

Лабораторная работа №5

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

Нилова Кристина

Российский университет дружбы народов, Москва, Россия

Информация

- Нилова Кристина
- студентка НБИбд-02-23
- Российский университет дружбы народов

Вводная часть

В данной лабораторной работе мы рассмотрим: - Команды для работы с файлами и каталогами - Копирование файлов и каталогов - Перемещение и переименование файлов и каталогов - Права доступа - Изменение прав доступа - Анализ файловой системы

- Команда touch - Команда fsck - Команда mkfs - Команда kill
- Команда mount - Команда cat - Команда less - Команда ls
- Команда chmod - Команда rm - Команда cp - Команда mv

Цель работы

Целью лабораторной работы является ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

Выполнение лабораторной работы

Первый этап

Выполняем все примеры, приведённые в первой части описания лабораторной работы.

```
kanilova@dk2n25 ~ $ mkdir feathers
kanilova@dk2n25 ~ $ cd feathers
kanilova@dk2n25 ~/feathers $ touch file.old
kanilova@dk2n25 ~/feathers $ cd
kanilova@dk2n25 ~ $ cp -r ~/feathers ~/file.old
kanilova@dk2n25 ~ $ mv ~/file.old ~/play
kanilova@dk2n25 ~ $ cp -r ~/play ~/fun
kanilova@dk2n25 ~ $ mv ~/fun ~/play
kanilova@dk2n25 ~ $ mv ~/fun/play ~/fun/games
mv: не удалось выполнить stat для '/afs/.dk.sci.pfu.edu.ru/home/k/a/kanilova/fun/play': Нет такого файла и
ли каталога
kanilova@dk2n25 ~ $ ls
arch-pc  dir1      feathers  parentdir  parentdir2  public_html  Доклад.md  Общедоступные
bin      dir2      GNUstep   parentdir3  parentdir3  tmp          Загрузки   'Рабочий стол'
chimp   dir3      hosts     parentdir   play        work        Изображения  'Снимки экрана'
chimp.pub Documents newdir1    parentdir1  public      Видео       Музыка       Шаблоны
kanilova@dk2n25 ~ $ cd play
kanilova@dk2n25 ~/play $ mv fun games
kanilova@dk2n25 ~/play $ ls
file.old  games
kanilova@dk2n25 ~/play $
```

Рис. 1: выполнение команд



Второй этап



2.1. Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. 2.2. В домашнем каталоге создадим директорию `~/ski.plases`. 2.3. Переместите файл `equipment` в каталог `~/ski.plases`. 2.4. Переименуйте файл `~/ski.plases/equipment` в `~/ski.plases/equiplist`.

```
kanilova@dk2n25 ~/play $ cd
kanilova@dk2n25 ~ $ cp /usr/include/sys/io.h equipment
kanilova@dk2n25 ~ $ ls
arch-pc  dir2      GNUstep  parentdir  public  Доклад.md  'Рабочий стол'
bin      dir3      hosts    parentdir1 public_html  Загрузки    'Снимки экрана'
chimp    Documents newdir1   parentdir2 tmp      Изображения  Шаблоны
chimp.pub equipment  parendir  parentdir3 work     Музыка
dir1     feathers  parendir3 play      Видео      Общедоступные
kanilova@dk2n25 ~ $ mkdir ~/ski.plases
kanilova@dk2n25 ~ $ ls
arch-pc  dir2      GNUstep  parentdir  public  Видео  Общедоступные
bin      dir3      hosts    parentdir1 public_html  Доклад.md  'Рабочий стол'
chimp    Documents newdir1   parentdir2 ski.plases  Загрузки    'Снимки экрана'
chimp.pub equipment  parendir  parentdir3 tmp      Изображения  Шаблоны
dir1     feathers  parendir3 play      work      Музыка
kanilova@dk2n25 ~ $ mv equipment ski.plases
kanilova@dk2n25 ~ $ mv ~/ski.plases/equipment ~/ski.plases/equiplist
kanilova@dk2n25 ~ $ ls ski.plases
equiplist
kanilova@dk2n25 ~ $
```

Рис. 2: выполнение команд

2.5. Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski.plases, назовите его equiplist2. 2.6. Создадим каталог с именем equipment в каталоге ~/ski.plases. 2.7. Переместим файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment. 2.8. Создадим и переместим каталог ~/newdir в каталог ~/ski.plases и назовем его plans.

```
equiplist
kanilova@dk2n25 ~ $ touch abc1
kanilova@dk2n25 ~ $ cp -r ~/ski.plases equiplist2
kanilova@dk2n25 ~ $ cd ~/ski.plases
kanilova@dk2n25 ~/ski.plases $ ls
equiplist
kanilova@dk2n25 ~/ski.plases $ cp ~/abc1 equiplist2
kanilova@dk2n25 ~/ski.plases $ ls
equiplist  equiplist2
kanilova@dk2n25 ~/ski.plases $ mkdir equipment
kanilova@dk2n25 ~/ski.plases $ ls
equiplist  equiplist2  equipment
kanilova@dk2n25 ~/ski.plases $ mv equiplist equiplist2 equipment
kanilova@dk2n25 ~/ski.plases $ ls equipment
equiplist  equiplist2
kanilova@dk2n25 ~/ski.plases $ mkdir ~/newdir
kanilova@dk2n25 ~/ski.plases $ mv ~/newdir plans
kanilova@dk2n25 ~/ski.plases $ ls
equipment  plans
kanilova@dk2n25 ~/ski.plases $
```

Рис. 3: выполнение команд

Третий этап

Определим опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет

drwxr-r- ... australia

```
kanilova@dk2n25 ~ $ mkdir australia
kanilova@dk2n25 ~ $ chmod g-x australia
kanilova@dk2n25 ~ $ chmod o-x australia
kanilova@dk2n25 ~ $ ls -l australia
итого 0
kanilova@dk2n25 ~ $ ls -l
итого 86
-rw-r--r-- 1 kanilova studsci    0 map 15 16:35 abc1
drwxr-xr-x 5 kanilova studsci 2048 фев 7 23:40 arch-pc
drwxr--r-- 2 kanilova studsci 2048 map 15 16:39 australia
```

Рис. 4: результат

drwx-x-x ... play

```
drwxr-xr-x 2 kanilova studsci 2048 сен 7 2023
kanilova@dk2n25 ~ $ chmod o-r play
kanilova@dk2n25 ~ $ s -l
bash: s: команда не найдена
kanilova@dk2n25 ~ $ ls -l
итого 86
-rwxr-xr-x 1 kanilova studsci 0 мар 15 16:25
```

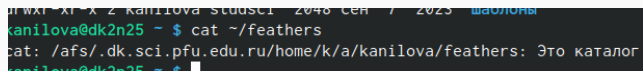
Рис. 5: результат

-r-xr-r- ... my_os

```
kanilova@dk2n25 ~ $ chmod u-w my_os
kanilova@dk2n25 ~ $ chod u+w my_os
bash: chod: команда не найдена
kanilova@dk2n25 ~ $ chmod u+w my_os
kanilova@dk2n25 ~ $ ls -l
итого 88
-rw-r--r-- 1 kanilova studsci      0 мар 15 16:35 abc1
drwxr-xr-x 5 kanilova studsci 2048 фев  7 23:40 arch-pc
drwxr--r-- 2 kanilova studsci 2048 мар 15 16:39 australia
```

Рис. 6: результат

-rw-rw-r- ... feathers

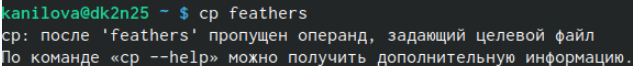
A terminal window with a dark background. The prompt is 'kanilova@dk2n25 ~ \$'. The command entered is 'cat ~/feathers'. The output is 'cat: /afs/.dk.sci.pfu.edu.ru/home/k/a/kanilova/feathers: Это каталог'.

```
kanilova@dk2n25 ~ $ cat ~/feathers  
cat: /afs/.dk.sci.pfu.edu.ru/home/k/a/kanilova/feathers: Это каталог  
kanilova@dk2n25 ~ $
```

Рис. 7: результат

Четвёртый этап

4.1. Просмотришь содержимое файла /etc/password. (такой файл отсутствует) 4.2. Скопируйте файл ~/feathers в файл ~/file.old.ls 4.3. Переместите файл ~/file.old в каталог ~/play. 4.4. Скопируйте каталог ~/play в каталог ~/fun. 4.5. Переместите каталог ~/fun в каталог ~/play и назовите его games.



```
kanilova@dk2n25 ~ $ cp feathers
cp: после 'feathers' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
```

Рис. 8: выполнение команд

4.6. Лишим владельца файла ~/feathers права на чтение. 4.7. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat? 4.8. Что произойдёт, если вы попытаетесь скопировать файл ~/feathers?

```
kanilova@dk2n25 ~$ chmod u-r feathers
kanilova@dk2n25 ~$ chmod u+r fethers
chmod: невозможно получить доступ к 'fethers': Нет такого файла или каталога
kanilova@dk2n25 ~$ chmod u+r feathers
kanilova@dk2n25 ~$
```

Рис. 9: выполнение команд

4.9. Дайте владельцу файла ~/feathers право на чтение.

```
kanilova@dk2n25 ~ $ chmod u-x ~/play
kanilova@dk2n25 ~ $ cd play
kanilova@dk2n25 ~/play $ ls -l
итого 2
-rw-r--r-- 1 kanilova studsci    0 мар 15 16:27 file.old
drwxr-xr-x 2 kanilova studsci 2048 мар 15 16:28 games
kanilova@dk2n25 ~/play $
```

Рис. 10: выполнение команд

4.10. Лишим владельца каталога ~/play права на выполнение. 4.11. Перейдем в каталог ~/play. Что произошло? ничего не произошло(

```
kanilova@dk2n25 ~/play $ cd
kanilova@dk2n25 ~ $ chmod u+x ~/play
kanilova@dk2n25 ~ $ ls -l
итого 88
-rw-r--r-- 1 kanilova studsci    0 мар 15 16:35 abc1
drwxr-xr-x 5 kanilova studsci 2048 фев  7 23:40 arch-pc
drwxr--r-- 2 kanilova studsci 2048 мар 15 16:39 australia
drwxr-xr-x 2 kanilova studsci 2048 мар  6 15:48 bin
-rw----- 1 kanilova studsci 2622 окт 27 15:22 chimp
-rw-r--r-- 1 kanilova studsci  583 окт 27 15:22 chimp.pub
drwxr-xr-x 2 kanilova studsci 2048 сен 14  2023 dir1
drwxr-xr-x 2 kanilova studsci 2048 сен 14  2023 dir2
drwxr-xr-x 2 kanilova studsci 2048 сен 14  2023 dir3
drwxr-xr-x 2 kanilova studsci 2048 окт  9 19:40 Documents
drwxr-xr-x 2 kanilova studsci 2048 мар 15 16:36 equiplist2
drwxr-xr-x 2 kanilova studsci 2048 мар 15 16:26 feathers
drwxr-xr-x 2 kanilova studsci 2048 мар  5 16:52 GNUstep
```

Рис. 11: выполнение команд

4.12. Дайте владельцу каталога ~/play право на выполнение.

```

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
  
```

Пятый этап



Прочитаем man по командам mount, fsck, mkfs, kill и кратко их охарактеризуем, приведя примеры.

```

fsck(8)                                System Administration                                fsck(8)
NAME
    fsck - check and repair a Linux filesystem
SYNOPSIS
    fsck [-lsAvrtnp] [-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=8866aaf6-8ac5-4a93-9408-bfc24057770d 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

Manual page fsck(8) line 1 (press h for help or q to quit)

```

Рис. 13: man

```

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.

```

Manual page mkfs(8) line 1 (press h for help or q to quit)

```

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 value
        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 [signal]
        List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.

    -L, --table

```

Manual page kill(1) line 1 (press h for help or q to quit)

```
drwxr-xr-x 2 kanilova studsci 2048 сен 7 2023 wa
kanilova@dk2n25 ~ $ man mount
kanilova@dk2n25 ~ $ man fsck
kanilova@dk2n25 ~ $ man mkfs
kanilova@dk2n25 ~ $ man kill
kanilova@dk2n25 ~ $
```

Рис. 16: man mkfs

Вывод

Мы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрели практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.