

Лабораторная работа №7

Дисциплина - Операционные Системы

Азарцова В. В.

28 марта 2025

Российский университет дружбы народов, Москва, Россия

Преподаватель Кулябов Д. С.

Информация

- Азарцова Вероника Валерьевна
- НКАбд-01-24, студ. билет №1132246751
- Российский университет дружбы народов
- 1132246751@pfur.ru
- <https://github.com/vvazarcova>

Целью данной лабораторной работы было 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.

Задачи лабораторной работы:

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

Теоретическое введение

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

Файловая система в Linux состоит из фалов 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.

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).

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
 - r чтение
 - w запись
 - x выполнение
- u (user) владелец файла
- g (group) группа
- o (other) все остальные
- - лишить права
- + дать право

Выполнение лабораторной работы

Выполнение лабораторной работы

1. Выполняю все примеры, приведённые в первой части описания лабораторной работы

```
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  nmdir     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 мар 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 мар 27 23:19 may
vvazarcova@vvazarcova:~$ chmod u-x may
vvazarcova@vvazarcova:~$ ls -l may
-rw-r--r-- 1 vvazarcova vvazarcova 0 мар 27 23:19 may
vvazarcova@vvazarcova:~$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
vvazarcova@vvazarcova:~$ chmod g-r, o-r monthly
```

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`

Выполнение лабораторной работы

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

Рис. 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

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

Рис. 3: Работа с `chmod`

Выполнение лабораторной работы

4. Проделываю приведённые ниже упражнения:

- Просмотрите содержимое файла /etc/passwd

```
vvazarcova@vvazarcova:~/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:/:/sbin/nologin
polkitd:x:114:114:User for polkitd:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
```


Выполнение лабораторной работы

- 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

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

vvazarcova@vvazarcova:~/task$ mv ~/file.old ~/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 файла или 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 ~
```

5. Прочитываю man по некоторым командам:

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

NAME

mount - mount a filesystem

SYNOPSIS

```
mount [-h|-V]
```

```
mount [-L] [-t fstype]
```

```
mount -a [-ffnrsvw] [-t fstype] [-O optlist]
```

```
mount [-fnrsvw] [-o options] device|mountpoint
```

```
mount [-fnrsvw] [-t fstype] [-o options] device mountpoint
```

```
mount --bind|--rbind|--move olddir newdir
```

```
mount --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable] mountpoint
```

DESCRIPTION

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

The standard form of the `mount` command is:

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

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

Выполнение лабораторной работы

- fsck

```
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. 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, fsck_mke2fs) available under Linux. The
```

- mkfs

```

MKFS(8)                                     System Administration                                     MKFS(8)

NAME
    mkfs - build a Linux filesystem

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

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

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

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

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

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

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

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

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

    -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

```

Выполнение лабораторной работы

- kill

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

NAME
    kill - terminate a process

SYNOPSIS
    kill [-signal|-s signal|-p] [-q value] [-a] [--timeout milliseconds signal] [--] pid|name...

    kill -l [number] | -L

DESCRIPTION
    The command kill sends the specified signal to the specified processes or process groups.

    If no signal is specified, the TERM signal is sent. The default action for this signal is to terminate the process. This signal should be used in preference to the KILL signal (number 9), since a process may install a handler for the TERM signal in order to perform clean-up steps before terminating in an orderly fashion. If a process does not terminate after a TERM signal has been sent, then the KILL signal may be used; be aware that the latter signal cannot be caught, and so does not give the target process the opportunity to perform any clean-up before terminating.

    Most modern shells have a builtin kill command, with a usage rather similar to that of the command described here. The --all, --pid, and --queue options, and the possibility to specify processes by command name, are local extensions.

    If signal is 0, then no actual signal is sent, but error checking is still performed.

ARGUMENTS
    The list of processes to be signaled can be a mixture of names and PIDs.

pid
    Each pid can be expressed in one of the following ways:

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

    0
        All processes in the current process group are signaled.

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

    -n
        where n is larger than 1. All processes in process group n 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.

name
    All processes invoked using this name will be signaled.
```

Выводы

Подводя итоги проведенной лабораторной работе, мне удалось:

- Ознакомиться с файловой системой Linux, её структурой, именами и содержанием каталогов.
- Приобрести практические навыки по:
 1. Применению команд для работы с файлами и каталогами
 2. Управлению процессами (и работами)
 3. Проверке использования диска и обслуживанию файловой системы

Если вам понравилось - посмотрите остальные мои презентации!