

Лабораторная работа №5

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

Кузнецова София Вадимовна

9 марта 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Кузнецова София Вадимовна
- Студенческий билет - 1132227132
- НПИбд-02-22
- Российский университет дружбы народов

Цели и задачи

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

Копируем файл `/usr/include/sys/io.h` в домашний каталог и назовём его `equipment`, используя команду `cp`.

```
[svkuznecova@fedora ~]$ cp /usr/include/sys/io.h ~/equipment
[svkuznecova@fedora ~]$ ls
abcl      id_rsa.pub  monthly    -o.pub     reports    Видео      Загрузки    Музыка      'Рабочий стол'
equipment may        -o         os-intro   work       Документы  Изображения  Общедоступные  Шаблоны
[svkuznecova@fedora ~]$
```

Рис. 1: Файл `equipment`

Создаём каталог ~/ski.plases и переместив в созданную директорию файл equipment.

```
[svkuznecova@fedora ~]$ mkdir ski.plases
[svkuznecova@fedora ~]$ ls
abcl      id_rsa.pub  monthly    -o.pub     reports    work       Документы  Изображения  Общедоступные  Шаблоны
equipment may         -o         os-intro   ski.plases Видео      Загрузки   Музыка       'Рабочий стол'
```

```
[svkuznecova@fedora ~]$ mv equipment ski.plases
[svkuznecova@fedora ~]$ ls ski.plases
equipment
[svkuznecova@fedora ~]$
```

Рис. 2: Перемещение файла equipment в каталог ~/ski.plases

Создаём файл abc1 (команда touch) и копируем его в каталог ski.places, назовём equiplist2.

```
[svkuznecova@fedora ~]$ touch abc1
[svkuznecova@fedora ~]$ cp abc1 ski.places/equiplist2
[svkuznecova@fedora ~]$ ls ski.places
equiplist  equiplist2
[svkuznecova@fedora ~]$
```

Рис. 3: Каталог ski.places

Создаём файлы my_os и feathers и каталоги australia и play, присвоим им следующие права доступа используя команду chmod.

```
[svkuznecova@fedora ~]$ mkdir australia play
[svkuznecova@fedora ~]$ touch my_os feathers
[svkuznecova@fedora ~]$ ls
abcl      id_rsa.pub  my_os      os-intro   ski.places  Документы  Музыка  Шаблоны
australia may         -o         play       work        Загрузки  Общедоступные
feathers  monthly    -o.pub     reports    Видео       Изображения  'Рабочий стол'
```

Рис. 4: Создание необходимых файлов и каталогов

```
[svkuznecova@fedora ~]$ chmod 744 australia
[svkuznecova@fedora ~]$ chmod 711 play
[svkuznecova@fedora ~]$ chmod 544 my_os
[svkuznecova@fedora ~]$ chmod 644 feathers
[svkuznecova@fedora ~]$ ls -lt
итого 12
-rw-r--r--. 1 svkuznecova svkuznecova 0 map 8 02:36 feathers
-r-xr--r--. 1 svkuznecova svkuznecova 0 map 8 02:36 my_os
drwxr--r--. 1 svkuznecova svkuznecova 0 map 8 02:35 australia
drwx--x--x. 1 svkuznecova svkuznecova 0 map 8 02:35 play
```

Рис. 5: Изменение прав доступа для файлов и каталогов

Просмотрим содержимое файла `/etc/passwd`.

```
[svkuznecova@fedora ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/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:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
```

Рис. 6: Файл passwd

Переместим файл ~/file.old в каталог ~/play.

```
[svkuznecova@fedora ~]$ mv file.old play  
[svkuznecova@fedora ~]$ ls play  
file.old  
[svkuznecova@fedora ~]$
```

Рис. 7: Перемещение файла ~/file.old

Лишим владельца файла ~/feathers права на чтение, попробуем посмотреть его и скопировать. В результате без прав на чтение владелец файла не сможет посмотреть и скопировать этот файл.

```
[svkuznecova@fedora ~]$ chmod u-r feathers
[svkuznecova@fedora ~]$ ls -lt
итого 12
drwx--x--x. 1 svkuznecova svkuznecova 26 мар  8 02:46 play
--w-r--r--. 1 svkuznecova svkuznecova  0 мар  8 02:36 feathers
```

Рис. 8: Изменение прав доступа на чтение к файду

```
[svkuznecova@fedora ~]$ cat feathers
cat: feathers: Отказано в доступе
[svkuznecova@fedora ~]$ cp feathers feathers2
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
[svkuznecova@fedora ~]$
```

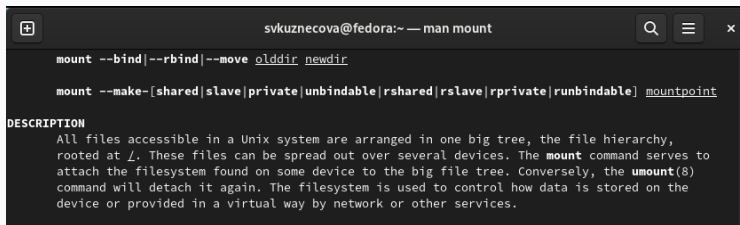
Рис. 9: Попытка изменить файл и скопировать его

Лишаем владельца каталога ~/play права на выполнение и попробуем перейти в него. Без прав на выполнение владелец не сможет перейти в каталог. Далее восстановим владельцу права на выполнение.

```
[svkuznecova@fedora ~]$ chmod u-x play  
[svkuznecova@fedora ~]$ cd play  
bash: cd: play: Отказано в доступе  
[svkuznecova@fedora ~]$ chmod u+x play  
[svkuznecova@fedora ~]$
```

Рис. 10: Изменение прав доступа на выполнение каталогу play

Команда mount используется для просмотра используемых в операционной системе файловых систем.



```
svkuznecova@fedora:~ — man mount

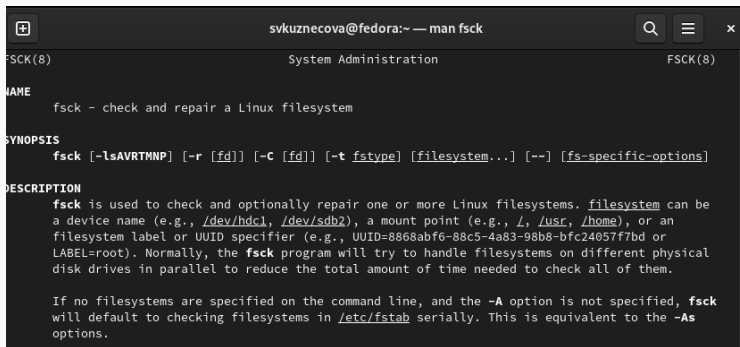
mount --bind|--rbind|--move olddir newdir

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.
```

Рис. 11: man mount

Команда fsck нужна для проверки и восстановления файловой системы.



```
svkuznecova@fedora:~ — man fsck
FSCK(8)                                     System Administration                                     FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

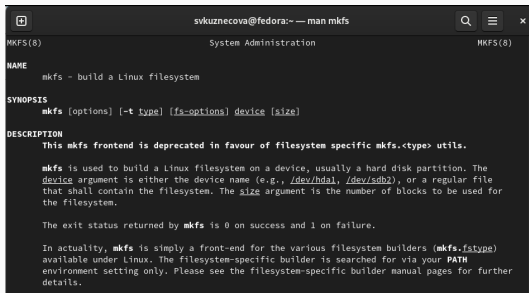
SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--] [fs-specific-options]

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 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.
```

Рис. 12: man fsck

Команда `mkfs` используется для создания файловой системы Linux на устройстве, обычно в разделе диска.



```
svkuznecova@fedora:~ — man 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.

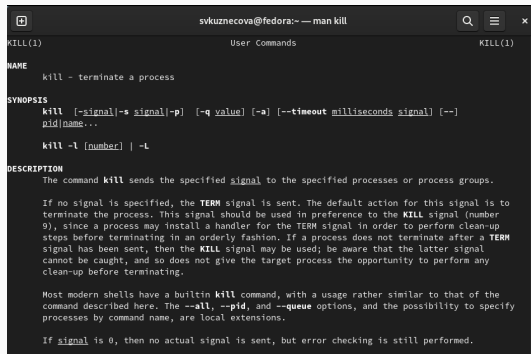
    The exit status returned by mkfs is 0 on success and 1 on failure.

    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.
```

Рис. 13: man mkfs

Команда kill

Команда kill отправляет сигнал процессу, указанному с помощью каждого из операндов идентификатора процесса.



```
svkuznecova@fedora:~ — man 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.
```

Рис. 14: man kill

Вывод

В ходе выполнения лабораторной работы 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.