

Лабораторная работа №5

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

Мальков Роман

Содержание

Цель работы	3
Задание	4
Теоретическое введение	6
Ход работы	8
1	8
2	13
3	16
4	20
5	22
Выводы	26

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.

Задание

1. Выполните все примеры, приведённые в первой части описания лабораторной работы.
2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
 - 2.1. 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.
 - 2.2. В домашнем каталоге создайте директорию `~/ski.places`.
 - 2.3. Переместите файл `equipment` в каталог `~/ski.places`.
 - 2.4. Переименуйте файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 - 2.5. Создайте в домашнем каталоге файл `abc1` и скопируйте его в каталог `~/ski.places`, назовите его `equiplist2`.
 - 2.6. Создайте каталог с именем `equipment` в каталоге `~/ski.places`.
 - 2.7. Переместите файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 - 2.8. Создайте и переместите каталог `~/newdir` в каталог `~/ski.places` и назовите его `plans`.
3. Определите опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:
 - 3.1. `drwxr-r- ... australia`

- 3.2. `drwx-x-x ... play`
 - 3.3. `-r-xr-r- ... my_os`
 - 3.4. `-rw-rw-r- ... feathers` При необходимости создайте нужные файлы.
4. Прделайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
- 4.1. Просмотрите содержимое файла `/etc/password`.
 - 4.2. Скопируйте файл `~/feathers` в файл `~/file.old`.
 - 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`.
 - 4.6. Лишите 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`?
 - 4.9. Дайте владельцу файла `~/feathers` право на чтение.
 - 4.10. Лишите владельца каталога `~/play` права на выполнение.
 - 4.11. Перейдите в каталог `~/play`. Что произошло?
 - 4.12. Дайте владельцу каталога `~/play` право на выполнение.
5. Прочитайте ман по командам `mount`, `fsck`, `mkfs`, `kill` и кратко их охарактеризуйте, приведя примеры.
- 6.

Теоретическое введение

Команды для работы с файлами и каталогами

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
```

Следующие клавиши используются для управления процессом просмотра:

- `Space` — переход к следующей странице,
- `ENTER` — сдвиг вперёд на одну строку,
- `b` — возврат на предыдущую страницу,
- `h` — обращение за подсказкой,
- `q` — выход из режима просмотра файла.

Команда `head` выводит по умолчанию первые 10 строк файла.

Формат команды:

```
head [-n] имя-файла,
```

где `n` — количество выводимых строк.

Команда `tail` выводит по умолчанию 10 последних строк файла.

Формат команды:

```
tail [-n] имя-файла,
```

где `n` — количество выводимых строк.

Рис. 1: Screenshot_26

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).

Право	Обозначение	Файл	Каталог	Формы записи пр	
				Двоичная	Восьмеричн
Чтение	r	Разрешены просмотр и копирование	Разрешён просмотр списка входящих файлов	111	7
Запись	w	Разрешены изменение и переименование	Разрешены создание и удаление файлов	110	6
				101	5
				100	4
Выполнение	x	Разрешено выполнение файла (скриптов и/или программ)	Разрешён доступ в каталог и есть возможность сделать его текущим	011	3
				010	2
				001	1
				000	0

chmod u+r modb к примеру данная команда меняет права доступа владельца, добавляя ему возможность чтения файла. chmod 300 modb а данный пример команды уже разрешает владельцу запись и выполнение файла без возможности его чтения, в тоже время забирая возможность других пользователей читать, записывать, и выполнять файл.

Ход работы

1

Выполняем копирование файла в текущем каталоге. Для этого используем следующие команды:

```
touch abc1  
cp abc1 april  
cp abc1 may
```

Далее выполняем копирование нескольких файлов(may и april) в каталог(monthly) одной командой(тут я немного ошибся и скопировал их двумя отдельными командами, следует быть внимательнее):

```
mkdir monthly  
cp april may monthly
```

Копирование файлов в произвольном каталоге:

```
cp monthly/may monthly/june
```

Копирование каталога в текущем каталоге:

```
mkdir monthly.00  
cp -r monthly monthly.00
```

Копирование каталогов в произвольном каталоге:


```
cp -r monthly.00 /tmp
```

Переименование файлов в текущем каталоге. Изменить название файла april на july в домашнем каталоге:

```
mv april july
```

Перемещение файлов в другой каталог. Переместить файл july в каталог monthly.00:

```
mv july monthly.00
```

```
ls monthly.00
```

Переименование каталогов в текущем каталоге. Переименовать каталог monthly.00 в monthly.01:

```
mv monthly.00 monthly.01
```

Перемещение каталога в другой каталог. Переместить каталог monthly.01 в каталог reports:

```
mkdir reports
```

```
mv monthly.01 reports
```

Переименование каталога, не являющегося текущим. Переименовать каталог reports/monthly.01 в reports/monthly:

```
mv reports/monthly.01 reports/monthly
```

Меняем права владельца файла ~/may, дав тем самым ему права на выполнение:

```
chmod u+x may
```

Лишаем владельца прав на выполнение:

```
chmod u-x may
```

Меняем права доступа monthly, запретив чтение для всех членов группы и всех остальных пользователей:

```
chmod g-r, o-r monthly
```

Меняем права доступа к файлу ~/abc1, предоставив право записи членам группы:

```
chmod g+w abc1
```

Скриншоты 1-2 к вышеописанным действиям прилагаются ниже:

```
rsmaljkov@rsmaljkov:~  
[rsmaljkov@rsmaljkov ~]$ touch abc1  
[rsmaljkov@rsmaljkov ~]$ cp abc1 may  
[rsmaljkov@rsmaljkov ~]$ cp abc1 april  
[rsmaljkov@rsmaljkov ~]$ ls  
abc1 Desktop Downloads Music Public tmp work  
april Documents may Pictures Templates Videos  
[rsmaljkov@rsmaljkov ~]$ cp abc1 april  
[rsmaljkov@rsmaljkov ~]$ ls  
abc1 Desktop Downloads Music Public tmp work  
april Documents may Pictures Templates Videos  
[rsmaljkov@rsmaljkov ~]$ mkdir monthly  
[rsmaljkov@rsmaljkov ~]$ cp may monthly  
[rsmaljkov@rsmaljkov ~]$ cp april monthly  
[rsmaljkov@rsmaljkov ~]$ cp monthly/may monthly/june  
[rsmaljkov@rsmaljkov ~]$ mkdir monthly.00  
[rsmaljkov@rsmaljkov ~]$ cp -r monthly monthly.00  
[rsmaljkov@rsmaljkov ~]$ cp -r monthly.00 /tmp  
[rsmaljkov@rsmaljkov ~]$ mv april july  
[rsmaljkov@rsmaljkov ~]$ mv july monthly.00  
[rsmaljkov@rsmaljkov ~]$ ls  
abc1 Documents may monthly.00 Pictures Templates Videos  
Desktop Downloads monthly Music Public tmp work  
[rsmaljkov@rsmaljkov ~]$ cd montly.00  
bash: cd: montly.00: No such file or directory  
[rsmaljkov@rsmaljkov ~]$ cd monthly.00  
[rsmaljkov@rsmaljkov monthly.00]$ cd  
[rsmaljkov@rsmaljkov ~]$ cd monthly.00  
[rsmaljkov@rsmaljkov monthly.00]$ cd  
[rsmaljkov@rsmaljkov ~]$ mv monthly.00 monthly.01  
[rsmaljkov@rsmaljkov ~]$ ls  
abc1 Documents may monthly.01 Pictures Templates Videos  
Desktop Downloads monthly Music Public tmp work  
[rsmaljkov@rsmaljkov ~]$ mkdir reports  
[rsmaljkov@rsmaljkov ~]$ mv monthly.01 reports  
[rsmaljkov@rsmaljkov ~]$ mv reports/monthly.01 reports/monthly  
[rsmaljkov@rsmaljkov ~]$ ls -l may  
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May 4 11:39 may  
[rsmaljkov@rsmaljkov ~]$ chmod u+x may  
[rsmaljkov@rsmaljkov ~]$ ls -l may  
-rwxrw-r--. 1 rsmaljkov rsmaljkov 0 May 4 11:39 may  
[rsmaljkov@rsmaljkov ~]$ 1 chmod u-x may  
bash: 1: command not found...  
[rsmaljkov@rsmaljkov ~]$ chmod u-x may  
[rsmaljkov@rsmaljkov ~]$ ls -l may  
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May 4 11:39 may  
[rsmaljkov@rsmaljkov ~]$ chmod g-r, o-r monthly  
chmod: invalid mode: 'g-r,'  
Try 'chmod --help' for more information.
```

(Скриншот 1)

```
rsmaljkov@rsmaljkov:~  
[rsmaljkov@rsmaljkov ~]$ mv monthly.01 reports  
[rsmaljkov@rsmaljkov ~]$ mv reports/monthly.01 reports/monthly  
[rsmaljkov@rsmaljkov ~]$ ls -l may  
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May  4 11:39 may  
[rsmaljkov@rsmaljkov ~]$ chmod u+x may  
[rsmaljkov@rsmaljkov ~]$ ls -l may  
-rwxrw-r--. 1 rsmaljkov rsmaljkov 0 May  4 11:39 may  
[rsmaljkov@rsmaljkov ~]$ 1 chmod u-x may  
bash: 1: command not found...  
[rsmaljkov@rsmaljkov ~]$ chmod u-x may  
[rsmaljkov@rsmaljkov ~]$ ls -l may  
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May  4 11:39 may  
[rsmaljkov@rsmaljkov ~]$ chmod g-r, o-r monthly  
chmod: invalid mode: 'g-r,'  
Try 'chmod --help' for more information.  
[rsmaljkov@rsmaljkov ~]$ chmod --help  
Usage: chmod [OPTION]... MODE[,MODE]... FILE...  
  or: chmod [OPTION]... OCTAL-MODE FILE...  
  or: chmod [OPTION]... --reference=RFILE FILE...  
Change the mode of each FILE to MODE.  
With --reference, change the mode of each FILE to that of RFILE.  
  
-c, --changes          like verbose but report only when a change is made  
-f, --silent, --quiet  suppress most error messages  
-v, --verbose          output a diagnostic for every file processed  
  --no-preserve-root    do not treat '/' specially (the default)  
  --preserve-root       fail to operate recursively on '/'  
  --reference=RFILE     use RFILE's mode instead of MODE values  
-R, --recursive        change files and directories recursively  
  --help               display this help and exit  
  --version             output version information and exit  
  
Each MODE is of the form '[ugoa]*([-+=[rwxXst]*|[ugo]))+|[-+=[0-7]+'.  
  
GNU coreutils online help: <https://www.gnu.org/software/coreutils/>  
Full documentation <https://www.gnu.org/software/coreutils/chmod>  
or available locally via: info '(coreutils) chmod invocation'  
[rsmaljkov@rsmaljkov ~]$ chmod g-r, o-r monthly  
chmod: invalid mode: 'g-r,'  
Try 'chmod --help' for more information.  
[rsmaljkov@rsmaljkov ~]$ chmod g-r , o-r monthly  
chmod: cannot access ',': No such file or directory  
chmod: cannot access 'o-r': No such file or directory  
[rsmaljkov@rsmaljkov ~]$ chmod g-r o-r monthly  
chmod: cannot access 'o-r': No such file or directory  
[rsmaljkov@rsmaljkov ~]$ chmod g-r,o-r monthly  
[rsmaljkov@rsmaljkov ~]$ chmod g+w abc1  
[rsmaljkov@rsmaljkov ~]$
```

(Скриншот 2)

2

Копируем файл `/usr/include/sys/io.h` в домашний каталог и называем его `equipment`:

```
cp /usr/include/sys/io.h /home/rsmaljkov  
mv io.h equipment
```

В домашнем каталоге создаем директорию `~/ski.places`:

```
mkdir ski.places
```

Переещаем файл `equipment` в каталог `~/ski.places` и переименовываем в `equiplist`:

```
mv equipment ~/ski.places/equiplist
```

Создаем в домашнем каталоге файл `abc1` и копируем его в каталог `~/ski.places`, называем его `equiplist2`:

```
cp abc1 ~/ski.places  
mv ~/ski.places/equiplist2
```

Создаем каталог с именем `equipment` в каталоге `~/ski.places`:

```
mkdir ski.places/equipment
```

Перемещаем файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`:

```
mv ~/ski.places/equiplist ~/ski.places/equipment/equiplist  
mv ~/ski.places/equiplist2 ~/ski.places/equipment/equiplist2
```

Создаем и переместите каталог `~/newdir` в каталог `~/ski.places` и назовите его `plans`:

```
mkdir newdir
```

```
mv newdir ski.places/plans
```

Скриншот 3 ,описывающий все вышеперечисленные действия пункта 2, прикреплен

```
rsmaljkov@rsmaljkov:~  
[rsmaljkov@rsmaljkov ~]$ cp ~/usr/include/sys/io.h  
cp: missing destination file operand after '/home/rsmaljkov/usr/include/sys/io.h'  
Try 'cp --help' for more information.  
[rsmaljkov@rsmaljkov ~]$ cp /usr/include/sys/io.h  
cp: missing destination file operand after '/usr/include/sys/io.h'  
Try 'cp --help' for more information.  
[rsmaljkov@rsmaljkov ~]$ cp /usr/include/sys/io.h ~/home/rsmaljkov  
cp: cannot create regular file '/home/rsmaljkov/home/rsmaljkov': No such file or  
directory  
[rsmaljkov@rsmaljkov ~]$ cp /usr/include/sys/io.h /home/rsmaljkov  
[rsmaljkov@rsmaljkov ~]$ ls  
abc1 Documents io.h monthly Pictures reports tmp work  
Desktop Downloads may Music Public Templates Videos  
[rsmaljkov@rsmaljkov ~]$ mv io.h equipment  
[rsmaljkov@rsmaljkov ~]$ ls  
abc1 Documents equipment monthly Pictures reports tmp work  
Desktop Downloads may Music Public Templates Videos  
[rsmaljkov@rsmaljkov ~]$ mkdir ski.places  
[rsmaljkov@rsmaljkov ~]$ ls  
abc1 Documents equipment monthly Pictures reports Templates Videos  
Desktop Downloads may Music Public ski.places tmp work  
[rsmaljkov@rsmaljkov ~]$ mv equipment ~/ski.places/equiplist  
[rsmaljkov@rsmaljkov ~]$ ls  
abc1 Documents may Music Public ski.places tmp work  
Desktop Downloads monthly Pictures reports Templates Videos  
[rsmaljkov@rsmaljkov ~]$ cd ski.places  
[rsmaljkov@rsmaljkov ski.places]$ ;s  
bash: syntax error near unexpected token `;'  
[rsmaljkov@rsmaljkov ski.places]$ ls  
equiplist  
[rsmaljkov@rsmaljkov ski.places]$ cd  
[rsmaljkov@rsmaljkov ~]$ cp abc1 ~/ski.places  
[rsmaljkov@rsmaljkov ~]$ mv /ski.places/abc1 /ski.places/equiplist2  
mv: cannot stat '/ski.places/abc1': No such file or directory  
[rsmaljkov@rsmaljkov ~]$ mv ~/ski.places/abc1 ~/ski.places/equiplist2  
[rsmaljkov@rsmaljkov ~]$ mkdir ski.places/equipment  
[rsmaljkov@rsmaljkov ~]$ mv ~/ski.places/equiplist ~/ski.places/equipment/equip  
list  
[rsmaljkov@rsmaljkov ~]$ mv ~/ski.places/equiplist2 ~/ski.places/equipment/equip  
list2  
[rsmaljkov@rsmaljkov ~]$ mkdir newdir  
[rsmaljkov@rsmaljkov ~]$ mv newdir ski.places/plans  
[rsmaljkov@rsmaljkov ~]$
```

ниже.

(Скриншот 3)

3

Определяем опции команды `chmod`, необходимые для того, чтобы присвоить выделенные права доступа, предварительно создав эти файлы(Скриншоты 4 - 8):

```
mkdir australia
```

```
mkdir play
```

```
touch my_os
```

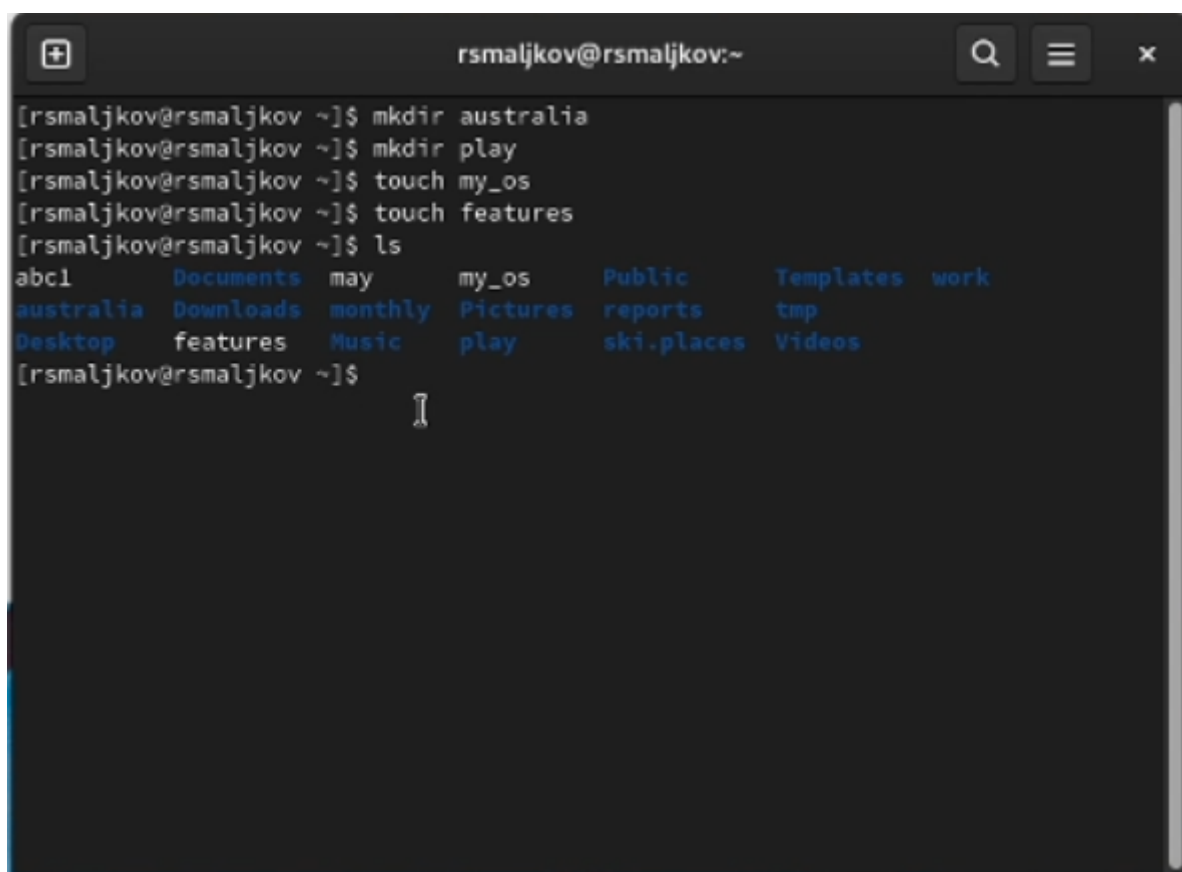
```
touch features
```

```
chmod 744 australia
```

```
chmod 711 play
```

```
chmod 544 my_os
```

```
chmod 654 features
```

A terminal window titled "rsmaljkov@rsmaljkov:~" with standard macOS window controls (zoom, menu, close). The terminal shows a series of commands and their outputs:

```
[rsmaljkov@rsmaljkov ~]$ mkdir australia
[rsmaljkov@rsmaljkov ~]$ mkdir play
[rsmaljkov@rsmaljkov ~]$ touch my_os
[rsmaljkov@rsmaljkov ~]$ touch features
[rsmaljkov@rsmaljkov ~]$ ls
abcl      Documents  may        my_os      Public     Templates  work
australia Downloads  monthly    Pictures    reports    tmp
Desktop   features   Music      play        ski.places Videos
[rsmaljkov@rsmaljkov ~]$
```

The output of the `ls` command lists files and directories in the home directory, including `abcl`, `Documents`, `may`, `my_os`, `Public`, `Templates`, `work`, `australia`, `Downloads`, `monthly`, `Pictures`, `reports`, `tmp`, `Desktop`, `features`, `Music`, `play`, `ski.places`, and `Videos`. A cursor is visible on the line following the last prompt.

Рис. 1: Screenshot_3

```
[rsmaljkov@rsmaljkov ~]$ chmod 744 australia
[rsmaljkov@rsmaljkov ~]$ ls -l australia
total 0
[rsmaljkov@rsmaljkov ~]$ ls -l
total 0
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May 4 11:38 abc1
drwxr--r--. 1 rsmaljkov rsmaljkov 0 May 4 12:32 australia
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Desktop
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 28 13:39 Documents
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Downloads
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May 4 12:33 features
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May 4 11:39 may
drwx-wx--x. 1 rsmaljkov rsmaljkov 24 May 4 11:40 monthly
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Music
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May 4 12:32 my_os
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Pictures
drwxrwxr-x. 1 rsmaljkov rsmaljkov 0 May 4 12:32 play
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Public
drwxrwxr-x. 1 rsmaljkov rsmaljkov 14 May 4 11:51 reports
drwxrwxr-x. 1 rsmaljkov rsmaljkov 28 May 4 12:16 ski.places
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Templates
drwxr-xr-x. 1 rsmaljkov rsmaljkov 36 Apr 27 15:45 tmp
```

Рис. 2: Screenshot_4

```

[rsmaljkov@rsmaljkov ~]$ chmod 711 play
[rsmaljkov@rsmaljkov ~]$ ls -l
total 0
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May  4 11:38 abc1
drwxr--r--. 1 rsmaljkov rsmaljkov 0 May  4 12:32 australia
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Desktop
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 28 13:39 Documents
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Downloads
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May  4 12:33 features
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May  4 11:39 may
drwx-wx--x. 1 rsmaljkov rsmaljkov 24 May  4 11:40 monthly
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Music
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May  4 12:32 my_os
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Pictures
drwx--x--x. 1 rsmaljkov rsmaljkov 0 May  4 12:32 play
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Public
drwxrwxr-x. 1 rsmaljkov rsmaljkov 14 May  4 11:51 reports
drwxrwxr-x. 1 rsmaljkov rsmaljkov 28 May  4 12:16 ski.places
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Templates
drwxr-xr-x. 1 rsmaljkov rsmaljkov 36 Apr 27 15:45 tmp
drwxr-xr-x. 1 rsmaljkov rsmaljkov 0 Apr 18 18:29 Videos
drwxrwxr-x. 1 rsmaljkov rsmaljkov 10 Apr 23 16:12 work
[rsmaljkov@rsmaljkov ~]$

```

Рис. 3: Screenshot_5

```

[rsmaljkov@rsmaljkov ~]$ chmod 544 my_os
[rsmaljkov@rsmaljkov ~]$ ls -l my_os
-r-xr--r--. 1 rsmaljkov rsmaljkov 0 May  4 12:32 my_os
[rsmaljkov@rsmaljkov ~]$

```

Рис. 4: Screenshot_6

```

[rsmaljkov@rsmaljkov ~]$ chmod 664 features
[rsmaljkov@rsmaljkov ~]$ ls -l features
-rw-rw-r--. 1 rsmaljkov rsmaljkov 0 May  4 12:33 features
[rsmaljkov@rsmaljkov ~]$

```

Рис. 5: Screenshot_8

(Скриншоты 4 - 8)

Просматриваем содержимое файла `/etc/passwd`, затем копируем файл `~/features` в файл `~/file.old`, перемещаем `~/file.old` в каталог `~/play`. После этого, копируем каталог `~/play` в каталог `~/fun`, перемещаем каталог `~/fun` в каталог `~/play` и называем его `games`. Далее мы лишаем владельца файла `~/features` права на чтение, пытаемся открыть и скопировать файл, и смотрим на результат, после этого возвращаем право на чтение. Затем лишаем владельца права на выполнение каталога `~/play`, переходим в каталог, смотрим результат, а после возвращаем право на выполнение(9-19).

```
[rsmaljkov@rsmaljkov ~]$ sudo -s
[sudo] password for rsmaljkov:
[root@rsmaljkov ~]# ls etc/passwd
[root@rsmaljkov ~]# cp home/rsmaljkov/features etc/passwd/file.old
[root@rsmaljkov ~]# ls etc/passwd
file.old
[root@rsmaljkov ~]#
```

Рис. 6: Screenshot_12

```
[root@rsmaljkov ~]# mv etc/passwd/file.old home/rsmaljkov/play
```

Рис. 7: Screenshot_13

```
[root@rsmaljkov ~]# mkdir home/rsmaljkov/fun
[root@rsmaljkov ~]# cp home/rsmaljkov/play home/rsmaljkov/fun
cp: -r not specified; omitting directory 'home/rsmaljkov/play'
[root@rsmaljkov ~]# cp -r home/rsmaljkov/play home/rsmaljkov/fun
```

Рис. 8: Screenshot_14

```
[root@rsmaljkov /]# mv home/rsmaljkov/fun home/rsmaljkov/play/games
```

Рис. 9: Screenshot_15

```
[root@rsmaljkov rsmaljkov]# chmod u-r features  
[root@rsmaljkov rsmaljkov]#
```

Рис. 10: Screenshot_16

```
[root@rsmaljkov rsmaljkov]# chmod u-r features  
[root@rsmaljkov rsmaljkov]# ls -l features  
--wxrwxrwx. 1 rsmaljkov rsmaljkov 0 May  4 12:33 features  
[root@rsmaljkov rsmaljkov]#
```

Рис. 11: Screenshot_17

```
[rsmaljkov@rsmaljkov ~]$ cat features  
cat: features: Permission denied  
[rsmaljkov@rsmaljkov ~]$ cp features etc  
cp: cannot open 'features' for reading: Permission denied  
[rsmaljkov@rsmaljkov ~]$
```

Рис. 12: Screenshot_18

```
[root@rsmaljkov rsmaljkov]# chmod u-x play  
[root@rsmaljkov rsmaljkov]#
```

Рис. 13: Screenshot_19

```
[rsmaljkov@rsmaljkov ~]$ cd play  
bash: cd: play: Permission denied  
[rsmaljkov@rsmaljkov ~]$
```

Рис. 14: Screenshot_9

```
[root@rsmaljkov rsmaljkov]# ls -l features
--wxrwxrwx. 1 rsmaljkov rsmaljkov 0 May  4 12:33 features
[root@rsmaljkov rsmaljkov]# chmod u+r features
[root@rsmaljkov rsmaljkov]# ls -l features
-rwxrwxrwx. 1 rsmaljkov rsmaljkov 0 May  4 12:33 features
```

Рис. 15: Screenshot_10

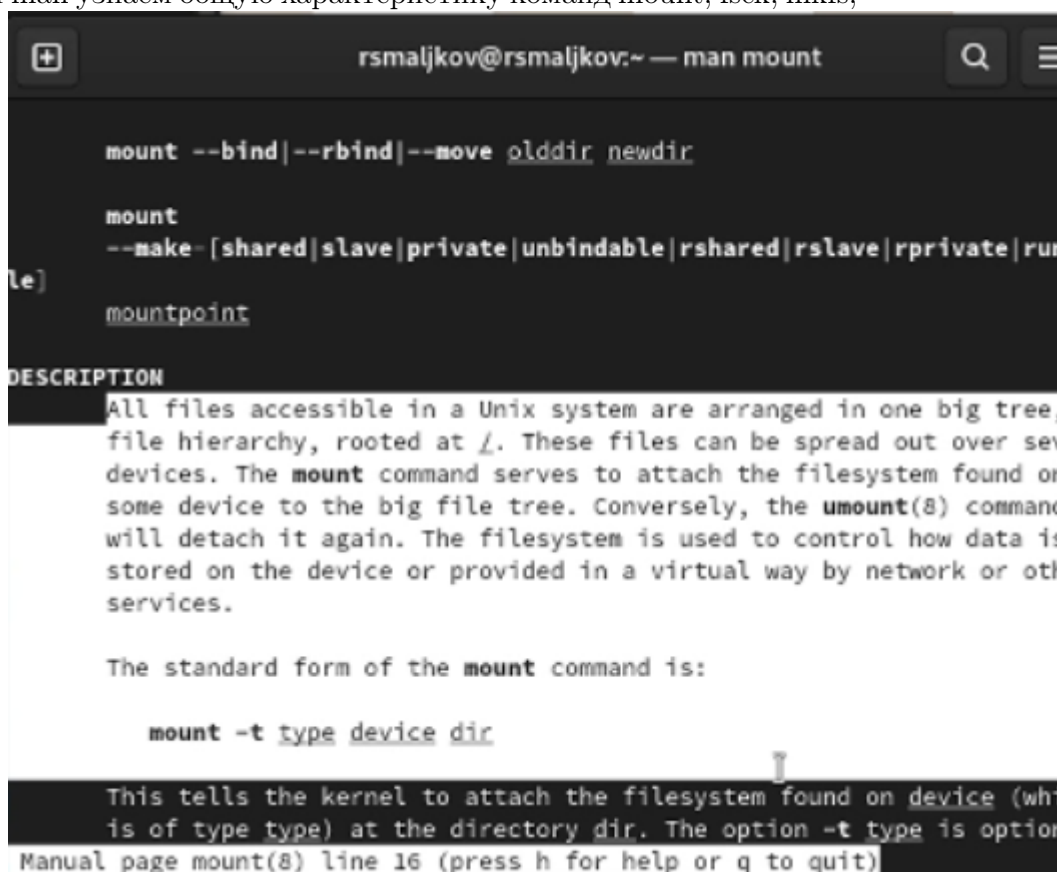
```
[root@rsmaljkov rsmaljkov]# chmod u+x play
[root@rsmaljkov rsmaljkov]#
```

Рис. 16: Screenshot_11

(Скриншоты 9-19)

5

При помощи команды `man` узнаем общую характеристику команд `mount`, `fsck`, `mkfs`,



```

+ rsmaljkov@rsmaljkov:~ — man mount

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

mount
--make=[shared|slave|private|unbindable|rshared|rslave|rprivate|run...
le]
mountpoint

DESCRIPTION
All files accessible in a Unix system are arranged in one big tree,
file hierarchy, rooted at /. These files can be spread out over sev
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 oth
services.

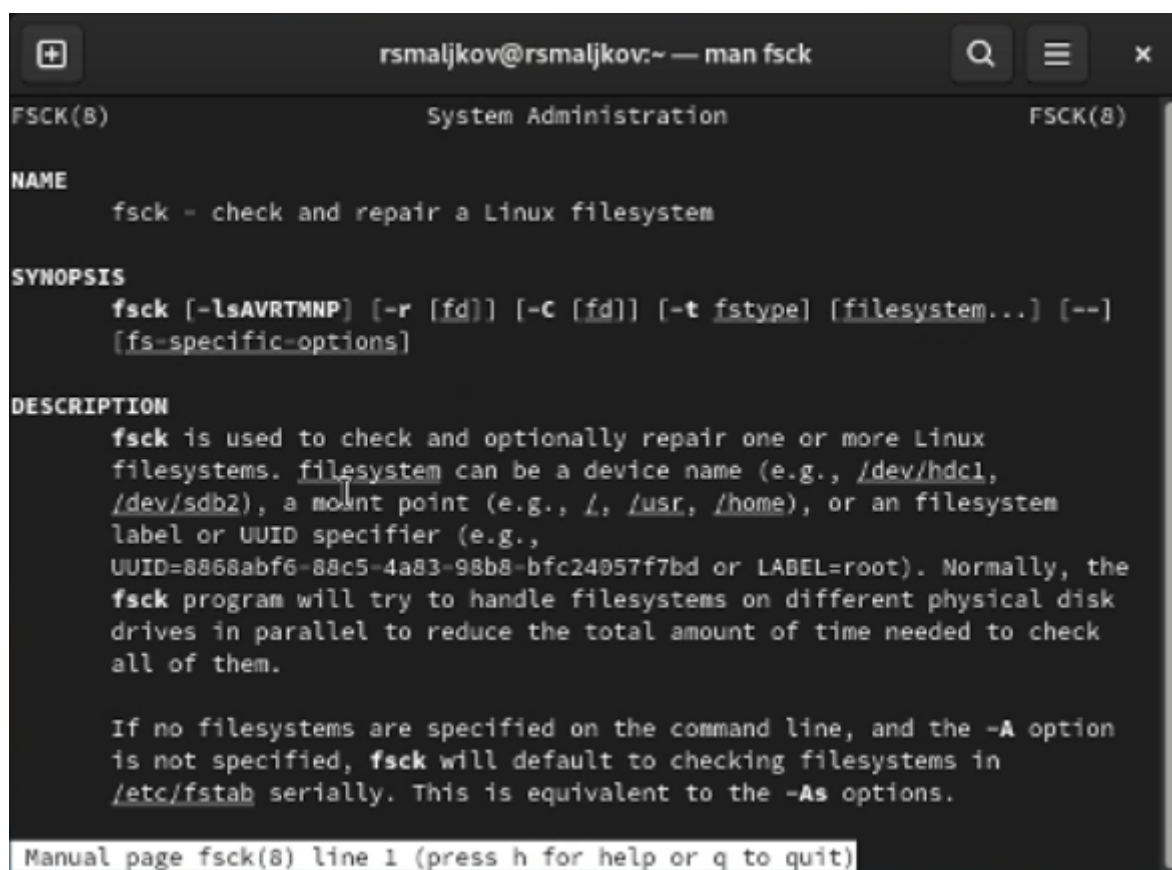
The standard form of the mount command is:

mount -t type device dir

This tells the kernel to attach the filesystem found on device (wh
is of type type) at the directory dir. The option -t type is option
Manual page mount(8) line 16 (press h for help or q to quit)

```

kill(Скриншоты 20 - 24).



```
rsmaljkov@rsmaljkov:~ — man fsck
FCK(8)                               System Administration                               FCK(8)

NAME
    fsck - check and repair a Linux filesystem

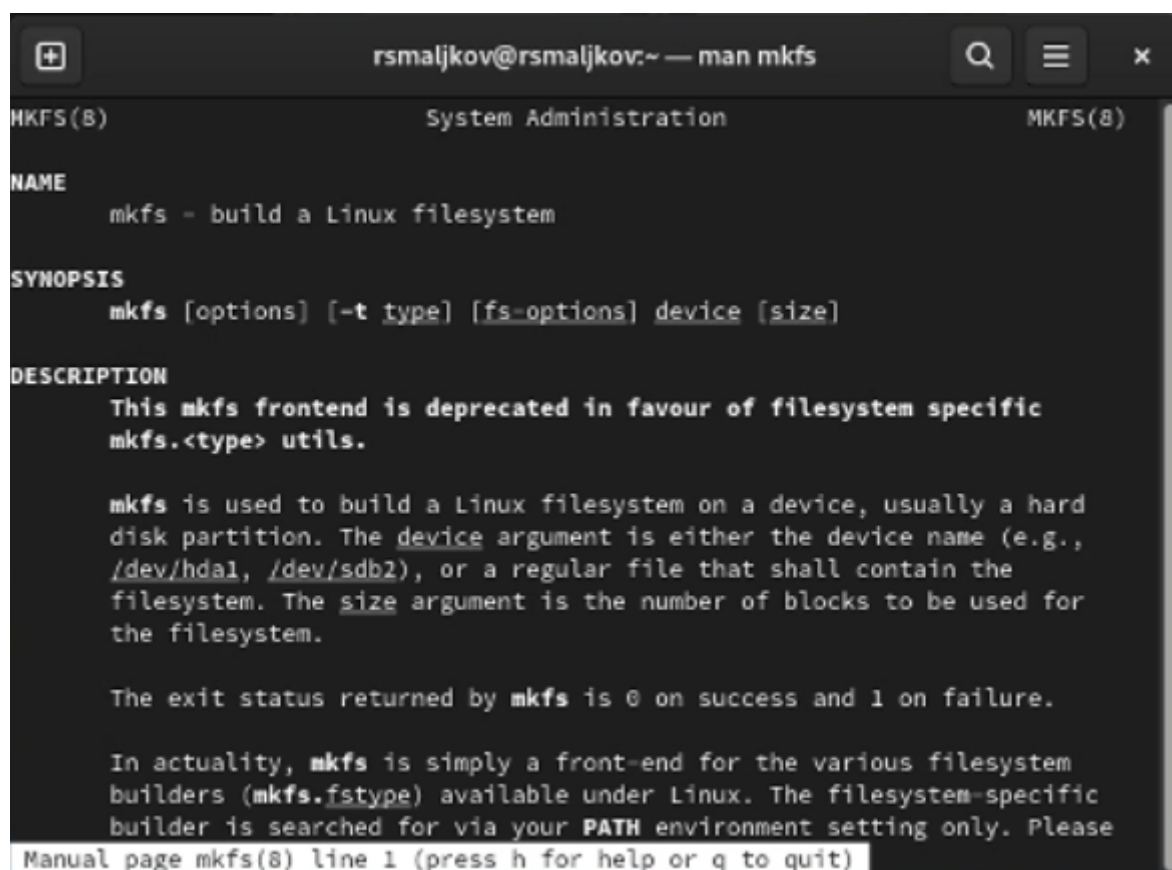
SYNOPSIS
    fsck [-lsAVRTMNP] [-r [fd]] [-C [fd]] [-t fstype] [filesystem...] [--]
    [fs-specific-options]

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 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.

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

Рис. 17: Screenshot_21



The screenshot shows a terminal window with a dark background. The title bar at the top reads "rsmaljkov@rsmaljkov:~ — man mkfs". The terminal content displays the manual page for the `mkfs` command. It includes sections for NAME, SYNOPSIS, and DESCRIPTION. The DESCRIPTION section notes that the `mkfs` frontend is deprecated and provides details on its usage and exit status. At the bottom, a status bar indicates the current position in the manual page.

```

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.

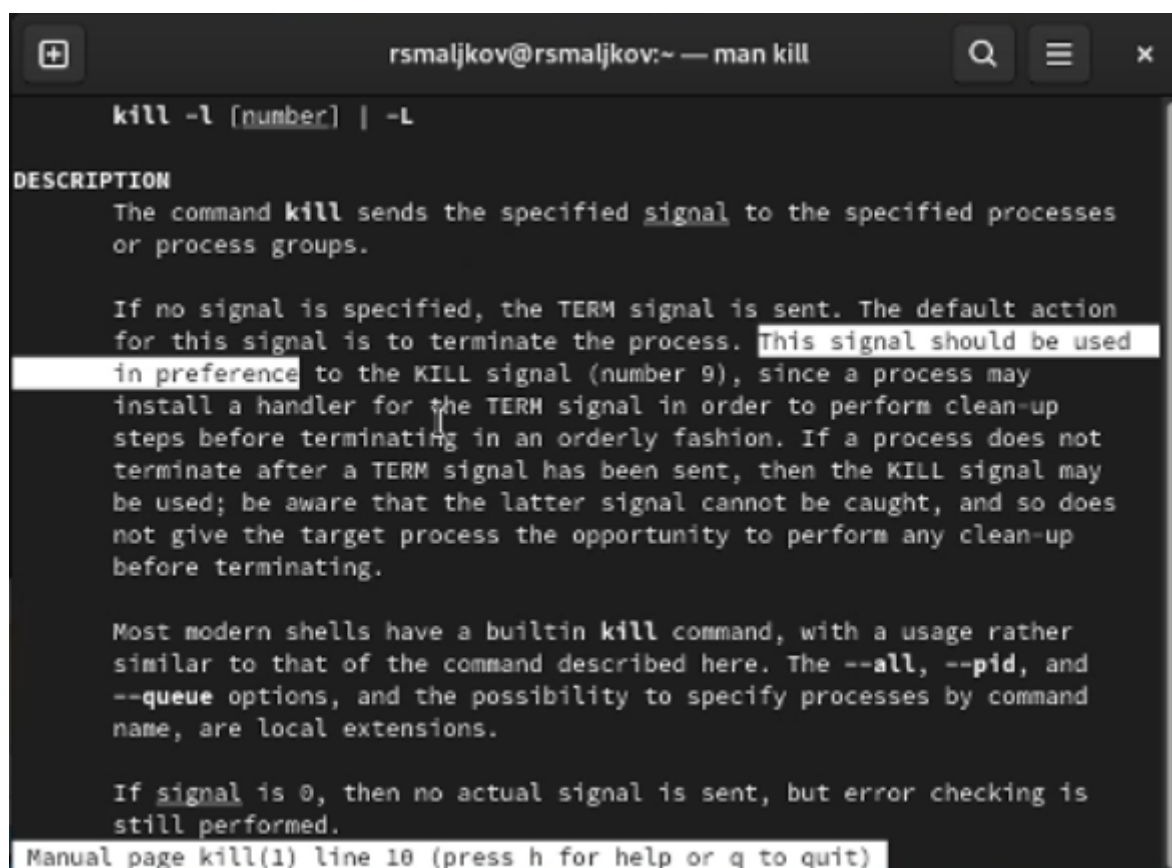
    mkfs is used to build a Linux filesystem on a device, usually a hard
    disk partition. The device argument is either the device name (e.g.,
    /dev/hda1, /dev/sdb2), or a regular file that shall contain the
    filesystem. The size argument is the number of blocks to be used for
    the filesystem.

    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

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

Рис. 18: Screenshot_22



```
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.

Manual page kill(1) line 10 (press h for help or q to quit)

Рис. 19: Screenshot_23

(Скриншоты 20-24)

Выводы

Мы ознакомились 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.