

# **Лабораторная работа №7**

**Дисциплина: Операционные системы**

Савостин Олег

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
<b>5</b>	<b>Выводы</b>	<b>16</b>
	<b>Список литературы</b>	<b>17</b>

# Список иллюстраций

4.1	Копирование файлов . . . . .	9
4.2	Копирование репозитория в другие репозитории . . . . .	9
4.3	Копирование репозитория . . . . .	10
4.4	Перемещение файлов . . . . .	10
4.5	Создание файлов . . . . .	10
4.6	Изменение права доступа . . . . .	10
4.7	fsck . . . . .	11
4.8	Создание нужных каталогов. . . . .	11
4.9	Выполнение команд . . . . .	11
4.10	Права доступа . . . . .	12
4.11	Содержимое файла . . . . .	12
4.12	Завершение выполнения задач . . . . .	13
4.13	mount . . . . .	13
4.14	fsck . . . . .	14
4.15	mkfs . . . . .	14
4.16	kill . . . . .	15

## **Список таблиц**

# 1 Цель работы

Целью работы является 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. Использование команд.

### 3 Теоретическое введение

Для создания текстового файла можно использовать команду `touch`. Формат команды: `1 touch имя-файла` Для просмотра файлов небольшого размера можно использовать команду `cat`. Формат команды: `1 cat имя-файла` Для просмотра файлов постранично удобнее использовать команду `less`. Формат команды: `1 less имя-файла` Следующие клавиши используются для управления процессом просмотра: `Space` — переход к следующей странице, `ENTER` — сдвиг вперёд на одну строку, `b` — возврат на предыдущую страницу, `h` — обращение за подсказкой, `q` — выход из режима просмотра файла. Команда `head` выводит по умолчанию первые 10 строк файла. Формат команды: `1 head [-n] имя-файла`, где `n` — количество выводимых строк. Команда `tail` выводит по умолчанию 10 последних строк файла. Формат команды: `1 tail [-n] имя-файла`, где `n` — количество выводимых строк.

Кулябов Д. С. и др. Операционные системы 47 5.2.2. Копирование файлов и каталогов Команда `cp` используется для копирования файлов и каталогов. Формат команды: `1 cp [-опции] исходный_файл целевой_файл` Примеры: 1. Копирование файла в текущем каталоге. Скопировать файл `~/abc1` в файл `april` и в файл `may`: `1 cd 2 touch abc1 3 cp abc1 april 4 cp abc1 may 2` Копирование нескольких файлов в каталог. Скопировать файлы `april` и `may` в каталог `monthly`: `1 mkdir monthly 2 cp april may monthly 3` Копирование файлов в произвольном каталоге. Скопировать файл `monthly/may` в файл с именем `june`: `1 cp monthly/may monthly/june 2 ls monthly` Опция `i` в команде `cp` выведет на экран запрос подтверждения о перезаписи файла. Для рекурсивного копирования каталогов, содержащих файлы, используется команда `cp` с опцией `r`. Примеры: 1. Копирование каталогов в

текущем каталоге. Скопировать каталог monthly в каталог monthly.00: 1 mkdir  
monthly.00 2 cp -r monthly monthly.00 2. Копирование каталогов в произвольном  
каталоге. Скопировать каталог monthly.00 в каталог /tmp 1 cp -r monthly.00 /tmp



## 4 Выполнение лабораторной работы

Выполняю все примеры в лабораторной работе. (рис. 4.10) (рис. 4.2) (рис. 4.3) (рис. 4.4) (рис. 4.5) (рис. 4.6) (рис. 4.7)

```
[sudo] пароль для esavostin:
root@vbox:~# touch abc1
root@vbox:~# cd abc1 april
-bash: cd: слишком много аргументов
root@vbox:~# cp abc1 april
root@vbox:~# cp abc1 may
root@vbox:~# ls
abc1      april  blog      LABS      may      pass2.txt  password.txt  texput.log      work
anaconda-ks.cfg  bin    git-extended  LICENSE  package.json  pass.txt    quit          theme-academic-cv
root@vbox:~# mkdir monthly
root@vbox:~# cp april may monthly
root@vbox:~# ls monthly
april  may
root@vbox:~# cp monthly/may monthly/june
root@vbox:~# ls monthly
april  june  may
root@vbox:~# mkdir monthly/00
root@vbox:~# cp -r monthly monthly.00
root@vbox:~# cp -r monthly.00 /tmp
cp: не удалось выполнить stat для 'monthly.00': Нет такого файла или каталога
root@vbox:~# ls
abc1      april  blog      LABS      may      monthly.00  pass2.txt  password.txt  texput.log      work
anaconda-ks.cfg  bin    git-extended  LICENSE  monthly  package.json  pass.txt    quit          theme-academic-cv
```

Рис. 4.1: Копирование файлов

```
root@vbox:~# ls monthly
00  april  june  may
root@vbox:~# cp -r monthly monthly.00
root@vbox:~# ls monthly
00  april  june  may
root@vbox:~# mc
root@vbox:~#
```

Рис. 4.2: Копирование репозитория в другие репозитории

```

root@vbox:~# cp -r monthly.00 /tmp
root@vbox:~# ls /tmp
monthly.00

```

Рис. 4.3: Копирование репозитория

```

root@vbox:~# cd
root@vbox:~# mv april july
root@vbox:~# ls monthly
00 april june may
root@vbox:~# ls
abc1 bin git-extended LABS may monthly.00 pass2.txt password.txt texput.log work
anaconda-ks.cfg blog july LICENSE monthly package.json pass2.txt quit theme-academic-cv
root@vbox:~# mv july monthly.00
root@vbox:~# ls
abc1 bin git-extended LICENSE monthly package.json pass.txt quit theme-academic-cv
anaconda-ks.cfg blog LABS may monthly.00 pass2.txt password.txt texput.log work
root@vbox:~# ls monthly.00
00 april july june may monthly
root@vbox:~#

```

Рис. 4.4: Перемещение файлов

```

root@vbox:~# touch may
root@vbox:~# ls -l may
-rw-r--r--. 1 root root 0 map 24 10:23 may
root@vbox:~# chmod u+x may
root@vbox:~# ls -l may
ls: невозможно получить доступ к '-': Нет такого файла или каталога
ls: невозможно получить доступ к 'l': Нет такого файла или каталога
may
root@vbox:~# ls -l may
-rwxr--r--. 1 root root 0 map 24 10:23 may
root@vbox:~#

```

Рис. 4.5: Создание файлов

```

root@vbox:~# chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
root@vbox:~# chmod o-r monthly
root@vbox:~# ls -l monthly
итого 0
drwxr-xr-x. 1 root root 0 map 24 10:19 00
-rw-r--r--. 1 root root 0 map 24 10:17 april
-rw-r--r--. 1 root root 0 map 24 10:18 june
-rw-r--r--. 1 root root 0 map 24 10:17 may
root@vbox:~# chmod abc1
chmod: пропущен операнд после «abc1»
По команде «chmod --help» можно получить дополнительную информацию.
root@vbox:~# chmod g+w abc1
root@vbox:~#

```

Рис. 4.6: Изменение права доступа

```

root@vbox:~# fsck /dev/sda1
fsck from util-linux 2.40.4
e2fsck 1.47.1 (20-May-2024)
ext2fs_open2: Bad magic number in super-block
fsck.ext2: Superblock invalid, trying backup blocks...
fsck.ext2: Bad magic number in super-block while trying to open /dev/sda1

The superblock could not be read or does not describe a valid ext2/ext3/ext4
filesystem. If the device is valid and it really contains an ext2/ext3/ext4
filesystem (and not swap or ufs or something else), then the superblock
is corrupt, and you might try running e2fsck with an alternate superblock:
    e2fsck -b 8193 <device>
or
    e2fsck -b 32768 <device>

root@vbox:~#

```

Рис. 4.7: fsck

Выполняю вторую часть работы.(рис. 4.8)

```

root@vbox:~# cp /usr/include/sys/io.h ~/equipment
root@vbox:~# ls
abc1      bin      equipment  LAB5     may      monthly.00  pass2.txt  password.txt  texput.log  work
anaconda-ks.cfg  blog  git-extended  LICENSE  monthly  package.json  pass.txt   quit         theme-academic-cv
root@vbox:~# mkdir ski.places
root@vbox:~# mv ~/equipment ~/ski.places/
root@vbox:~# ls
abc1      bin      git-extended  LICENSE  monthly  package.json  pass.txt   quit         texput.log  work
anaconda-ks.cfg  blog  LAB5         may      monthly.00  pass2.txt    password.txt  ski.places  theme-academic-cv
root@vbox:~# ld sky.places
ld: невозможно найти sky.places.: Нет такого файла или каталога
root@vbox:~# ls sky.places
ls: невозможно получить доступ к 'sky.places.': Нет такого файла или каталога
root@vbox:~# ls ski.places
equipment
root@vbox:~# touch ~/ski.places/equipment ls ~/ski.places/equiplist
root@vbox:~# ls ski.places
equiplist  equipment
root@vbox:~# mc

root@vbox:~# mv ~/ski.places/equipment ~/ski.places/equiplist
root@vbox:~# ls ski.places
equiplist
root@vbox:~#

```

Рис. 4.8: Создание нужных каталогов.

Следую указаниям в лабораторной работе (рис. 4.9)

```

root@vbox:~# cp ~/abc1 ~/ski.places/equiplist2
root@vbox:~# ls ski.places
equiplist  equiplist2
root@vbox:~# mkdir ~/ski.places/equipment
root@vbox:~# ls
abc1      bin      git-extended  LICENSE  may      monthly.00  pass2.txt  password.txt  ski.places  theme-academic-cv
anaconda-ks.cfg  blog  LAB5         ls      monthly  package.json  pass.txt   quit         texput.log  work
root@vbox:~# ls ski.places
equiplist  equiplist2  equipment
root@vbox:~# mv ~/ski.places/equiplist, ~/ski.places/equiplist2 ~/ski.places/equipment/
mv: не удалось выполнить stat для '/root/ski.places/equiplist,': Нет такого файла или каталога
root@vbox:~# mv ~/ski.places/equiplist ~/ski.places/equipment/
mv: не удалось выполнить stat для '/root/ski.places/equiplist2': Нет такого файла или каталога
root@vbox:~# ls ski.places
equipment
root@vbox:~# ls ~/ski.places/equipment/
equiplist  equiplist2
root@vbox:~# mkdir newdir
root@vbox:~# mv ~/newdir ~/ski.places/.plans
root@vbox:~# ls
abc1      bin      git-extended  LICENSE  may      monthly.00  pass2.txt  password.txt  ski.places  theme-academic-cv
anaconda-ks.cfg  blog  LAB5         ls      monthly  package.json  pass.txt   quit         texput.log  work
root@vbox:~# ls ski.places
equipment  plans
root@vbox:~#

```

Рис. 4.9: Выполнение команд

Выполняю команды на выдачу конкретных прав для файлов (рис. 4.10)

```

root@vbox:~# chmod 744 australia
root@vbox:~# ls -l australia
-rw-r--r-- 1 root 0
root@vbox:~# ls -l australia
-rw-r--r-- 1 root 0
root@vbox:~# getfacl australia
# file: australia
# owner: root
# group: root
user::rw-
group::r--
other::r--

root@vbox:~# ls
abc1      australia  blog      LABS      ls         monthly   package.json  pass.txt  quit      texput.log  work
anaconda-ks.cfg  bin        git-extended  LICENSE  may        monthly.00  pass2.txt     password.txt  ski.places  theme-academic-cv

root@vbox:~# mkdir play
root@vbox:~# chmod 711 play
root@vbox:~# getfacl play
# file: play
# owner: root
# group: root
user::rwx
group::-x
other::-x

root@vbox:~# touch my_os
root@vbox:~# chmod 544 my_os
root@vbox:~# ls -l my_os
-r-xr-xr-- 1 root root 0 map 24 11:35 my_os
root@vbox:~# touch feathers
root@vbox:~# chmod 664 feathers
root@vbox:~# ls -l feathers
-rw-rw-r-- 1 root root 0 map 24 11:36 feathers
root@vbox:~#

```

Рис. 4.10: Права доступа

Проверяю содержимое файла passwd(рис. 4.11)

```

/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

```

Рис. 4.11: Содержимое файла

Завершаю выполнение задач (рис. 4.12)

```

root@vbox:~# cd ~
root@vbox:~# cp ~/feathers ~/file.old
root@vbox:~# mv ~/file.old ~/play
root@vbox:~# ls play
file.old
root@vbox:~# cp ~/play ~/fun
cp: не указан -r; пропускается каталог '/root/play'
root@vbox:~# cp -r ~/play ~/fun
root@vbox:~# ls fun
file.old
root@vbox:~# mv ~/fun ~/play/games
root@vbox:~# ls -l ~/feathers
-rw-rw-r--. 1 root root 0 map 24 11:36 /root/feathers
root@vbox:~# chmod 264 feathers
root@vbox:~# ls -l feathers
--w-rw-r--. 1 root root 0 map 24 11:36 feathers
root@vbox:~# cat feathers
root@vbox:~# cp ~/feathers ~/feathers2
root@vbox:~# chmod 664 feathers
root@vbox:~# ls -l feathers
-rw-rw-r--. 1 root root 0 map 24 11:36 feathers
root@vbox:~# ls -l play
итого 0
-rw-r--r--. 1 root root 0 map 24 11:38 file.old
dwx--x--x. 1 root root 16 map 24 11:39 games
root@vbox:~# getfacl play
# file: play
# owner: root
# group: root
user::rwx
group::--x
other::--x
root@vbox:~# chmod 611 play
root@vbox:~# cd play
root@vbox:~/play# cd -
/root
root@vbox:~# chmod 711 play

```

Рис. 4.12: Завершение выполнения задач

Теперь проверяю что делают команды `fsck mount mkfs kill` (рис. 4.13) (рис. 4.14) (рис. 4.15) (рис. 4.16)

```

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|rsshared|rslave|rsprivate|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 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 --make-[shared|slave|private|unbindable|rsshared|rslave|rsprivate|runbindable] options to specify the mount options. For example:

```

Рис. 4.13: mount

```

FSCK(8)                                     System Administration                               FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-lsAVRTMWP] [-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

    4      Filesystem errors left uncorrected

    8      Operational error

    16     Usage or syntax error

    32     Checking canceled by user request

    128

```

Рис. 4.14: fsck

```

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.

    -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

```

Рис. 4.15: mkfs

```

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:

        0
            where n is larger than 0. The process with PID n is signaled.

        0
            All processes in the current process group are signaled.

        -1
            All processes with a PID larger than 1 are signaled.

```

Рис. 4.16: kill

## **5 Выводы**

В ходе данной лабораторной работы я освоил навыки пользования файловой системой Linux, её структурой, именами и содержанием каталогов



# Список литературы

Лабораторная работа 7 ::: {#refs} :::