

Отчёт по лабораторной работе №5

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

Гибшер Кирилл Владимирович

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	9
3.1	Команды для работы с файлами и каталогами.	9
3.2	Копирование файлов и каталогов	10
3.3	Перемещение и переименование файлов и каталогов	10
3.4	Права доступа	10
3.5	Изменение прав доступа	10
3.6	Анализ файловой системы	11
4	Выполнение лабораторной работы	12
5	Выводы	19
	Список литературы	20

Список иллюстраций

4.1	Команда <code>cp</code>	12
4.2	Команда <code>mv</code>	12
4.3	Команда <code>chmod</code>	13
4.4	Команда <code>chmod 1.1</code>	13
4.5	Команда <code>chmod 1.2</code>	13
4.6	Изменение имени файла <code>io.h</code>	13
4.7	Создание каталога, перемещение файла в каталог и изменение его имени	14
4.8	Создание файла <code>abc</code> , перемещение и изменение имени.	14
4.9	Создание каталога и перемещение в него файлов	14
4.10	Создание каталога, изменение имени и перемещение	14
4.11	Команда <code>chmod</code>	15
4.12	Команда <code>chmod</code>	15
4.13	Содержимое <code>/etc/passwd</code>	15
4.14	Копирование и перемещение файла. Копирование каталога в другой.	16
4.15	Перемещение и изменение имени каталога	16
4.16	<code>chmod</code> и <code>cat</code>	16
4.17	17
4.18	Информация о команде <code>mount</code>	17
4.19	Информация о команде <code>fsck</code>	17
4.20	Информация о команде <code>mkfs</code>	18
4.21	Информация о команде <code>kill</code>	18

Список таблиц

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. 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. Прodelайте 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 `~/fun` 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 man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, приведя примеры.

3 Теоретическое введение

3.1 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.

3.2 Копирование файлов и каталогов

- Команда `cp` используется для копирования файлов и каталогов. Формат команды: `cp -опции исходный_файл целевой_файл`

3.3 Перемещение и переименование файлов и каталогов

- Команды `mv` и `mvdir` предназначены для перемещения и переименования файлов и каталогов. Формат команды `mv`: `mv -опции старый_файл новый_файл`

3.4 Права доступа

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).

3.5 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`

3.6 Анализ файловой системы

Файловая система в Linux состоит из файлов 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.

4 Выполнение лабораторной работы

Выполним все примеры, приведённые в первой части описания лабораторной работы. (рис. [4.1])

[illegible]

Рис. 4.1: Команда ср

[kvgsbaser@kvgsbaser ~]\$ mv april july							
[kvgsbaser@kvgsbaser ~]\$ ls							
abc1 july monthly	password-2.19.2	work	документы	наблюдения	облачные ресурсы	наблюдения	
abc1 may monthly.00	may monthly.00		Зарплаты	Налогов	"Федерный счет"		
[kvgsbaser@kvgsbaser ~]\$ mv july monthly.00	password-2.19.2-11my-ssm4-4a.ge						
[kvgsbaser@kvgsbaser ~]\$ ls monthly.00							
july monthly							
[kvgsbaser@kvgsbaser ~]\$ mv monthly.00 monthly.01							
[kvgsbaser@kvgsbaser ~]\$ ls							
abc1 may monthly	password-2.19.2-11my-ssm4-4a.ge	work	документы	Зарплаты	Налогов	"Федерный счет"	
abc1 monthly	password-2.19.2	work		наблюдения	облачные ресурсы	наблюдения	
[kvgsbaser@kvgsbaser ~]\$ mv01 reports							
[kvgsbaser@kvgsbaser ~]\$ mv monthly.01 reports							
[kvgsbaser@kvgsbaser ~]\$ mv reports							
[kvgsbaser@kvgsbaser ~]\$ mv reports/monthly.01 reports/monthly							
[kvgsbaser@kvgsbaser ~]\$ ls reports							

Рис. 4.2: Команда mv

```

kvgibsher@kvgibsher:~$ touch may
kvgibsher@kvgibsher:~$ ls -l may
-rw-r--r-- 1 kvgibsher kvgibsher 0 map  5 18:21 may
kvgibsher@kvgibsher:~$ chmod u+x may
kvgibsher@kvgibsher:~$ ls -l may
-rwxr--r-- 1 kvgibsher kvgibsher 0 map  5 18:21 may
kvgibsher@kvgibsher:~$ cd
kvgibsher@kvgibsher:~$ mkdir monthly
kvgibsher@kvgibsher:~$ ls
abc1  may
kvgibsher@kvgibsher:~$ ls
abc1  may

```

Рис. 4.3: Команда chmod

```

kvgibsher@kvgibsher:~$ chmod g-r,g-r monthly
kvgibsher@kvgibsher:~$ ls -l
total 1072
-rw-r--r-- 1 kvgibsher kvgibsher  0 map  5 18:09 abc1
drwxr-xr-x 1 kvgibsher kvgibsher  8 feb 23 16:56 bin
-rw-r--r-- 1 kvgibsher kvgibsher  0 map  5 18:21 may
drwxr-xr-x 1 kvgibsher kvgibsher 24 map  5 18:14 monthly
drwxr-xr-x 1 kvgibsher kvgibsher 16 mar 22 2022 pandoc-2.19.1
-rw-r--r-- 1 kvgibsher kvgibsher 17174200 mar 22 2022 pandoc-2.19.2-linux-amd64.tar.gz
drwxr-xr-x 1 kvgibsher kvgibsher 14 map  5 18:20 reports
drwxr-xr-x 1 kvgibsher kvgibsher 56 feb 23 17:52 work
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 documents
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 documents
drwxr-xr-x 1 kvgibsher kvgibsher 340 feb 23 16:52 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher 50 feb 23 15:23 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa

```

Рис. 4.4: Команда chmod 1.1

```

kvgibsher@kvgibsher:~$ touch abc1
kvgibsher@kvgibsher:~$ chmod g-w abc1
kvgibsher@kvgibsher:~$ ls -l
total 1072
-rw-r--r-- 1 kvgibsher kvgibsher  0 map  5 18:28 abc1
drwxr-xr-x 1 kvgibsher kvgibsher  8 feb 23 16:56 bin
-rw-r--r-- 1 kvgibsher kvgibsher  0 map  5 18:21 may
drwxr-xr-x 1 kvgibsher kvgibsher 24 map  5 18:14 monthly
drwxr-xr-x 1 kvgibsher kvgibsher 16 mar 22 2022 pandoc-2.19.1
-rw-r--r-- 1 kvgibsher kvgibsher 17174200 mar 22 2022 pandoc-2.19.2-linux-amd64.tar.gz
drwxr-xr-x 1 kvgibsher kvgibsher 14 map  5 18:20 reports
drwxr-xr-x 1 kvgibsher kvgibsher 56 feb 23 17:52 work
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 documents
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 documents
drwxr-xr-x 1 kvgibsher kvgibsher 340 feb 23 16:52 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher 50 feb 23 15:23 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa
drwxr-xr-x 1 kvgibsher kvgibsher  0 feb  9 21:21 karyussa

```

Рис. 4.5: Команда chmod 1.2

Скопируем файл /usr/include/sys/io.h в домашний каталог и назовем его equipment.(рис. [4.6]).

```

kvgibsher@kvgibsher:~$ cd /usr/include/sys
kvgibsher@kvgibsher:~/usr/include/sys$ ls
acct.h  elf.h  file.h  io.h  mch.h  pldfd.h  ptrace.h  reg.h  sh.h  soundcard.h  sysinfo.h  timerfd.h  ucontext.h  vfs.h
auxv.h  epoll.h  fault.h  ipc.h  mtd.h  poll.h  quota.h  rseq.h  signal.h  statfs.h  syslog.h  times.h  uio.h  vlmitt.h
bittypes.h  errno.h  gmon.h  kd.h  param.h  poll.h  quota.h  rseq.h  signal.h  statfs.h  syslog.h  times.h  uio.h  vlmitt.h
defs.h  eventfd.h  gmon.out.h  klogd.h  pci.h  procfs.h  raw.h  sen.h  socket.h  statfs.h  syslog.h  times.h  uio.h  vlmitt.h
debugreg.h  fanotify.h  fanotify.h  man.h  para.h  procfs.h  raw.h  sen.h  socket.h  statfs.h  syslog.h  times.h  uio.h  vlmitt.h
dir.h  fcntl.h  fcntl.h  mount.h  personality.h  profil.h  reboot.h  sendfile.h  socketvar.h  syscall.h  time.h  types.h  utname.h  xattr.h
kvgibsher@kvgibsher:~/usr/include/sys$ cp io.h ./
kvgibsher@kvgibsher:~/usr/include/sys$ ls
abc1  io.h
kvgibsher@kvgibsher:~/usr/include/sys$ cd
kvgibsher@kvgibsher:~$ mv io.h equipment
kvgibsher@kvgibsher:~$ ls
abc1  equipment
kvgibsher@kvgibsher:~$ ls
abc1  equipment

```

Рис. 4.6: Изменение имени файла io.h

В домашнем каталоге создадим директорию ~/ski.plases. Переместим файл equipment в каталог ~/ski.plases.Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist.(рис. [4.7]).

```

[kvgibsherkvgibsh@ ~]$ mkdir ski_places
[kvgibsherkvgibsh@ ~]$ ls
abc1  equipment  monthly  pentos-4.19.3-linux-amd64.tar.gz  ski_places  reports  work
[kvgibsherkvgibsh@ ~]$ mv equipment ski_places
[kvgibsherkvgibsh@ ~]$ ls ski_places
equipment
[kvgibsherkvgibsh@ ~]$ mv ski_places/equipment ski_places/equiplist
[kvgibsherkvgibsh@ ~]$ ls ski_places
equiplist
[kvgibsherkvgibsh@ ~]$

```

Рис. 4.7: Создание каталога, перемещение файла в каталог и изменение его имени

Создадим в домашнем каталоге файл abc1 и скопируем его в каталог ~/ski_places, назовем его equiplist2. Создадим каталог с именем equipment в каталоге ~/ski_places. Переместим файлы ~/ski_places/equiplist и equiplist2 в каталог ~/ski_places/equipment.(рис. [4.8]).

```

[kvgibsherkvgibsh@ ~]$ cp abc1 ski_places
[kvgibsherkvgibsh@ ~]$ ls ski_places
abc1  equiplist
[kvgibsherkvgibsh@ ~]$ mv abc1 equiplist2
[kvgibsherkvgibsh@ ~]$ ls ski_places
abc1  equiplist2
[kvgibsherkvgibsh@ ~]$ mv ski_places/abc1 ski_places/equiplist2
[kvgibsherkvgibsh@ ~]$ ls ski_places
equiplist  equiplist2
[kvgibsherkvgibsh@ ~]$

```

Рис. 4.8: Создание файла abc , перемещение и изменение имени.

```

[kvgibsherkvgibsh@ ~]$ mkdir ski_places/equipment
[kvgibsherkvgibsh@ ~]$ ls ski_places
abc1  equiplist2
[kvgibsherkvgibsh@ ~]$ mv ski_places/equiplist ski_places/equipment
[kvgibsherkvgibsh@ ~]$ ls
abc1  equiplist2
[kvgibsherkvgibsh@ ~]$ mv equiplist2 ski_places
[kvgibsherkvgibsh@ ~]$ mv ski_places/equiplist2 ski_places/equipment
[kvgibsherkvgibsh@ ~]$ ls ski_places
equiplist  equiplist2
[kvgibsherkvgibsh@ ~]$ ls ski_places/equipment
equiplist  equiplist2
[kvgibsherkvgibsh@ ~]$

```

Рис. 4.9: Создание каталога и перемещение в него файлов

Создадим и переместим каталог ~/newdir в каталог ~/ski_places и назовите его plans (рис. [4.10]).

```

[kvgibsherkvgibsh@ ~]$ mkdir newdir
[kvgibsherkvgibsh@ ~]$ mv newdir ski_places
[kvgibsherkvgibsh@ ~]$ ls ski_places
equiplist2  newdir
[kvgibsherkvgibsh@ ~]$ mv newdir plans
ls: не удалось получить стат. дан. 'newdir': Нет такого файла или каталога
[kvgibsherkvgibsh@ ~]$ mv ski_places/newdir ski_places/plans
[kvgibsherkvgibsh@ ~]$ ls ski_places
equiplist2  plans

```

Рис. 4.10: Создание каталога, изменение имени и перемещение

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 (рис. [4.11]).

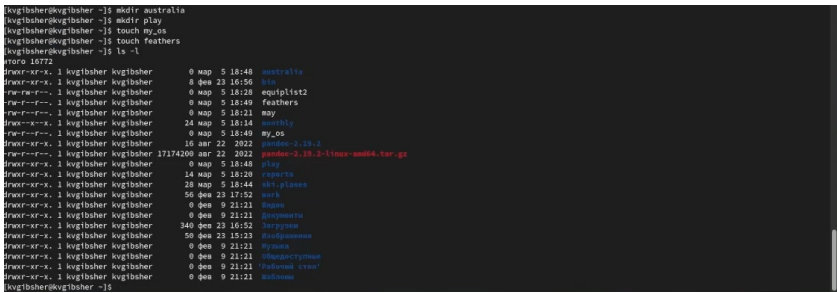


Рис. 4.11: Команда chmod

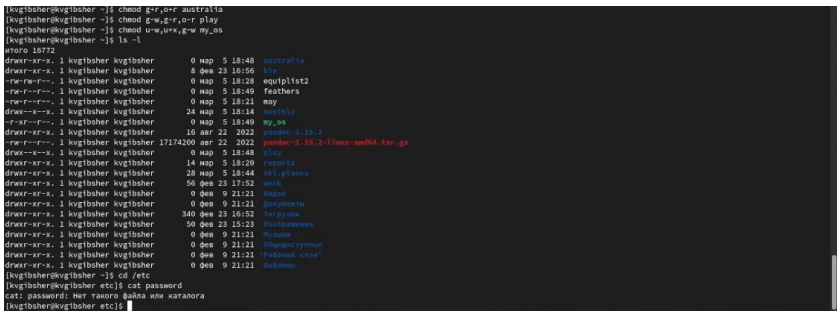


Рис. 4.12: Команда `chmod`

Просмотрим содержимое файла `/etc/passwd` (рис. [4.13]).

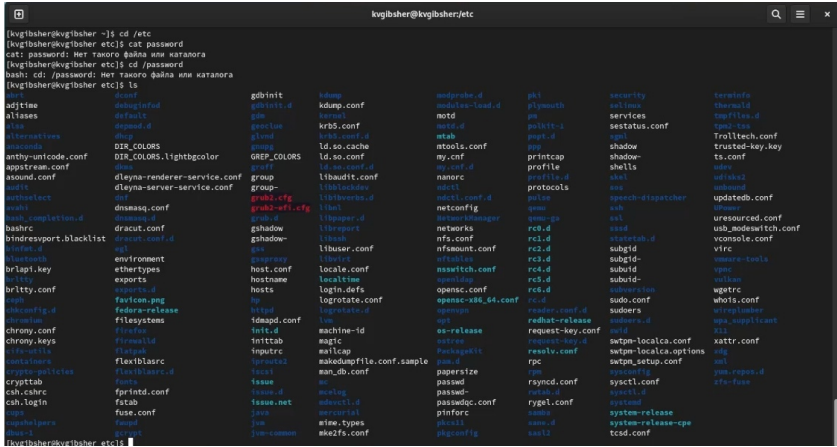


Рис. 4.13: Содержимое /etc/passwd

Скопируем файл ~/feathers в файл ~/file.old. Переместим файл ~/file.old в каталог ~/play. Скопируем каталог ~/play в каталог ~/fun. (рис. [4.14]).

```
[kvgibsherkvgib@shell ~]$ cd
[kvgibsherkvgib@shell ~]$ cp feathers file.old
cp: не удается создать файл 'file.old': Нет такого файла или каталога
[kvgibsherkvgib@shell ~]$ cp feathers file.old
[kvgibsherkvgib@shell ~]$ ls
bin          file.old    monthly    pandas-2.19.2  play        ski-places  maps        documents    images    kaggle-datasets  'Personal chat'  datasets
conda-envs  equiplist2  my        pandas-2.19.2-linux-64bit.tar.gz  reports     work
[kvgibsherkvgib@shell ~]$ mv file.old play
[kvgibsherkvgib@shell ~]$ ls play
[kvgibsherkvgib@shell ~]$ cp play fun
cp: не удается создать файл 'fun': Нет такого файла или каталога
[kvgibsherkvgib@shell ~]$ cp -r play fun
[kvgibsherkvgib@shell ~]$ ls fun
[kvgibsherkvgib@shell ~]$ cd fun
[kvgibsherkvgib@fun ~]$ cp -r play fun
[kvgibsherkvgib@fun ~]$ ls fun
[kvgibsherkvgib@fun ~]$
```

Рис. 4.14: Копирование и перемещение файла. Копирование каталога в другой.

Переместим каталог ~/fun в каталог ~/play и назовите его games. (рис. [4.15]).

```
[kvgibsherkvgib@shell ~]$ cd
[kvgibsherkvgib@shell ~]$ mv fun play/games
[kvgibsherkvgib@shell ~]$
```

Рис. 4.15: Перемещение и изменение имени каталога

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. (рис. [4.16]).

```
[kvgibsherkvgib@shell ~]$ cd
[kvgibsherkvgib@shell ~]$ ls feathers
[kvgibsherkvgib@shell ~]$ cat feathers
cat: feathers: Permission denied
[kvgibsherkvgib@shell ~]$ cp feathers play
cp: feathers: Permission denied
[kvgibsherkvgib@shell ~]$ ls play
play
[kvgibsherkvgib@shell ~]$ cd play
[kvgibsherkvgib@play ~]$ ls -l
total 1072
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 0 map 5 18:48 australia
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 8 feb 23 16:56 bin
-rw-r--r--. 1 kvgibsherkvgibsherkvgib 0 map 5 18:28 equiplist2
-rw-r--r--. 1 kvgibsherkvgibsherkvgib 0 map 5 18:49 feathers
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 24 map 5 18:56 fun
-rw-r--r--. 1 kvgibsherkvgibsherkvgib 0 map 5 18:21 my
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 24 map 5 18:14 monthly
-rw-r--r--. 1 kvgibsherkvgibsherkvgib 0 map 5 18:46 my_os
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 16 apr 22 2022 pandas-2.19.2
-rw-r--r--. 1 kvgibsherkvgibsherkvgib 17174200 apr 22 2022 pandas-2.19.2-linux-64bit.tar.gz
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 42 map 5 19:05 play
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 14 map 5 18:29 reports
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 28 map 5 18:44 ski-places
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 56 feb 23 17:22 work
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 0 feb 9 21:21 maps
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 0 feb 9 21:21 documents
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 240 feb 23 16:52 images
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 50 feb 23 15:23 kaggle-datasets
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 0 feb 9 21:21 'Personal chat'
drwxr-xr-x. 1 kvgibsherkvgibsherkvgib 0 feb 9 21:21 datasets
[kvgibsherkvgib@play ~]$
```

Рис. 4.16: chmod и cat

Лишите владельца каталога ~/play права на выполнение. Перейдите в каталог ~/play. Что произошло? Дайте владельцу каталога ~/play право на выполнение. (рис. [4.17]).


```
kvgibshern@kvgibshern ~$ cd play
bash: cd: play: Ошибка: не существует
kvgibshern@kvgibshern ~$ chmod u+x plya
chmod: невозможно получить доступ к 'plya': Нет такого файла или каталога
kvgibshern@kvgibshern ~$ chmod u+x play
kvgibshern@kvgibshern ~$ cd play
```

Рис. 4.17:

Прочитаем информацию о командах `mount`, `mkfs`, `fsck`, `kill` с помощью `man` (рис. [4.18]).

```

kuglsaher@kuglsaher:~$ man mount
System Administration                                MOUNT(8)

MOUNT(8)
mount - mount a filesystem

SYNOPSIS
mount [-h -V]
mount [-L] [-t fstype]
mount -o [-ffstype] [-t fstype] [-o options]
mount [-ffstype] [-o options] devicemountpoint
mount [-ffstype] [-t fstype] [-o options] device mountpoint
mount --bind [--bind=]--move olddir newdir
mount --make-shared|slave|private|unbindable|shared|slave|private|rbindable 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:

```

Рис. 4.18: Информация о команде mount

```

kvgibsher@kvgibsher:~$ play --man fsck
System Administration
FSCK(B)

NAME
    fsck - check and repair a Linux filesystem

SYNOPSIS
    fsck [-laAVRTnmp] [-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/sdb1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=56da6a9f-b0c5-4a33-b0a8-b7c2d0377fad or UUID=kernel). 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

```

Рис. 4.19: Информация о команде fsck

```
kugibsher@kugibsher:~$ play --man mkfs
mkfs(8)
System Administration
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.
[Manual page mkfs(8): line 1: (search for help or a fs utility)]
```

Рис. 4.20: Информация о команде mkfs

```
kugibsher@kugibsher:~$ play --man kill
kill(1)
User Commands
NAME
    kill - terminate a process
SYNOPSIS
    kill [-signal|--signal=-p] [-s value] [-n] [--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,
[Manual page kill(1): line 1: (press h for help or q to quit)]
```

Рис. 4.21: Информация о команде kill

5 Выводы

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

Список литературы

1. Лабораторная работа №5 [Электронный ресурс] - Режим доступа:https://esystem.rudn.ru/plab_files.pdf