

Отчет по лабораторной работе №6

Дисциплина: Операционные системы

Морозова Анастасия Владимировна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	26

List of Tables

List of Figures

2.1	Выполнение примеров команд из лабораторной работы	7
2.2	Выполнение примеров команд из лабораторной работы	7
2.3	Выполнение примеров команд из лабораторной работы	8
2.4	Выполнение примеров команд из лабораторной работы	8
2.5	Работа с каталогами и файлами	9
2.6	Работа с каталогами и файлами	10
2.7	Создаю каталоги, присваиваю права	10
2.8	Команда ls -l	11
2.9	Содержимое файла /etc/passwd	11
2.10	Работа с файлами	12
2.11	Действия с выдачей и лишением прав	13
2.12	Команда man	13
2.13	man mount	14
2.14	man fsck	15
2.15	man mkfs	16
2.16	Опция -a	17
2.17	Файловые системы	19

1 Цель работы

Ознакомиться с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрести практические навыки 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. Выполню все примеры, приведённые в первой части описания лабораторной работы
 - Скопирую файл ~/abc1 в файл april и в файл may. Для этого создадим файл abc1 (команда touch abc1), осуществляю копирование (команды cp abc1 april и cp abc1 may).
 - Скопирую файлы april и may в каталог monthly (команды mkdir monthly – для создания каталога monthly и cp april may monthly – для копирования).
 - Скопирую файл monthly/may в файл с именем june (команды cp monthly/may monthly/june и ls monthly (для просмотра содержимого каталога)).
 - Скопирую каталог monthly в каталог monthly.00. Создаю каталог monthly.00 (команда mkdir monthly.00) и осуществляю копирование (команда cp -r monthly monthly.00)(команда cp с опцией -r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами).
 - Скопирую каталог monthly.00 в каталог /tmp (команда cp -r monthly.00 /tmp) (рис. 2.1)

```

avmorofova@avmorofova:~$ cd
avmorofova@avmorofova:~$ touch abc1
avmorofova@avmorofova:~$ cp abc1 april
avmorofova@avmorofova:~$ cp abc1 may
avmorofova@avmorofova:~$ mkdir monthly
avmorofova@avmorofova:~$ cp april may monthly
avmorofova@avmorofova:~$ cp monthly/may monthly/june
avmorofova@avmorofova:~$ ls monthly
april  june  may
avmorofova@avmorofova:~$ mkdir monthly.00
avmorofova@avmorofova:~$ cp -r monthly monthly.00
avmorofova@avmorofova:~$ ls monthly.00
monthly
avmorofova@avmorofova:~$ cp -r monthly.00 /tmp

```

Figure 2.1: Выполнение примеров команд из лабораторной работы

- Изменяю название файла april на july в домашнем каталоге (команда mv april july).
- Перемещаю файл july в каталог monthly.00 (команда mv july monthly.00). Проверяю результат(команда ls monthly.00).
- Переименовываю каталог monthly.00 в monthly.01 (команда mv monthly.00 monthly.01).
- Перемещаю каталог monthly.01 в каталог reports. Создаю каталог reports (команда mkdir reports) и выполняю перемещение (команда mv monthly.01 reports).
- Переименовываю каталог reports/monthly.01 в reports/monthly (команда mv reports/monthly.01 reports/monthly). (рис. 2.2)

```

avmorofova@avmorofova:~$ mv april july
avmorofova@avmorofova:~$ mv july monthly.00
avmorofova@avmorofova:~$ ls monthly.00
july  monthly
avmorofova@avmorofova:~$ mv monthly.00 monthly.01
avmorofova@avmorofova:~$ mkdir reports
avmorofova@avmorofova:~$ mv monthly.01 reports
avmorofova@avmorofova:~$ ls reports
monthly.01
avmorofova@avmorofova:~$ mv reports/monthly.01 reports/monthly
avmorofova@avmorofova:~$ ls reports
monthly

```

Figure 2.2: Выполнение примеров команд из лабораторной работы

- Создаю файл ~/may с правом выполнения для владельца. Команды: touch may - создание файла ls -l may - просмотр сведений о файле chmod u+x may - изменение прав ls -l may
- Лишаю владельца файла ~/may права на выполнение. Команды: chmod u-x may - изменение прав ls -l may - просмотр сведений о файле. (рис. 2.3)
- Создаю каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Команды: mkdir monthly - создание каталога chmod go-r monthly - изменение прав
- Создаю файл ~/abc1 с правом записи для членов группы. Команды: touch abc1 - создание файла chmod g+w abc1 - изменение прав (рис. 2.4)

```
avmorofova@avmorofova:~$ touch may
avmorofova@avmorofova:~$ ls -l may
-rw-rw-r-- 1 avmorofova avmorofova 0 мая 12 14:10 may
avmorofova@avmorofova:~$ chmod u+x may
avmorofova@avmorofova:~$ ls -l may
-rwxrw-r-- 1 avmorofova avmorofova 0 мая 12 14:10 may
avmorofova@avmorofova:~$ chmod u-x may
avmorofova@avmorofova:~$ ls -l may
-rw-rw-r-- 1 avmorofova avmorofova 0 мая 12 14:10 may
```

Figure 2.3: Выполнение примеров команд из лабораторной работы

```
avmorofova@avmorofova:~$ chmod go-r monthly
avmorofova@avmorofova:~$ touch abc1
avmorofova@avmorofova:~$ chmod g+w abc1
avmorofova@avmorofova:~$ ls -l abc1
-rw-rw-r-- 1 avmorofova avmorofova 0 мая 12 14:20 abc1
```

Figure 2.4: Выполнение примеров команд из лабораторной работы

2. Выполняю следующие действия:

- Копирую файл /usr/include/sys/aio.h в домашний каталог (команда cp /usr/include/sys/aio.h ~). Назовоау его equipment (команда mv aio.h equipment).

- В домашнем каталоге создаю директорию ~/ski.plases (команда mkdir ski.plases).
- Перемещаю файл equipment в каталог ~/ski.plases (команда mv equipment ski.plases).
- Переименовываю файл ~/ski.plases/equipment в ~/ski.plases/equiplist (команда mv ski.plases/equipment ski.plases/equiplist) (рис. 2.5)
- Создаю в домашнем каталоге файл abc1 (команда touch abc1) и копирую его в каталог ~/ski.plases (команда cp abc1 ski.plases). Называю его equiplist2 (команда mv ski.plases/abc1 ski.plases/equiplist2).
- Создаю каталог с именем equipment в каталоге ~/ski.plases (команда mkdir ski.plases/equipment).
- Перемещаю файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment (команда mv ski.plases/equiplist ski.plases/equiplist2 ski.plases/equipment).
- Создаю (команда mkdir newdir») и перемещаю каталог ~/newdir в каталог ~/ski.plases (команда mv newdir ski.plases)и называю его plans (команда mv ski.plases/newdir ski.plases/plans) (рис. 2.6)

```
avmorozova@avmorozova:~$ cp /usr/include/sys/io.h ~
cp: cannot stat '/usr/include/sys/io.h': No such file or directory
avmorozova@avmorozova:~$ cp usr/include/sys/io.h ~
avmorozova@avmorozova:~$ ls
abc1                               Downloads  may        Pictures  Templates
academic-laboratory-report-template  io.h      monthly   Public    usr
Desktop                           lab2      Music     reports   Videos
Documents                          lab_OS    newdir     snap
avmorozova@avmorozova:~$ mv io.h equipment
avmorozova@avmorozova:~$ mkdir ski.plases
avmorozova@avmorozova:~$ mv equipment ski.plases
avmorozova@avmorozova:~$ ls
abc1                               lab2      newdir     snap
academic-laboratory-report-template  lab_OS    Pictures   Templates
Desktop                             may       Public     usr
Documents                           monthly   reports    Videos
Downloads                           Music     ski.plases
avmorozova@avmorozova:~$ cd ski.plases
avmorozova@avmorozova:~/ski.plases$ ls
equipment
avmorozova@avmorozova:~/ski.plases$ cd
avmorozova@avmorozova:~$ mv ski.plases/equipment ski.plases/equiplist
```

Figure 2.5: Работа с каталогами и файлами

```

avmorofova@avmorofova:~/ski.places$ cd
avmorofova@avmorofova:~$ touch abc1
avmorofova@avmorofova:~$ mv abc1 ski.places
avmorofova@avmorofova:~$ mv ski.places/abc1 ski.places/equiplist2
avmorofova@avmorofova:~$ cd ski.places
avmorofova@avmorofova:~/ski.places$ ls
equiplist2  equiplist2
avmorofova@avmorofova:~/ski.places$ mkdir equipment
avmorofova@avmorofova:~/ski.places$ cd
avmorofova@avmorofova:~$ mv ski.places/equiplist2 ski.places/equipment
avmorofova@avmorofova:~$ ls equipment
ls: cannot access 'equipment': No such file or directory
avmorofova@avmorofova:~$ ls ski.places/equipment
equiplist2  equiplist2
avmorofova@avmorofova:~$ cd
avmorofova@avmorofova:~$ mkdir newdir
mkdir: cannot create directory 'newdir': File exists
avmorofova@avmorofova:~$ mv newdir ski.places
avmorofova@avmorofova:~$ mv ski.places/newdir ski.places/plans
avmorofova@avmorofova:~$ ls ski.places
equipment  plans

```

Figure 2.6: Работа с каталогами и файлами

3. Определяю опции команды `chmod`, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Создаю необходимые файлы, используя команды: `mkdir australia`, `mkdir play`, `touch my_os`, `touch feathers` (рис. 2.7)

```

avmorofova@avmorofova:~$ mkdir australia
avmorofova@avmorofova:~$ mkdir play
avmorofova@avmorofova:~$ touch my_os
avmorofova@avmorofova:~$ touch feathers
avmorofova@avmorofova:~$ chmod 744 australia
avmorofova@avmorofova:~$ chmod 711 play
avmorofova@avmorofova:~$ chmod 544 my_os
avmorofova@avmorofova:~$ chmod 664 feathers

```

Figure 2.7: Создаю каталоги, присваиваю права

- **drwxr-r- ... australia** (команда `chmod 744 australia`) (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение)
- **drwx-x-x ... play** (команда `chmod 711 play`) (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение)

- **-r-xr-r- ... my_o** (команда `chmod 544 my_o`)(это файл, владелец имеет право на чтение и выполнение, группа владельца и остальные –только чтение).
- **-rw-rw-r- ... feathers** (команды `chmod 664 feathers`)(это файл, владелец и группа владельца имеют право на чтение и запись, остальные –только чтение)

Командой `ls -l` проверяю правильность выполненных действий. (рис. 2.8)

```
avmorozova@avmorozova:~$ ls -l
total 72
drwxrwxr-x 5 avmorozova avmorozova 4096 апр 30 14:15 academic-laboratory-report
-template
drwxr--r-- 2 avmorozova avmorozova 4096 мая 12 17:01 australia
drwxr-xr-x 2 avmorozova avmorozova 4096 апр 28 13:48 Desktop
drwxr-xr-x 2 avmorozova avmorozova 4096 апр 28 13:48 Documents
drwxr-xr-x 4 avmorozova avmorozova 4096 мая 11 13:37 Downloads
-rw-rw-r-- 1 avmorozova avmorozova 0 мая 12 17:02 feathers
drwxrwxr-x 3 avmorozova avmorozova 4096 апр 29 12:55 lab2
drwxrwxr-x 5 avmorozova avmorozova 4096 мая 11 13:15 lab_05
-rw-rw-r-- 1 avmorozova avmorozova 0 мая 12 14:10 may
drwx-wx--x 2 avmorozova avmorozova 4096 мая 12 13:57 monthly
drwxr-xr-x 2 avmorozova avmorozova 4096 апр 28 13:48 Music
-r-xr--r-- 1 avmorozova avmorozova 0 мая 12 17:02 my_os
drwxr-xr-x 2 avmorozova avmorozova 4096 мая 1 02:20 Pictures
drwx--x--x 2 avmorozova avmorozova 4096 мая 12 17:02 play
```

Figure 2.8: Команда `ls -l`

4. Выполняю следующие действия:

- Просматриваю содержимое файла `/etc/passwd` (команда `cat /etc/passwd`) (рис. 2.9)

```
avmorozova@avmorozova:/etc$ cd
avmorozova@avmorozova:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

Figure 2.9: Содержимое файла `/etc/passwd`

- Копирую файл ~/feathers в файл ~/file.old (команда `cp feathers file.old`).
- Перемещаю файл ~/file.old в каталог ~/play (команда `mv file.ord play`).
- Скопирую каталог ~/play в каталог ~/fun (команда `cp -r play fun`).
- Перемещаю каталог ~/fun в каталог ~/play (команда `mv fun play`) и называю games (команда `mv play/fun play/games`).
- Лишаю владельца файла ~/feathers права на чтение (команда `chmod u-r feathers`) (рис. 2.10)

```
avmorofova@avmorofova:~$ cp feathers file.old
avmorofova@avmorofova:~$ mv file.old play
avmorofova@avmorofova:~$ cp -r play fun
avmorofova@avmorofova:~$ mv fun play
avmorofova@avmorofova:~$ mv play/fun play/games
avmorofova@avmorofova:~$ chmod u-r feathers
```

Figure 2.10: Работа с файлами

- При попытке просмотреть файл ~/feathers командой `cat`, получаю отказ в доступе, т.к. в предыдущем пункте лишила владельца права на чтение данного файла.
- При попытке скопировать файл ~/feathers получаю отказ в доступе, по той же причине
- Даю владельцу файла ~/feathers право на чтение (команда `chmod u+r feathers`).
- Лишаю владельца каталога ~/play права на выполнение (команда `chmod u-x play`).
- Перехожу в каталог ~/play (команда `cd play`). Получаю отказ в доступе, т.к. в предыдущем пункте лишила владельца права на выполнение данного каталога.
- Даю владельцу каталога ~/play право на выполнение (команда `chmod u+x play`). (рис. 2.11)

```

avmorozova@avmorozova:~$ cat feathers
cat: feathers: Permission denied
avmorozova@avmorozova:~$ ls
academic-laboratory-report-template  feathers  Music      reports  Videos
australia                             lab2     my_os      ski.places
Desktop                               lab_OS   Pictures   snap
Documents                             may      play       Templates
Downloads                             monthly  Public     usr
avmorozova@avmorozova:~$ cp feathers montly
cp: cannot open 'feathers' for reading: Permission denied
avmorozova@avmorozova:~$ chmod u+r feathers
avmorozova@avmorozova:~$ chmod u-x play
avmorozova@avmorozova:~$ cd play
bash: cd: play: Permission denied
avmorozova@avmorozova:~$ chmod u+x play

```

Figure 2.11: Действия с выдачей и лишением прав

5. Используя команды `man mount`, `man fsck`, `man mkfs`, `man kill`, получаю информацию о соответствующих командах (рис. 2.12)

```

avmorozova@avmorozova:~$ man mount
avmorozova@avmorozova:~$ man fsck
avmorozova@avmorozova:~$ man mkfs
avmorozova@avmorozova:~$ man kill

```

Figure 2.12: Команда `man`

- Команда **mount**: предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого дерева обозначается как `/`. Физически файлы могут располагаться на различных устройствах. Команда `mount` служит для подключения файловых систем разных устройств к этому большому дереву. (рис. 2.13)

```
MOUNT(8)                                System Administration                                MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-l|-h|-V]

    mount -a [-fFnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device|dir

    mount [-fnrsvw] [-t fstype] [-o options] device dir

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 another 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
```

Figure 2.13: man mount

- Команда fsck: это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Он использует программы, специфичные для типа файловой системы, которую он проверяет. У команды fsck следующий синтаксис: `fsck [параметр] -[параметры ФС] [...]`. Например, если нужно восстановить («починить») файловую систему на некотором устройстве `/dev/sdb2`, следует воспользоваться командой: «`sudo fsck -y /dev/sdb2`» (рис. 2.14)


```
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. filesystems 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 code 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
```

Figure 2.14: man fsck

- Команда mkfs: создаёт новую файловую систему Linux. Имеет синтаксис: `mkfs [-V] [-tfstype] [fs-options] filesystem [blocks]`. 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). Аргументом blocks указывается количество блоков, которые выделяются для использования этой файловой системой. По окончании работы mkfs возвращает 0 - в случае успеха, а 1 - при неудачной операции. Например, команда «`mkfs -t ext2 /dev/hdb1`» создаёт файловую систему типа ext2 в разделе /dev/hdb1 (второй жёсткий диск) (рис. 2.15)

```
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 code 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.
```

Figure 2.15: man mkfs

- Команда kill: посылает сигнал процессу или выводит список допустимых сигналов. Имеет следующий синтаксис: kill[опции]PID, гдеPID–это PID (числовой идентификатор) процесса или несколько PID процессов, если требуется послать сигнал сразу нескольким процессам. Например, команда «kill -KILL 3121» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс(рис. 2.16)


```
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    I
    The default signal for kill is TERM. Use -l or -L to list available
    signals. Particularly useful signals include HUP, INT, KILL, STOP,
    CONT, and 0. Alternate signals may be specified in three ways: -9,
    -SIGKILL or -KILL. Negative PID values may be used to choose whole
    process groups; see the PGID column in ps command output. A PID of -1
    is special; it indicates all processes except the kill process itself
    and init.

OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<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.
```

Figure 2.16: Опция -a

6. Контрольные вопросы:

- Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df-Th». Из рисунка видно, что на моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. (рис. 2.17)
- **devtmpfs** позволяет ядру создать экземпляр tmpfs с именем devtmpfs при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором / минором будет предоставлять узел устройства в devtmpfs. devtmpfs монтируется на /dev и содержит специальные файлы устройств для всех устройств.
- **tmpfs** – временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначена для быстрого и ненадёжного хранения временных данных. Хорошо подходит для /tmp и массовой сборки пакетов/образов. Предполагает наличие достаточного объёма виртуальной

памяти. Файловая система **tmpfs** предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска.

- **ext4** – имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «**extent file system**», который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (**delayed allocation – delalloc**), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных. Характеристики: максимальный размер файла: 16 ТБ; максимальный размер раздела: 16 ТБ; максимальный размер имени файла: 255 символов. Рекомендации по использованию: наилучший выбор для SSD; наилучшая производительность по сравнению с предыдущими Ext-системами; она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3
- **iso9660** – стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Также известен как CDFS (**Compact Disc File System**). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

```
avmorofova@avmorofova:~/lab_OS/2020-2021/may/lab06$ df -Th
Filesystem      Type      Size      Used Avail Use% Mounted on
udev            devtmpfs  1.9G       0    1.9G   0% /dev
tmpfs           tmpfs     394M       1,4M 393M   1% /run
/dev/sda5       ext4      20G       11G   7,7G  58% /
tmpfs           tmpfs     2,0G       0    2,0G   0% /dev/shm
tmpfs           tmpfs     5,0M       4,0K 5,0M   1% /run/lock
tmpfs           tmpfs     2,0G       0    2,0G   0% /sys/fs/cgroup
/dev/loop0      squashfs  9,2M       9,2M   0 100% /snap/canonical-livepatch/98
/dev/loop2      squashfs  99M        99M   0 100% /snap/core/11081
/dev/loop3      squashfs  56M        56M   0 100% /snap/core18/1997
/dev/loop1      squashfs 100M       100M   0 100% /snap/core/10958
/dev/loop4      squashfs 219M       219M   0 100% /snap/gnome-3-34-1804/66
/dev/loop5      squashfs  66M        66M   0 100% /snap/gtk-common-themes/1515
/dev/loop6      squashfs  56M        56M   0 100% /snap/core18/1988
/dev/loop7      squashfs  65M        65M   0 100% /snap/gtk-common-themes/1514
/dev/loop8      squashfs  33M        33M   0 100% /snap/snapd/11588
/dev/loop10     squashfs  33M        33M   0 100% /snap/snapd/11841
/dev/loop9      squashfs  52M        52M   0 100% /snap/snap-store/518
/dev/sda1       vfat      511M       4,0K 511M   1% /boot/efi
shared         vboxsf    120G       76G   45G   64% /media/sf_shared
для_лаб        vboxsf    120G       76G   45G   64% /media/sf_
tmpfs           tmpfs     394M       68K 394M   1% /run/user/1000
/dev/sr0        iso9660   74M        74M   0 100% /media/avmorofova/VBox_GAs_6.0.2
4
```

Figure 2.17: Файловые системы

- Файловая система Linux/UNIX физически представляет собой пространство раздела диска, разбитое на блоки фиксированного размера, кратные размеру сектора – 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел – / (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам
- / – корень – это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога. Только пользователь root имеет право читать и изменять файлы в этом каталоге.
- /BIN – бинарные файлы пользователя. Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления.
- /SBIN – системные исполняемые файлы. Так же как и /bin, содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки,

когда не примонтирован каталог `/usr`. Но здесь находятся программы, которые можно выполнять только с правами суперпользователя.

- `/ETC` – конфигурационные файлы В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, в системе инициализации `Init Scripts`, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ.
- `/DEV` – файлы устройств В Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры – это просто файлы в каталоге `/dev/`. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов
- `/PROC` – информация о процессах По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его `Pid`, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов.
- `/VAR` – переменные файлы. Название каталога `/var` говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кешы, базы данных и так далее.
- `/TMP` – временные файлы. В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию.
- `/USR` – программы пользователя Это самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники

программ, различные ресурсы приложений, картинки, музыку и документацию.

- /HOME – домашняя папка В этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т.д.
- /BOOT – файлы загрузчика Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящиеся в каталоге /boot/grub.
- /LIB – системные библиотеки Содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах /bin и /sbin./OPT– дополнительные программы В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями.
- /MNT – монтирование. В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы.
- /MEDIA – съемные носители. В этот каталог система монтирует все подключаемые внешние накопители –USB флешки, оптические диски и другие носители информации.
- /SRV – сервер В этом каталоге содержатся файлы серверов и сервисов.
- /RUN - процессы. Каталог, содержащий PID файлы процессов, похожий на /var/run, но в отличие от него, он размещен в TMPFS, а поэтому после перезагрузки все файлы теряются
- Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount.
- Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного

выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду `fsck`.

- Файловую систему можно создать, используя команду `mkfs`.
- Для просмотра текстовых файлов существуют следующие команды: `cat`
Задача команды `cat` очень проста – она читает данные из файла или стандартного ввода и выводит их на экран. Синтаксис утилиты: `cat [опции] файл1 файл2 ...` Основные опции:
 - `-b` – нумеровать только непустые строки
 - `-E` – показывать символ `$` в конце каждой строки
 - `-n` – нумеровать все строки
 - `-s` – удалять пустые повторяющиеся строки
 - `-T` – отображать табуляции в виде `^I`
 - `-h` – отобразить справку
 - `-v` – версия утилиты

`nl`: команда `nl` действует аналогично команде `cat`, но выводит еще и номера строк в столбце слева. `less`: существенно более развитая команда для пролистывания текста. При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста. Синтаксис аналогичный синтаксису команды `cat`. Некоторые опции: `-g` – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения) `-N` – показывать номера строк `head`: команда `head` выводит начальные строки (по умолчанию – 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита. Синтаксис аналогичный синтаксису команды `cat`. Основные опции: `-c` (`-bytes`) – позволяет

задавать количество текста не в строках, а в байтах -n (-lines) – показывает заданное количество строк вместо 10, которые выводятся по умолчанию -q (-quiet, -silent) – выводит только текст, не добавляя к нему название файла -v (-verbose) – перед текстом выводит название файла -z (-zero-terminated) – символы перехода на новую строку заменяет символами завершения строк tail: эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме. Синтаксис аналогичный синтаксису команды cat. Основные опции: -c – выводить указанное количество байт с конца файла -f – обновлять информацию по мере появления новых строк в файле -n – выводить указанное количество строк из конца файла -pid – используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс -q – не выводить имена файлов -retry – повторять попытки открыть файл, если он недоступен -v – выводить подробную информацию о файле

- Утилита `cp` позволяет полностью копировать файлы и директории. Синтаксис: `cp [опции] файл-источник файл-приемник`. После выполнения команды файл-источник будет полностью перенесен в файл-приемник. Если в конце указан слэш, файл будет записан в заданную директорию с оригинальным именем. Основные опции: -attributes-only – не копировать содержимое файла, а только флаги доступа и владельца -f, -force – перезаписывать существующие файлы -i, -interactive – спрашивать, нужно ли перезаписывать существующие файлы -L – копировать не символические ссылки, а то, на что они указывают -n – не перезаписывать существующие файлы -P – не следовать символическим ссылкам -r – копировать папку Linux рекурсивно -s – не выполнять копирование файлов в Linux, а создавать символические ссылки -u – скопировать файл, только если он был изменён -x – не выходить за пределы этой файловой системы -p – сохранять владельца, временные метки и флаги доступа при копировании -t – считать файл-приемник директорией и копировать файл-источник в эту директорию

- Команда `mv` используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий. Синтаксис: `mv [-опции] старый_файл новый_файл`
Основные опции: `-help` – выводит на экран официальную документацию об утилите `-version` – отображает версию `mv` `-b` – создает копию файлов, которые были перемещены или перезаписаны `-f` – при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла `-i` – наоборот, будет спрашивать разрешение у владельца `-n` – отключает перезапись уже существующих объектов `-strip-trailing-slashes` – удаляет завершающий символ `/` у файла при его наличии `-t [директория]` – перемещает все файлы в указанную директорию `-u` – осуществляет перемещение только в том случае, если исходный файл новее объекта назначения `-v` – отображает сведения о каждом элементе во время обработки команды

Команда `rename` также предназначена, чтобы переименовать файл. Синтаксис: `rename [опции] старое_имя новое_имя файлы` Основные опции: `-v` – вывести список обработанных файлов `-n` – тестовый режим, на самом деле никакие действия выполнены не будут `-f` – принудительно перезаписывать существующие файлы

- Права доступа – совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её установленных носителям, правовыми процессам и документами другим или ресурсам) собственником, владельцем информации. Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Синтаксис команды: `chmod режим имя_файла` Режим имеет следующие компоненты структуры и способ записи:
 - `=` установить право
 - `-`лишить права
 - `+`дать право

- r чтение
- w запись
- x выполнение
- u (user) владелец файла
- g (group) группа, к которой принадлежит владелец файла
- o (others) все остальные

3 Выводы

В ходе выполнения лабораторной работы я ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрела практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.