

Лабораторная работа №5

Анализ файловой системы Linux. Команды для работы с файлами и каталогами.

Карпова Есения Алексеевна

Содержание

1	Цель работы	1
2	Задание.....	1
3	Теоретическое введение	2
3.1	Основные команды:	2
3.2	Права доступа.....	2
3.3	Изменение прав доступа	2
3.4	Анализ файловой системы.....	2
4	Выполнение лабораторной работы	2
5	Выводы.....	10

1 Цель 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. Выполнение примеров
2. Перемещение и переименования файлов и каталогов
3. Изменение прав доступа
4. Копирование файлов и каталогов
5. Описание команд с помощью мануала

3 Теоретическое введение

3.1 Основные команды:

- Для создания текстового файла можно использовать команду touch. Формат команды: touch имя-файла
- Для просмотра файлов небольшого размера можно использовать команду cat. Формат команды: cat имя-файла
- Для просмотра файлов постранично удобнее использовать команду less. Формат команды: less имя-файла
- Команда cp используется для копирования файлов и каталогов. Формат команды: cp [-опции] исходный_файл целевой_файл
- Команды mv и mvdir предназначены для перемещения и переименования файлов и каталогов. Формат команды mv: mv [-опции] старый_файл новый_файл

3.2 Права доступа

Каждый файл или каталог имеет права доступа (табл. 5.1). 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).

3.3 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. Формат команды: chmod режим имя_файла

3.4 Анализ файловой системы

С помощью команды fsck можно проверить (а в ряде случаев восстановить) целостность файловой системы: Формат команды: fsck имя_устройства

4 Выполнение лабораторной работы

1. Выполнение примеров

Проверим работу примеров команд из первой части лабораторной работы. Пример копирования файлов и каталогов(рис. 1).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch abc1
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp abc1 april
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp abc1 may
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir monthly
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp april may monthly
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp monthly/may monthly/june
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ ls monthly
april  june  may
```

Рис. 1: Пример копирования файлов и каталогов

Пример копирования файлов с утилитой -r (рис. 2).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir monthly.00
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp -r monthly monthly.00
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp -r monthly.00 /tmp
```

Рис. 2: Пример копирования файлов с утилитой

Пример перемещения и переименования файлов и каталогов (рис. 3).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv april july
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv july monthly.00
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ ls monthly.00
july  monthly
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv monthly.00 monthly.01
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir reports
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv monthly.01 reports
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv reports/monthly.01 reports/monthly
```

Рис. 3: Пример перемещения и переименования файлов и каталогов

Пример изменения прав доступа (рис. 4).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch may
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ ls -l may
-rw-rw-r-- 1 eakarpova eakarpova 0 map 18 10:25 may
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod u+x may
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ ls -l may
-rwxrw-r-- 1 eakarpova eakarpova 0 map 18 10:25 may
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod u-x may
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ ls -l may
-rw-rw-r-- 1 eakarpova eakarpova 0 map 18 10:25 may
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir monthly
mkdir: cannot create directory 'monthly': File exists
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod g-r, o-r monthly
chmod: invalid mode: 'g-r,'
Try 'chmod --help' for more information.
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 755 monthly
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ ещгср фис1
ещгср: command not found
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch abc1
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod g+w abc1
```

Рис. 4: Пример изменения прав доступа

2. Перемещение и переименования файлов и каталогов

Копирую файл `~/usr/include/aio.h` в домашний каталог и переименовываю его в `equipment`. Затем в домашнем каталоге создаю директорию `~/ski.plases` и перемещаю в нее этот файл. Переименовываю `equipment` в `equiplist`. Создаю в домашнем каталоге файл `abc1` и копирую его в `~/ski.plases` с новым названием `equiplist2` (рис. 5).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv aio.h equipment
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir ~/ski.plases
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv equipment ~/ski.plases
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/ski.plases/equipment ~/ski.
plases/equiplist
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch abc1
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp abc1 ~/ski.plases
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/ski.plases/abc1 ~/ski.plase
s/equiplist2
```

Рис. 5: Работа с директорией `ski.plases`

Создаю каталог `equipmnet` в `~/ski.plases` и перемещаю в него файлы `equiplist` и `equiplist2`, после чего создаю директорию `newdir` и перемещаю его в `~/ski.plases` с новым названием `plans` (рис. 6).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir ~/ski.plases/equipment
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/ski.plases/equiplist ~/ski.
plases/equiplist2 ~/ski.plases/equipment
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir ~/newdir
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/newdir ~/ski.plases
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/ski.plases/newdir ~/ski.pla
ses/plans
```

Рис. 6: Работа с директорией `equipment`

3. Изменение прав доступа

Создаю необходимые файлы с помощью команды touch. С помощью команды chmod и цифровой записи формы доступа присваиваю им выделенные права доступа (рис. 7).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch australia play my_os feathers
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 744 australia
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 711 play
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 754 my_os
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod 664 feathers
```

Рис. 7: Изменение прав доступа

4. Копирование файлов и каталогов

С помощью команды cat просматриваю содержимое файла /etc/passwd (рис. 8).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ 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:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
```

Рис. 8: Просмотр содержимого файла

Копирую файл ~/feathers в файл ~/file.old, после чего перемещаю его в каталог ~/play и копирую ~/play в директорию ~/fun (рис. 9).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp ~/feathers ~/file.old
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mkdir ~/play
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/file.old ~/play
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch ~/file.old
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp ~/feathers ~/file.old
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/file.old ~/play
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp ~/play ~/fun
```

Рис. 9: Работа с директорией ~/feathers

Перемещаю каталог ~/fun в ~/play и переименовываю его в games. С помощью команды chmod лишаю владельца файла права на чтение с помощью утилиты -г. Проверяю это командой cat - все сработало, так как мне отказано в доступе. Также я

не могу скопировать этот файл. Возвращаю владельцу файла право на чтение с помощью команды `chmod` и утилиты `+`. Затем лишаю владельца каталога `~/play` права на выполнение. После чего пытаюсь перейти в эту директорию и получаю ошибку - отказано в доступе, значит команда сработала верно. Возвращаю владельцу право на выполнение.(рис. 10).

```
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp -r ~/play ~/fun
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/fun ~/play
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ mv ~/play/fun ~/play/games
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod -r ~/feathers
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cat ~/feathers
cat: /home/eakarpova/feathers: Permission denied
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ touch newfile
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cp ~/feathers ~/newfile
cp: cannot open '/home/eakarpova/feathers' for reading: Permission denied
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod +r ~/feathers
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod -x ~/play
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ cd ~/play
bash: cd: /home/eakarpova/play: Permission denied
eakarpova@eakarpova-Redmi-Book-Pro-14-2022:~$ chmod +x ~/play
```

Рис. 10: Работа с правами доступа

5. Описание команд с помощью мануала

С помощью команды `man` узнаю характеристики команды `mount`: Команда `mount` в операционной системе Linux используется для подключения файловых систем к директориям в иерархии файловой системы. Это позволяет доступ к содержимому файловой системы. Пример использования: `mount /dev/sdb1 /mnt` Эта команда подключает устройство `/dev/sdb1` к директории `/mnt`. (рис. 11).

```

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|runbin
able]
    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

```

Рис. 11: Описание mount

С помощью команды `man` узнаю характеристики команды `fsck`: Команда `fsck` (File System Consistency Check) используется для проверки и исправления целостности файловой системы. Она помогает обнаружить и исправить ошибки на диске. Пример использования: `fsck /dev/sda1` Эта команда проверяет файловую систему на устройстве `/dev/sda1`. (рис. 12).


```

FCK(8)                                     System Administration      FCK(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

```

Рис. 12: Описание fsck

С помощью команды man узнаю характеристики команды mkfs: Команда mkfs (Make File System) используется для создания новой файловой системы на устройстве. Это позволяет инициализировать диск для использования. Пример использования: mkfs.ext4 /dev/sdc1 Эта команда создает файловую систему ext4 на устройстве /dev/sdc1.

(рис. 13).


```

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

```

Рис. 13: Описание mkfs

С помощью команды man узнаю характеристики команды kill: Команда kill используется для отправки сигнала процессу или группе процессов для завершения их работы. Это позволяет управлять процессами в системе. Пример использования: kill -9 1234 Эта команда отправляет сигнал SIGKILL процессу с идентификатором 1234 для принудительного завершения (рис. 14).

KILL(1)	User Commands	KILL(1)
NAME		
kill - send a signal to a process		
SYNOPSIS		
kill [options] <pid> [...]		
DESCRIPTION		
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.		
-q, --queue <u>value</u> Use sigqueue(3) rather than kill(2) and the value argument is used to specify an integer to be sent with the signal. If the receiving process has installed a handler for this signal using the SA_SIGINFO flag to sigaction(2) , then it can obtain this data via the si_value field of the siginfo_t structure.		
-l, --list [<u>signal</u>] List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.		
-L, --table List signal names in a nice table.		

Рис. 14: Описание kill

5 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.