

Отчёт по лабораторной работе №7

Дисциплина: Операционные Системы

Азарцова Вероника Валерьевна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Краткое описание команд	7
3.2	Анализ файловой системы	7
3.3	Права доступа	8
3.4	Изменение прав доступа	8
4	Выполнение лабораторной работы	10
5	Контрольные вопросы	18
6	Выводы	22
	Список литературы	23

Список иллюстраций

4.1	Выполнение примеров	10
4.2	Выполнение действий	11
4.3	Работа с <code>chmod</code>	12
4.4	Содержимое файла <code>/etc/passwd</code>	12
4.5	Дополнительные упражнения	13
4.6	<code>man mount</code>	14
4.7	<code>man fsck</code>	15
4.8	<code>man mkfs</code>	16
4.9	<code>kill</code>	17

Список таблиц

1 Цель работы

Цель данной 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. Выполнить все примеры в первой части лабораторной работы
2. Выполнить действия по копированию и перемещению файлов
3. Определить опции команды `chmod` для присваивания прав доступа
4. Прodelать дополнительные упражнения
5. Ответить на контрольные вопросы

3 Теоретическое введение

3.1 Краткое описание команд

Краткое описание команд, нужных для выполнения лабораторной работы:

- man: Просмотр руководства по основным командам Linux
- cd: Перемещение по файловой системе
- ls: Просмотр содержимого каталога
- mkdir: Создание каталогов
- rm: Удаление файлов (или каталогов, с параметрами)
- cp: Копирование файлов или каталогов
- chmod: Установка прав доступа к файлу или каталогу

3.2 Анализ файловой системы

Файловая система в Linux состоит из файлов и каталогов. Каждому физическому носителю соответствует своя файловая система.

Существует несколько типов файловых систем. Перечислим наиболее часто встречающиеся типы:

- ext2fs (second extended filesystem);
- ext3fs (third extended file system);
- ext4 (fourth extended file system);
- ReiserFS;

- xfs;
- fat (file allocation table);
- ntfs (new technology file system).

Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой `mount` без параметров.

3.3 Права доступа

Каждый файл или каталог имеет права доступа. В сведениях о файле или каталоге указываются: - тип файла (символ (-) обозначает файл, а символ (d) — каталог); - права для владельца файла (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует); - права для членов группы (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует); - права для всех остальных (r — разрешено чтение, w — разрешена запись, x — разрешено выполнение, - — право доступа отсутствует).

3.4 Изменение прав доступа

Права доступа к файлу или каталогу можно изменить, воспользовавшись командой `chmod`. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.

Режим (в формате команды) 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

В работе с правами доступа можно использовать их цифровую запись (восьмеричное значение) вместо символьной.

4 Выполнение лабораторной работы

1. Выполняю все примеры, приведённые в первой части описания лабораторной работы (рис. 4.1).

```
egrep: warning: egrep is obsolescent; using grep -E
egrep: warning: egrep is obsolescent; using grep -E
vvazarcova@vvazarcova:~$ touch abc1
vvazarcova@vvazarcova:~$ cp abc1 april
vvazarcova@vvazarcova:~$ cp abc1 may
vvazarcova@vvazarcova:~$ mkdir monthly
vvazarcova@vvazarcova:~$ cd april may monthly
bash: cd: слишком много аргументов
vvazarcova@vvazarcova:~$ cp april may monthly
vvazarcova@vvazarcova:~$ ls
abc1      Downloads  monthly    work       Изображения  Шаблоны
april     github.io  newdir     Видео      Музыка
bin       LICENSE    Pictures   Документы  Общедоступные
Documents may        repos      Загрузки   'Рабочий стол'
vvazarcova@vvazarcova:~$ cd monthly
vvazarcova@vvazarcova:~/monthly$ ls
april may
vvazarcova@vvazarcova:~/monthly$ cd ~
vvazarcova@vvazarcova:~$ cp monthly/may monthly/june
vvazarcova@vvazarcova:~$ ls monthly
april june may
vvazarcova@vvazarcova:~$ mkdir monthly.00
vvazarcova@vvazarcova:~$ cp -r monthly monthly.00
vvazarcova@vvazarcova:~$ cp -r monthly.00 /tmp
vvazarcova@vvazarcova:~$ ^[[200~mv april july
bash: mv: команда не найдена...
vvazarcova@vvazarcova:~$ mv april july
vvazarcova@vvazarcova:~$ mv july monthly.00
vvazarcova@vvazarcova:~$ ls monthly.00
july  monthly
vvazarcova@vvazarcova:~$ mv monthly.00 monthly.01
vvazarcova@vvazarcova:~$ mkdir reports
vvazarcova@vvazarcova:~$ mv monthly.01 reports
vvazarcova@vvazarcova:~$ mv reports/monthly.01 reports/monthly
vvazarcova@vvazarcova:~$ touch may
vvazarcova@vvazarcova:~$ ls -l may
-rw-r--r--. 1 vvazarcova vvazarcova 0 map 27 23:19 may
vvazarcova@vvazarcova:~$ 4 chmod u+x may
bash: 4: команда не найдена...
vvazarcova@vvazarcova:~$ chmod u+x may
vvazarcova@vvazarcova:~$ ls -l may
-rwxr--r--. 1 vvazarcova vvazarcova 0 map 27 23:19 may
vvazarcova@vvazarcova:~$ chmod u-x may
vvazarcova@vvazarcova:~$ ls -l may
-rw-r--r--. 1 vvazarcova vvazarcova 0 map 27 23:19 may
vvazarcova@vvazarcova:~$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
vvazarcova@vvazarcova:~$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
vvazarcova@vvazarcova:~$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
vvazarcova@vvazarcova:~$ touch abc1
vvazarcova@vvazarcova:~$ chmod g+w abc1
```

Рис. 4.1: Выполнение примеров

2. Выполняю действия, приведенные в последовательности выполнения лабораторной работы:

- 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` (рис. 4.2).

```
vvazarcova@vvazarcova:~$ cd /usr/include/sys
vvazarcova@vvazarcova:~$ cp io.h ~/
vvazarcova@vvazarcova:~$ cd ~
vvazarcova@vvazarcova:~$ mv io.h equipment
vvazarcova@vvazarcova:~$ mkdir ~/ski.places
vvazarcova@vvazarcova:~$ ls
abcl      Downloads  LICENSE   newdir    reps      Видео      Изображения  'Рабочий стол'
bin       equipment  may       Pictures  ski.places  Документы  Музыка       Шаблоны
Documents github.io  monthly   reports   work       Загрузки    Общедоступные
vvazarcova@vvazarcova:~$ mv equipment ~/ski.places
vvazarcova@vvazarcova:~$ ls ski.places
equipment
vvazarcova@vvazarcova:~$ mv ~/ski.places/equipment ~/ski.places/equiplist
vvazarcova@vvazarcova:~$ ls ski.places
equiplist
vvazarcova@vvazarcova:~$ touch abcl
vvazarcova@vvazarcova:~$ cp abcl ~/ski.places
vvazarcova@vvazarcova:~$ mv ~/ski.places/abcl ~/ski.places/equiplist2
vvazarcova@vvazarcova:~$ mkdir ~/ski.places/equipment
vvazarcova@vvazarcova:~$ mv ~/ski.places/equiplist ~/ski.places/equipment
vvazarcova@vvazarcova:~$ mv ~/ski.places/equiplist2 ~/ski.places/equipment
vvazarcova@vvazarcova:~$ mkdir ~/newdir
mkdir: невозможно создать каталог «/home/vvazarcova/newdir»: Файл существует
vvazarcova@vvazarcova:~$ mkdir ~/newdir
vvazarcova@vvazarcova:~$ mv ~/newdir ~/ski.places
mv: не удалось выполнить stat для '/home/vvazarcova/newdir': Нет такого файла или каталога
vvazarcova@vvazarcova:~$ mv ~/newdir ~/ski.places
vvazarcova@vvazarcova:~$ mv ~/ski.places/newdir ~/ski.places/plans
vvazarcova@vvazarcova:~$
```

Рис. 4.2: Выполнение действий

3. Определяю опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в

начале таких прав нет: australia, play, my_os, feathers.

Сначала создаю нужные файлы (рис. 4.3).

```
vazarcova@vazarcova:~$ mv /usr/games/newdir /usr/games/plans
vazarcova@vazarcova:~$ mkdir task
vazarcova@vazarcova:~$ cd task
vazarcova@vazarcova:~/task$ touch australia play my_os feathers
vazarcova@vazarcova:~/task$ chmod g+r,o+r australia
vazarcova@vazarcova:~/task$ chmod g-w,g-r,o-r play
vazarcova@vazarcova:~/task$ chmod u-w,u+x,g-w my_os
vazarcova@vazarcova:~/task$ chmod u-x,g-x,o-w,o-x feathers
vazarcova@vazarcova:~/task$ cat /etc/passwd
```

Рис. 4.3: Работа с chmod

4. Проделываю приведённые ниже упражнения:

- Просмотрите содержимое файла /etc/passwd (рис. 4.4)

```
vazarcova@vazarcova:~/task$ 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-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/usr/sbin/nologin
systemd-oom:x:998:998:systemd Userspace OOM Killer:/usr/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/usr/sbin/nologin
qemu:x:107:107:qemu user:/usr/sbin/nologin
polkitd:x:114:114:User for polkitd:/usr/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/usr/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/usr/sbin/nologin
```

Рис. 4.4: Содержимое файла /etc/passwd

- 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 (рис. 4.5).

```

vvazarcova@vvazarcova:~/task$ cp ~/task/feathers ~/file.old
vvazarcova@vvazarcova:~/task$ mc ~/file.old ~/play

vvazarcova@vvazarcova:~/task$ mv ~/file.old ~/task/play
vvazarcova@vvazarcova:~/task$ mv ~/play ~/file.old
vvazarcova@vvazarcova:~/task$ mc ~/file.old ~/task/play

vvazarcova@vvazarcova:~/task$ mv ~/file.old ~/task/play
vvazarcova@vvazarcova:~/task$ cp ~/task/play ~/task/fun
vvazarcova@vvazarcova:~/task$ mv ~/task/fun ~/task/play
vvazarcova@vvazarcova:~/task$ mv ~/task/play ~/task/games
vvazarcova@vvazarcova:~/task$ chmod u-r ~/task/feathers
vvazarcova@vvazarcova:~/task$ cat ~/task/feathers
cat: /home/vvazarcova/task/feathers: Отказано в доступе
vvazarcova@vvazarcova:~/task$ chmod u-x ~/task/play
chmod: невозможно получить доступ к '/home/vvazarcova/task/play': Нет такого каталога
vvazarcova@vvazarcova:~/task$ mkdir ~/task/play
vvazarcova@vvazarcova:~/task$ chmod u-x ~/task/play
vvazarcova@vvazarcova:~/task$ cd ~/task/play
bash: cd: /home/vvazarcova/task/play: Отказано в доступе
vvazarcova@vvazarcova:~/task$ chmod u+x ~/task/play
vvazarcova@vvazarcova:~/task$ cd ~/task/play
vvazarcova@vvazarcova:~/task/play$ cd ~

```

Рис. 4.5: Дополнительные упражнения

5. Прочитываю man по некоторым командам:

- mount (рис. 4.6).

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

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

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

    mount [-fnrsw] [-o options] device mountpoint

    mount [-fnrsw] [-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 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).

Listing the mounts
    The listing mode is maintained for backward compatibility only.

    For more robust and customizable output use findmnt(8), especially in your scripts. Note that control characters in the mountpoint
```

Рис. 4.6: man mount

- fsck (рис. 4.7).

```
F5CK(8)                                     System Administration                                     F5CK(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 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
        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.

    In actuality, fsck is simply a front-end for the various filesystem checkers (fsck.fstype) available under Linux. The filesystem-specific checker is searched for in the PATH environment variable. If the PATH is undefined then fallback to /sbin.

    Please see the filesystem-specific checker manual pages for further details.

OPTIONS
    -l
```

Рис. 4.7: man fsck

- mkfs (рис. 4.8).

```

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
    All generic options must precede and not be combined with filesystem-specific options. Some filesystem-specific programs do not automatically detect the device size and require the size parameter to be specified.

AUTHORS
    David Engel <david@ods.com>, Fred N. van Kempen <waltje@u.walt.nl.mugnet.org>, Ron Sommeling <sommel@sci.kun.nl>.

    The manual page was shamelessly adapted from Remy Card's version for the ext2 filesystem.

SEE ALSO
    fs(5), badblocks(8), fsck(8), mkdosfs(8), mke2fs(8), mkfs.bfs(8), mkfs.ext2(8), mkfs.ext3(8), mkfs.ext4(8), mkfs.minix(8),

```

Рис. 4.8: man mkfs

- kill (рис. 4.9).

KILL(1)	User Commands	KILL(1)
NAME		
kill - terminate a process		
SYNOPSIS		
kill [-signal -s <u>signal</u> -p] [-q <u>value</u>] [-a] [--timeout <u>milliseconds</u> <u>signal</u>] [--] <u>pid</u> / <u>name</u> ...		
kill -l [<u>number</u>] -L		
DESCRIPTION		
The command kill sends the specified <u>signal</u> 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 <u>signal</u> 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.		
<u>pid</u>		
Each <u>pid</u> can be expressed in one of the following ways:		
<u>n</u>		
where <u>n</u> is larger than 0. The process with PID <u>n</u> is signaled.		
0		
All processes in the current process group are signaled.		
-1		
All processes with a PID larger than 1 are signaled.		
- <u>n</u>		
where <u>n</u> is larger than 1. All processes in process group <u>n</u> are signaled. When an argument of the form '-n' is given, and it is meant to denote a process group, either a signal must be specified first, or the argument must be preceded by a '--' option, otherwise it will be taken as the signal to send.		
<u>name</u>		
All processes invoked using this <u>name</u> will be signaled.		
OPTIONS		
-s, --signal <u>signal</u>		
The signal to send. It may be given as a name or a number.		
-l, --list [<u>number</u>]		
Print a list of signal names, or convert the given signal number to a name. The signals can be found in		

Рис. 4.9: kill

5 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 экзабайта.

NTFS — это файловая система по умолчанию, используемая операционными системами на базе Windows NT, начиная с 1993 года с Windows NT 3.1 и вплоть до Windows 11 включительно. Она предлагает расширенные функции, такие как права доступа к файлам, шифрование, сжатие и ведение журнала.

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 `inode` (принадлежит нескольким файлам).

- Блок помечен как свободный, но в то же время занят (на него ссылается inode).
- 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.

6 Выводы

Подводя итоги проведенной лабораторной работе, я 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 и ознакомилась с файловой системой Linux, её структурой, именами и содержанием каталогов.

Список литературы

1. GDB: The GNU Project Debugger. — URL: <https://www.gnu.org/software/gdb/>.
2. GNU Bash Manual. — 2016. — URL: <https://www.gnu.org/software/bash/manual/>.
3. Midnight Commander Development Center. — 2021. — URL: <https://midnight-commander.org/>.
4. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. — O'Reilly Media, 2005. — 354 с. — (In a Nutshell). — ISBN 0596009658. — URL: <http://www.amazon.com/Learningbash-Shell-Programming-Nutshell/dp/0596009658>.
6. Robbins A. Bash Pocket Reference. — O'Reilly Media, 2016. — 156 с. — ISBN 978-1491941591.
7. The NASM documentation. — 2021. — URL: <https://www.nasm.us/docs.php>.
8. Zarrelli G. Mastering Bash. — Packt Publishing, 2017. — 502 с. — ISBN 9781784396879.
9. Колдаев В. Д., Лупин С. А. Архитектура ЭВМ. — М. : Форум, 2018.