	chmod	8
2.10	Копирование	8
2.11	Перемещение	8
2.12	fun	8
2.13	Перемещение	9
2.14	chmod 300	9
2.15	cat	9
2.16	ski.plases/	9
2.17	chmod 600	9
2.18	man mount	10
2.19	man fsck	10
2.20	man kill	11

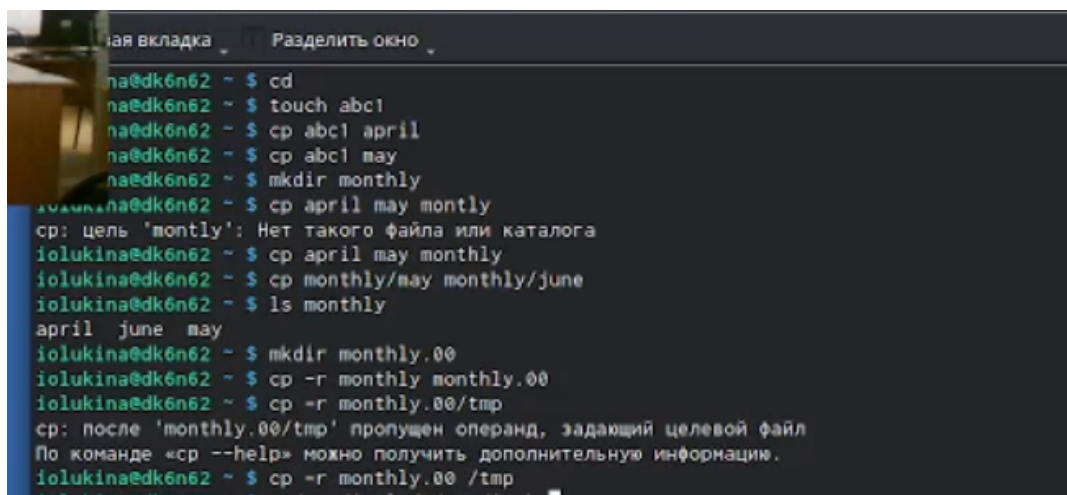
Список таблиц

1 Цель работы

Ознакомление с файловой системой Linux, ее 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 Выполнение лабораторной работы

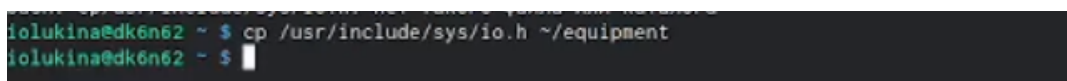
1. Выполняю примеры команд согласно ходу выполнения работы.



```
na@dk6n62 ~ $ cd
na@dk6n62 ~ $ touch abc1
na@dk6n62 ~ $ cp abc1 april
na@dk6n62 ~ $ cp abc1 may
na@dk6n62 ~ $ mkdir monthly
na@dk6n62 ~ $ cp april may montly
cp: цель 'montly': Нет такого файла или каталога
iolukina@dk6n62 ~ $ cp april may monthly
iolukina@dk6n62 ~ $ cp monthly/may monthly/june
iolukina@dk6n62 ~ $ ls monthly
april  june  may
iolukina@dk6n62 ~ $ mkdir monthly.00
iolukina@dk6n62 ~ $ cp -r monthly monthly.00
iolukina@dk6n62 ~ $ cp -r monthly.00/tmp
cp: после 'monthly.00/tmp' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
iolukina@dk6n62 ~ $ cp -r monthly.00 /tmp
iolukina@dk6n62 ~ $ cp /tmp/monthly.00/tmp/abc1 /tmp/abc1
```

Рис. 2.1: Выполнение примеров

2. Копирую файл и даю ему название через команду cp.



```
iolukina@dk6n62 ~ $ cp /usr/include/sys/io.h ~/equipment
iolukina@dk6n62 ~ $
```

Рис. 2.2: equipment

3. Создаю директорию через mkdir.



```
iolukina@dk6n62 ~ $ mkdir ski.places/
iolukina@dk6n62 ~ $
```

Рис. 2.3: ski.places/

4. Перемещаю файл в каталог командой mv и проверяю.

```
iolukina@dk6n62 ~ $ mv ski.places/equipment ski.places/equiplist
iolukina@dk6n62 ~ $ ls ski.places/
equiplist
```

Рис. 2.4: Перемещение

5. Создаю новый файл и копирую его в каталог, одновременно даю новое название.

```
iolukina@dk6n62 ~ $ touch abc1
iolukina@dk6n62 ~ $ cp abc1 ski.places/equiplist2
iolukina@dk6n62 ~ $ ls
abc1  GNUstep  memos    monthly.00  parentdir2  public_html  tmp      Документы  Музыка  Шаблоны
april  letters  misk     parentdir   parentdir3  R            work     Загрузки  Общедоступные
bin    may      monthly  parentdir1  public      ski.places   Видео    Изображения  'Рабочий стол'
```

Рис. 2.5: abc1

6. Создаю новый каталог в ski.places/

```
iolukina@dk6n62 ~ $ mkdir ski.places/equipment
iolukina@dk6n62 ~ $ c ski.places/
bash: c: команда не найдена
iolukina@dk6n62 ~ $ cd ski.places/
iolukina@dk6n62 ~/ski.places $ ls
equiplist  equiplist2  equipment
```

Рис. 2.6: equipment

7. Перемещаю файлы.

```
iolukina@dk6n62 ~/ski.places $ mv equiplist equipment/
iolukina@dk6n62 ~/ski.places $ mv equiplist2 equipment
iolukina@dk6n62 ~/ski.places $
```

Рис. 2.7: Перемещение

8. Создаю каталог newdir и перемещаю, дав новое имя.

```

iolukina@dk6n62 ~ $ mkdir newdir
iolukina@dk6n62 ~ $ mv newdir skiplaces/plans
mv: невозможно переместить 'newdir' в 'skiplaces/plans': Нет такого файла или каталога
iolukina@dk6n62 ~ $ mv newdir ski.places/plans
iolukina@dk6n62 ~ $ ls ski.places/
equipment  plans
iolukina@dk6n62 ~ $

```

Рис. 2.8: Каталог newdir

9. Создаю новые файлы и определяю опции команды chmod.

```

iolukina@dk6n62 ~ $ mkdir australia
iolukina@dk6n62 ~ $ mkdir play
iolukina@dk6n62 ~ $ touch my_os
iolukina@dk6n62 ~ $ touch feathers
iolukina@dk6n62 ~ $ chmod 711 australia
iolukina@dk6n62 ~ $ chmod 744 australia
iolukina@dk6n62 ~ $ chmod 711 play
iolukina@dk6n62 ~ $ chmod 544 my_os
iolukina@dk6n62 ~ $

```

Рис. 2.9: chmod

10. Копирую файл в другой файл.

```

iolukina@dk6n62 ~ $ cp feathers file.old
iolukina@dk6n62 ~ $

```

Рис. 2.10: Копирование

11. Перемещаю каталог play в fun.

```

iolukina@dk6n62 ~ $ mv file.old play/
iolukina@dk6n62 ~ $

```

Рис. 2.11: Перемещение

12. Создаю каталог fun и перемещаю его.

```

iolukina@dk6n62 ~ $ mkdir fun
iolukina@dk6n62 ~ $ cp -r play fun
iolukina@dk6n62 ~ $ ls fun/
play

```

Рис. 2.12: fun

13. Перемещаю в каталог play и даю название games.

```
iolukina@dk6n62 ~ $ mv fun play/games
iolukina@dk6n62 ~ $
```

Рис. 2.13: Перемещение

14. Лишаю владельца файла права на чтение.

```
iolukina@dk6n62 ~ $ chmod 300 feathers
```

Рис. 2.14: chmod 300

15. Через команду cat пробую проверить feathers.

```
iolukina@dk6n62 ~ $ cat feathers
cat: feathers: Отказано в доступе
```

Рис. 2.15: cat

16. Пробую скопировать.

```
iolukina@dk6n62 ~ $ cat feathers
cat: feathers: Отказано в доступе
iolukina@dk6n62 ~ $ cp feathers fun/
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
iolukina@dk6n62 ~ $
```

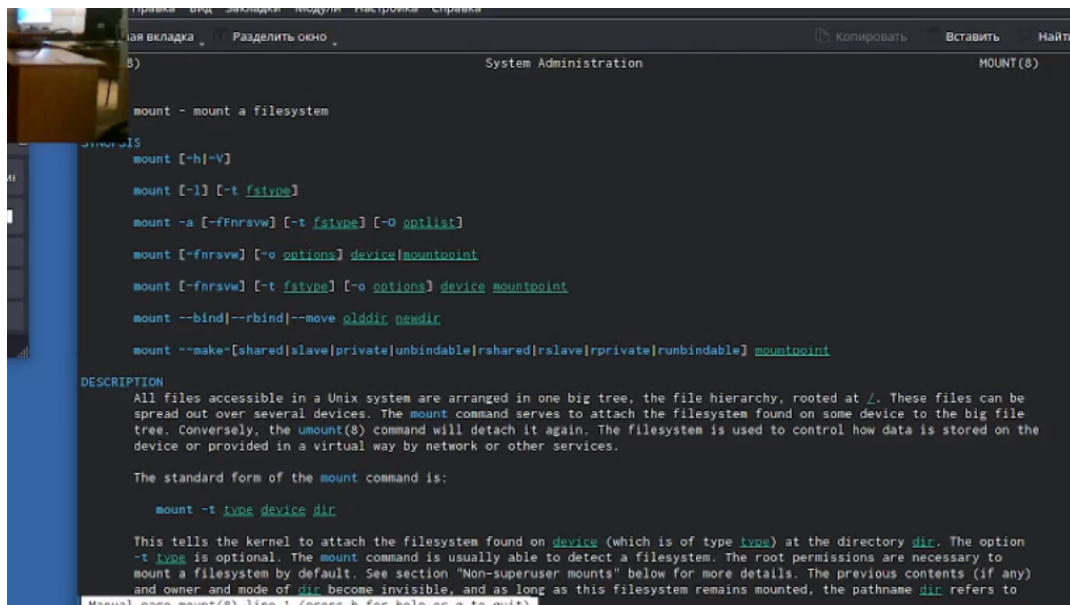
Рис. 2.16: ski.places/

17. Лишаю права на просмотр каталога play.

```
iolukina@dk6n62 ~ $ chmod 600 play/
iolukina@dk6n62 ~ $ cd play/
iolukina@dk6n62 ~/play $
```

Рис. 2.17: chmod 600

18. Просматриваю разные опции команд через man.



The screenshot shows a terminal window with the 'man mount' command executed. The window title is 'System Administration' and the subtitle is 'MOUNT(8)'. The content displays the manual page for the 'mount' command, including its syntax, description, and usage examples. The syntax section lists various options like [-h|-V], [-l] [-t fstype], [-a [-ffnrsvw] [-t fstype] [-O optlist]], [-f fstype] [-o options] device[mountpoint], [-t fstype] [-o options] device mountpoint, --bind|--rbind|--move olddir newdir, and --make[+shared|+slave|+private|+unbindable|+rshared|+rslave|+rprivate|+runbindable] mountpoint. The description explains that the mount command is used to attach a filesystem found on a device to a directory in the file hierarchy. The standard form of the mount command is shown as 'mount -t type device dir'. The usage section explains that the kernel tells the user 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 mounted filesystem.

```
mount - mount a filesystem

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.

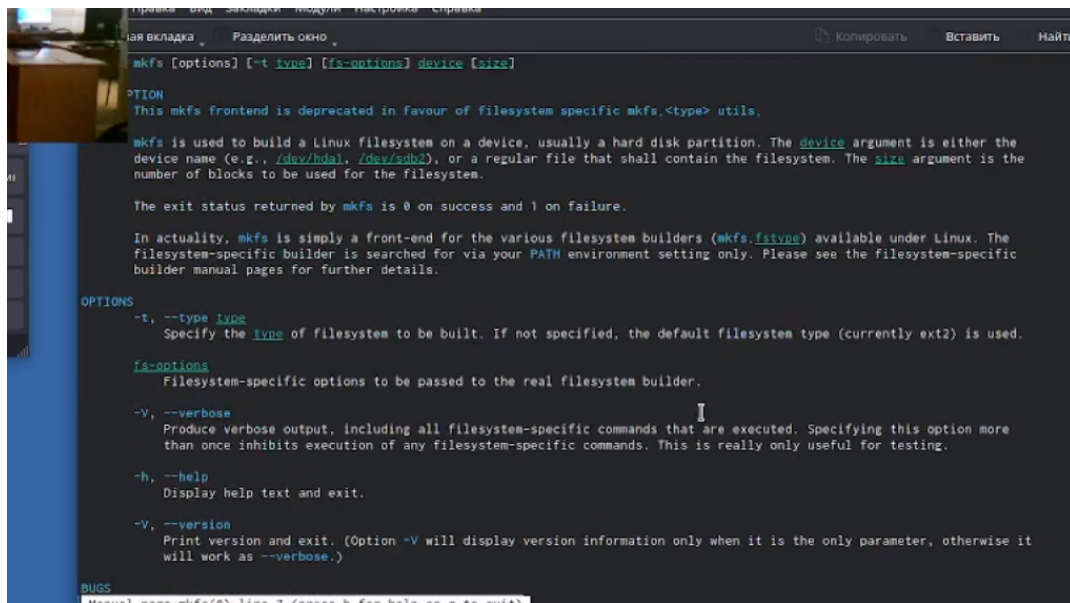
The standard form of the mount command is:

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 mounted filesystem.

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

Рис. 2.18: man mount



The screenshot shows a terminal window with the 'man mkfs' command executed. The window title is 'System Administration' and the subtitle is 'MKFS(8)'. The content displays the manual page for the 'mkfs' command, including its syntax, description, and usage examples. The syntax section lists various options like [-h|-V], [-l] [-t fstype] [-fs-options] device [size]. The description explains that 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. The options section lists -t, --type type (Specify the type of filesystem to be built. If not specified, the default filesystem type (currently ext2) is used.), -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.), and -V, --version (Print version and exit. (Option -V will display version information only when it is the only parameter, otherwise it will work as --verbose.)).

```
mkfs [options] [-t fstype] [-fs-options] device [size]

OPTION
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.

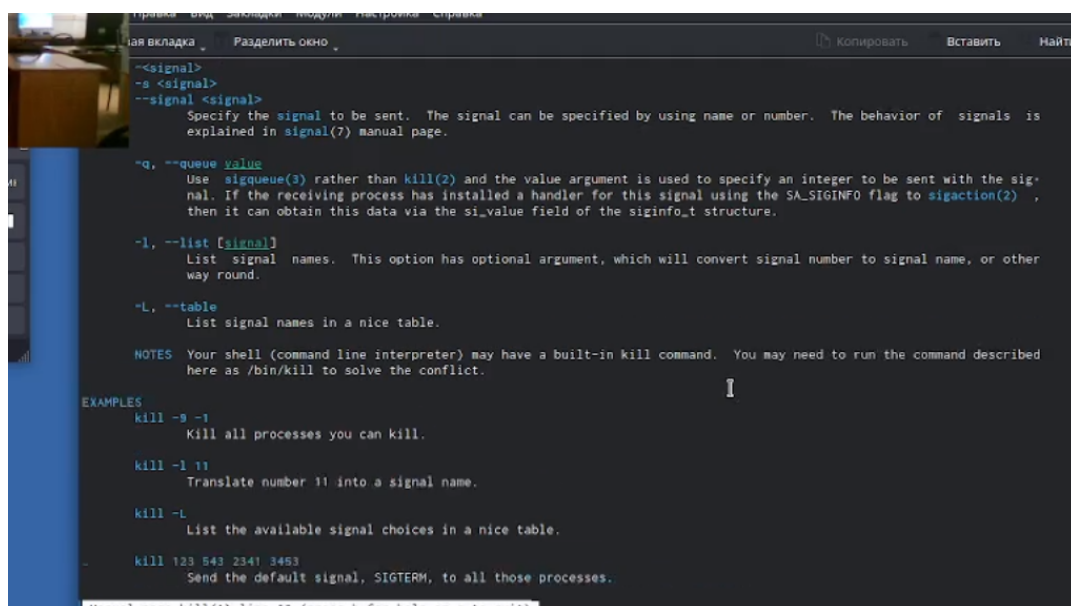
-V, --verbose
Produce verbose output, including all filesystem-specific commands that are executed. Specifying this option more than once inhibits execution of any filesystem-specific commands. This is really only useful for testing.

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

-V, --version
Print version and exit. (Option -V will display version information only when it is the only parameter, otherwise it will work as --verbose.)

BUGS
Manual page mkfs(8) line 7 (press h for help or q to quit)
```

Рис. 2.19: man fsck



```
~<signal>
~s <signal>
--signal <signal>
    Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is explained in signal(7) manual page.

-q, --queue <value>
    Use sigqueue(3) rather than kill(2) and the value argument is used to specify an integer to be sent with the signal. If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigaction(2), then it can obtain this data via the si_value field of the siginfo_t structure.

-l, --list [<signal>]
    List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.

-L, --table
    List signal names in a nice table.

NOTES Your shell (command line interpreter) may have a built-in kill command. You may need to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
kill -9 -1
    Kill all processes you can kill.

kill -l 11
    Translate number 11 into a signal name.

kill -L
    List the available signal choices in a nice table.

kill 123 543 2341 3453
    Send the default signal, SIGTERM, to all those processes.
```

Manual page kill(1) line 30 (space h for help or q to quit)

Рис. 2.20: man kill

3 Выводы

В ходе выполнения работы я приобрела практические навыки в работе с файловой системой Linux.

Список литературы