

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Операционные системы

Студент: Губина Ольга Вячеславовна Группа: НПИбд-01-20

Преподаватель: Велиева Татьяна Рефатовна

МОСКВА

2021 г.

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.

Теоретические сведения:

Поскольку мы собираемся работать с файловыми системами, нам необходимо ознакомиться с командами для работы с файлами и каталогами.

Команда создания текстового файла *touch*:

Синтаксис: `touch <имя файла> .`

Команды просмотра текстовых файлов:

- для просмотра небольших файлов - **cat**.

Синтаксис: `cat <имя файла> .`

- для просмотра больших файлов - **less**.

Позволяет осуществлять постраничный просмотр файлов (длина страницы соответствует размеру экрана)

Синтаксис: `less <имя файла> .`

- для просмотра начала файла **head**.

По умолчанию она выводит первые 10 строк файла.

Синтаксис: `head [-n] <имя файла> , где n — количество выводимых строк.`

- команда **tail** выводит несколько (по умолчанию 10) последних строк файла.

Синтаксис: `tail [-n] <имя файла> , где n — количество выводимых строк.`

Команда копирования файлов и каталогов *cp*:

```
cp [-опции] <исходный_файл> <целевой_файл>
```

Команды для перемещения и переименования файлов и каталогов *mv* и *mkdir*:

```
mv [-опции] <старый_файл> <новый_файл>
```

Выполнение работы:

1. Выполняем все примеры, приведенные в материалах лабораторной работы №6^[1].

- **Копирование файлов и каталогов:**

Примеры:

1. Копирование файла в текущем каталоге. Скопируем файл ~/abc1 в файл april и в файл may, предварительно создаем файл abc1 командой touch (рисунок 1). Само копирование осуществляется командой cp :

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

Рисунок 1: копирование файла в текущем каталоге:

```
[ovgubina@localhost ~]$ cd
[ovgubina@localhost ~]$ touch abc1
[ovgubina@localhost ~]$ cp abc1 april
[ovgubina@localhost ~]$ cp abc1 may
[ovgubina@localhost ~]$ ls
abc1  bin  work  Документы  Изображения  Общедоступные  Шаблоны
april  may  Видео  Загрузки  Музыка      Рабочий стол
[ovgubina@localhost ~]$
```

Проверяем правильность действий командой ls - просматриваем содержимое домашнего каталога. Видим, что все новые файлы на месте.

2. Копирование нескольких файлов в каталог. Скопируем файлы april и may в каталог monthly командой cp (рисунок 2) (предварительно создали каталог monthly командой mkdir):

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

Рисунок 2: копирование нескольких файлов в каталог:

```
[ovgubina@localhost ~]$ mkdir monthly
[ovgubina@localhost ~]$ cp april may monthly
[ovgubina@localhost ~]$ ls ~/monthly
april  may
[ovgubina@localhost ~]$
```

Командой ls проверяем содержимое каталога monthly. Видим, что файлы april и may скопировались в наш каталог.

3. Копирование файлов в произвольном каталоге. Скопируем файл monthly/may в файл с именем june командой cp и проверим верность наших действий, просмотрев каталог monthly командой ls (рисунок 3):

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

Рисунок 3: копирование файлов в произвольном каталоге:

```
[ovgubina@localhost ~]$ cp monthly/may monthly/june
[ovgubina@localhost ~]$ ls monthly
april  june  may
[ovgubina@localhost ~]$
```

Видим, что скопированный файл may отображается в каталоге как june, как и задумывалось.

4. Копирование каталогов в текущем каталоге. Скопируем каталог monthly в каталог monthly.00. Для этого воспользуемся командой cp -r , которая позволяет копировать каталоги вместе с входящими в них файлами и каталогами (рисунок 4), но сначала создадим каталог monthly.00 командой mkdir:

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

Рисунок 4: копирование каталогов в текущем каталоге:

```
[ovgubina@localhost ~]$ mkdir monthly.00
[ovgubina@localhost ~]$ cp -r monthly monthly.00
[ovgubina@localhost ~]$ ls monthly.00
monthly
[ovgubina@localhost ~]$
```

Проверим верность действий командой ls - посмотрим содержимое каталога monthly.00. Видим, что каталог monthly был успешно скопирован в каталог monthly.00.

5. Копирование каталогов в произвольном каталоге. Скопируем каталог monthly.00 в каталог /tmp все той же командой cp -r (рисунок 5):

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

Рисунок 5: копирование каталогов в произвольном каталоге:

```
[ovgubina@localhost ~]$ cp -r monthly.00 /tmp
[ovgubina@localhost ~]$ ls /tmp
anaconda.log
hsperfdata_root
ifcfg.log
ks-script-j9jNjt
monthly.00
packaging.log
program.log
sensitive-info.log
```

Просмотрим содержимое каталога /tmp (команда ls). Видим, что каталог monthly.00 теперь действительно находится в нем (он подсвечен синим).

- **Перемещение и переименование файлов и каталогов:**

1. Переименование файлов в текущем каталоге. Изменим название файла april на july в домашнем каталоге. Воспользуемся командой `mv` (рисунок 6):

```
cd
mv april july
```

Рисунок 6: переименование файлов в текущем каталоге:

```
[ovgubina@localhost ~]$ cd
[ovgubina@localhost ~]$ mv april july
[ovgubina@localhost ~]$ ls
abcl  july  monthly  work  Документы  Изображения  Общедоступные  Шаблоны
bin   may   monthly.00  Видео  Загрузки  Музыка  Рабочий стол
[ovgubina@localhost ~]$
```

Просматриваем содержимое каталога и видим, что файл был успешно переименован (выделено черным).

2. Перемещение файлов в другой каталог. Переместим файл july в каталог monthly.00, вновь воспользовавшись командой `mv` (рисунок 7):

```
mv july monthly.00
ls monthly.00
```

Рисунок 7: перемещение файлов в другой каталог:

```
[ovgubina@localhost ~]$ mv july monthly.00
[ovgubina@localhost ~]$ ls monthly.00
july  monthly
[ovgubina@localhost ~]$
```

Смотрим содержимое каталога monthly.00 и убеждаемся в том, что файл july был успешно перемещен в него.

3. Переименование каталогов в текущем каталоге. Переименуем каталог monthly.00 в monthly.01 командой `mv` (рисунок 8):

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

Рисунок 8: переименование каталогов в текущем каталоге:

```
[ovgubina@localhost ~]$ mv monthly.00 monthly.01
[ovgubina@localhost ~]$ ls
abcl  may  monthly.01  Видео  Загрузки  Музыка  Рабочий стол
bin   monthly  work  Документы  Изображения  Общедоступные  Шаблоны
[ovgubina@localhost ~]$
```

Проверим содержимое домашнего каталога. Видим, что каталог monthly.00 был переименован в monthly.01 (подсвечено синим).

4. Перемещение каталога в другой каталог. Переместим каталог monthly.01 в каталог reports командой `mv`, предварительно создадим каталог reports командой `mkdir` (рисунок 9):

```
mkdir reports
mv monthly.01 reports
```

Рисунок 9: перемещение каталога в другой каталог:

```
[ovgubina@localhost ~]$ mkdir reports
[ovgubina@localhost ~]$ mv monthly.01 reports
[ovgubina@localhost ~]$ ls reports
monthly.01
[ovgubina@localhost ~]$
```

Просматриваем каталог reports и видим, что каталог monthly.01 был успешно перемещен в него.

5. Переименование каталога, не являющегося текущим. Переименуем каталог reports/monthly.01 в reports/monthly (рисунки 10):

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

Рисунок 10: переименование каталога, не являющегося текущим:

```
[ovgubina@localhost ~]$ mv reports/monthly.01 reports/monthly
[ovgubina@localhost ~]$ ls reports
monthly
[ovgubina@localhost ~]$
```

Просматриваем каталог reports и видим, что каталог monthly.01 был успешно переименован в monthly.

- **Изменение прав доступа:**

1. Создадим файл ~/may (команда touch) с правом выполнения для владельца (меняем права доступа командой chmod, используем комбинацию u+x) между делом проверяем свойства файла командой ls -l (рисунки 11):

```
cd
touch may
ls -l may
chmod u+x may
ls -l may
```

Рисунок 11: файл ~/may с правом выполнения для владельца:

```
[ovgubina@localhost ~]$ cd
[ovgubina@localhost ~]$ touch may
[ovgubina@localhost ~]$ ls -l may
-rw-rw-r--. 1 ovgubina ovgubina 0 май  4 22:49 may
[ovgubina@localhost ~]$ chmod u+x may
[ovgubina@localhost ~]$ ls -l may
-rwxrw-r--. 1 ovgubina ovgubina 0 май  4 22:49 may
[ovgubina@localhost ~]$
```

По выводам команды ls понимаем, что задание выполнено верно.

2. Лишим владельца файла ~/may права на выполнение комбинацией u-x и проверим правильность действий командой ls -l (рисунки 12):

```
chmod u-x may
ls -l may
```

Рисунок 12: лишение владельца права на выполнение:

```
-rwxrw-r--. 1 ovgubina ovgubina 0 май  4 22:49 may
[ovgubina@localhost ~]$ chmod u-x may
[ovgubina@localhost ~]$ ls -l may
-rw-rw-r--. 1 ovgubina ovgubina 0 май  4 22:49 may
[ovgubina@localhost ~]$
```

По выводам команды ls понимаем, что задание выполнено верно.

3. Создадим каталог monthly (командой mkdir) с запретом на чтение для членов группы и всех остальных пользователей (g-r, o-r) (рисунки 13):

```
cd
mkdir monthly
chmod g-r monthly
chmod o-r monthly
```

Рисунок 13: каталог monthly с запретом на чтение для членов группы и всех остальных пользователей:

```
[ovgubina@localhost ~]$ cd
[ovgubina@localhost ~]$ mkdir monthly
mkdir: невозможно создать каталог «monthly»: Файл существует
[ovgubina@localhost ~]$ ls -l monthly
итого 0
-rw-rw-r--. 1 ovgubina ovgubina 0 май  4 22:28 april
-rw-rw-r--. 1 ovgubina ovgubina 0 май  4 22:30 june
-rw-rw-r--. 1 ovgubina ovgubina 0 май  4 22:28 may
[ovgubina@localhost ~]$ chmod g-r, o-r monthly
chmod: неверный режим: «g-r,»
По команде «chmod --help» можно получить дополнительную информацию.
[ovgubina@localhost ~]$ chmod g-r o-r monthly
chmod: невозможно получить доступ к «o-r»: Нет такого файла или каталога
[ovgubina@localhost ~]$ chmod g-r monthly
[ovgubina@localhost ~]$ chmod o-r monthly
```

4. Создадим файл ~/abc1 (команда touch) с правом записи для членов группы (g+w) (рисунок 14):

```
cd
touch abc1
chmod g+w abc1
```

Рисунок 14: файл ~/abc1 с правом записи для членов группы:

```
[ovgubina@localhost ~]$ cd
[ovgubina@localhost ~]$ touch abc1
[ovgubina@localhost ~]$ chmod g+w abc1
```

2. 2.1. Убедимся в том, что искомый файл io.h существует и находится в каталоге /usr/include/sys, для этого воспользуемся командой просмотра содержимого ls. Скопируем файл /usr/include/sys/io.h в домашний каталог (команда cp) и назовем его equipment - переименуем командой mv. (рисунок 15)

Рисунок 15: копирование и переименовывание файла io.h

```
[ovgubina@localhost ~]$ ls /usr/include/sys
acct.h      gmon.h      pci.h        select.h     sysctl.h     uio.h
auxv.h      gmon_out.h  perm.h       sem.h        sysinfo.h    ultrasound.h
bitypes.h   inotify.h   personality.h sendfile.h   syslog.h     un.h
cdefs.h     ioctl.h     poll.h       shm.h        sysmacros.h  unistd.h
debugreg.h  io.h        prctl.h      signalfd.h   termios.h    user.h
dir.h       ipc.h       procfs.h     signal.h     timeb.h      ustat.h
elf.h       kdaemon.h  profil.h     socket.h     time.h       utsname.h
epoll.h     kd.h       ptrace.h     socketvar.h  timerfd.h    vfs.h
errno.h     klog.h     queue.h      soundcard.h  times.h      vlimit.h
eventfd.h   mman.h     quota.h      statfs.h     timex.h      vm86.h
fanotify.h  mount.h    raw.h        stat.h       ttychars.h   vt.h
fcntl.h     msg.h      reboot.h     statvfs.h    ttydefaults.h vtimes.h
file.h      mtio.h     reg.h        swap.h       types.h      wait.h
fsuid.h     param.h    resource.h   syscall.h    ucontext.h   xattr.h
[ovgubina@localhost ~]$ cp /usr/include/sys/io.h /home/ovgubina
[ovgubina@localhost ~]$ ls
bin work Документы Изображения Общедоступные Шаблоны
io.h Видео Загрузки Музыка Рабочий стол
[ovgubina@localhost ~]$ mv io.h equipment
[ovgubina@localhost ~]$ ls
bin work Документы Изображения Общедоступные Шаблоны
equipment Видео Загрузки Музыка Рабочий стол
[ovgubina@localhost ~]$ █
```

Смотрим на промежуточные и конечный выходы команды ls, которую используем в качестве проверки наших действий, видим, что действия выполнены верно, и теперь в домашнем каталоге содержится файл equipment.

- 2.2. В домашнем каталоге создадим директорию ~/ski.places командой mkdir и проверим правильность действий просмотром содержимого домашнего каталога (рисунок 16):

Рисунок 16: создание каталога ~/ski.places:

```
[ovgubina@localhost ~]$ cd
[ovgubina@localhost ~]$ mkdir ski.places
[ovgubina@localhost ~]$ ls
bin ski.places Видео Загрузки Музыка Рабочий стол
equipment work Документы Изображения Общедоступные Шаблоны
[ovgubina@localhost ~]$
```

По результату вывода команды ls видим, что теперь каталог ~/ski.places содержится в домашнем каталоге (подсвечен синим).

2.3. Переместим файл equipment в каталог ~/ski.plases командой mv. Просмотрим содержимое каталога ~/ski.plases командой ls (рисунок 17).

Рисунок 17: перемещение файла equipment:

```
[ovgubina@localhost ~]$ mv equipment ~/ski.plases
[ovgubina@localhost ~]$ ls ~/ski.plases
equipment
[ovgubina@localhost ~]$ █
```

Видим, что файл действительно находится на своем месте.

2.4. Переименуем файл ~/ski.plases/equipment в ~/ski.plases/equiplist командой mv (рисунок 18).

Рисунок 18: переименование файла equipment:

```
[ovgubina@localhost ~]$ mv ~/ski.plases/equipment ~/ski.plases/equiplist
[ovgubina@localhost ~]$ ls ~/ski.plases
equiplist
[ovgubina@localhost ~]$ █
```

Просмотрим содержимое каталога и убедимся в правильности действий - теперь катлог содержит файл equiplist.

2.5. Создаем в домашнем каталоге файл abc1 командой touch и копируем его в каталог ~/ski.plases командой cp, называем его equiplist2 - переименовываем в equiplist2 командой mv (рисунок 19).

Рисунок 19:

```
[ovgubina@localhost ~]$ cd
[ovgubina@localhost ~]$ touch abc1
[ovgubina@localhost ~]$ cp abc1 ~/ski.plases
[ovgubina@localhost ~]$ ls ~/ski.plases
abc1 equiplist
[ovgubina@localhost ~]$ mv ~/ski.plases/abc1 ~/ski.plases/equiplist2
[ovgubina@localhost ~]$ ls ~/ski.plases
equiplist equiplist2
[ovgubina@localhost ~]$
```

Правильность действий проверяем периодическим вызовом команды ls для просмотра содержимого каталога ~/ski.plases. Смотрим на результаты выводов и видим, что мы выполнили все верно и теперь в каталоге содержится 2 разных файла equiplist.

2.6. Создадим каталог с именем equipment в каталоге ~/ski.plases с помощью команды mkdir, для этого сначала перейдем в каталог ~/ski.plases (рисунок 20).

Рисунок 20: создание каталога equipment:

```
[ovgubina@localhost ~]$ cd ~/ski.plases
[ovgubina@localhost ski.plases]$ mkdir equipment
[ovgubina@localhost ski.plases]$ ls
equiplist equiplist2 equipment
[ovgubina@localhost ski.plases]$
```

Проверяем правильность действий просмотром содержимого каталога. Видим, что новый каталог был в него добавлен (выделен синим)

2.7. Переместим файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment с помощью команды mv и проверяем содержимое каталога equipment командой ls, видим, что все верно (рисунок 21).

Рисунок 21: перемещение файлов

```
[ovgubina@localhost ski.plases]$ mv equiplist equiplist2 equipment
[ovgubina@localhost ski.plases]$ ls equipment
equiplist equiplist2
[ovgubina@localhost ski.plases]$ █
```

2.8. Создадим (mkdir) и переместим каталог ~/newdir в каталог ~/ski.plases, используя команду mv, и назовем его plans - переименуем при помощи той же команды mv (рисунок 22). По ходу действий проверяем верность выполнения просмотром содержимого каталога ls.

Рисунок 22:

```
[ovgubina@localhost ski.plases]$ cd
[ovgubina@localhost ~]$ mkdir newdir
[ovgubina@localhost ~]$ mv ~/newdir ~/ski.plases
[ovgubina@localhost ~]$ ls ~/ski.plases
equipment newdir
[ovgubina@localhost ~]$ mv ~/ski.plases/newdir ~/ski.plases/plans
[ovgubina@localhost ~]$ ls ~/ski.plases
equipment plans
[ovgubina@localhost ~]$
```

Смотрим на выводы ls - работа выполнена верно.

3. Определяем опции команды chmod:

3.1. drwxr--r-- ... australia (рисунок 23)

Нам нужно обозначить неограниченные права для владельца каталога и право только на чтение для группы и других пользователей.

Для этого сначала создадим каталог `australia` командой `mkdir` (создаем каталог, поскольку первый знаком является `d`, следовательно `australia` - это каталог). Проверим его свойства и зададим новые права доступа командой `chmod` и следующими комбинациями: `u+r+w+x` - для пользователя, `g+r-w-x` - для группы и остальных пользователей.

Рисунок 23: каталог `australia`:

```
[ovgubina@localhost ~]$ cd
[ovgubina@localhost ~]$ mkdir australia
[ovgubina@localhost ~]$ ls -al australia
итого 4
drwxrwxr-x. 2 ovgubina ovgubina 6 май 7 15:55 .
drwx-----. 22 ovgubina ovgubina 4096 май 7 15:55 ..
[ovgubina@localhost ~]$ chmod u+r+w+x australia
[ovgubina@localhost ~]$ chmod g+r-w-x australia
[ovgubina@localhost ~]$ chmod o+r-w-x australia
[ovgubina@localhost ~]$ ls -al australia
итого 4
drwxr--r--. 2 ovgubina ovgubina 6 май 7 15:55 .
drwx-----. 22 ovgubina ovgubina 4096 май 7 15:55 ..
[ovgubina@localhost ~]$
```

Проверим свойства командой `ls`. Видим, что права доступа стали именно такими, как требовалось (рисунок 23 - подсвечено черным).

3.2. `drwx-x-x ... play` (рисунок 24)

Нам нужно обозначить неограниченные права для владельца каталога и право только на выполнение для группы и других пользователей.

Для этого сначала создадим каталог `play` командой `mkdir` (создаем каталог, поскольку первый знаком является `d`, следовательно `play` - это каталог). Проверим его свойства и зададим новые права доступа командой `chmod` и следующими комбинациями: `u+r+w+x` - для пользователя, `g-r-w+x` - для группы и остальных пользователей.

Рисунок 24: каталог `play`:

```
[ovgubina@localhost ~]$ mkdir play
[ovgubina@localhost ~]$ ls -al play
итого 4
drwxrwxr-x. 2 ovgubina ovgubina 6 май 7 15:58 .
drwx-----. 23 ovgubina ovgubina 4096 май 7 15:58 ..
[ovgubina@localhost ~]$ chmod u+r+w+x play
[ovgubina@localhost ~]$ chmod g-r-w+x play
[ovgubina@localhost ~]$ chmod o-r-w+x play
[ovgubina@localhost ~]$ ls -al play
итого 4
drwx-x-x-x. 2 ovgubina ovgubina 6 май 7 15:58 .
drwx-----. 23 ovgubina ovgubina 4096 май 7 15:58 ..
[ovgubina@localhost ~]$
```

Проверим свойства командой `ls`. Видим, что права доступа стали именно такими, как требовалось (рисунок 24).

3.3. `-r-xr--r-- ... my_os` (рисунок 25)

Нам нужно обозначить права для владельца файла на его чтение и выполнение и право только на чтение для группы и других пользователей.

Для этого сначала создадим файл `my_os` командой `touch` (создаем файл, поскольку первый знаком является `-`, следовательно `my_os` - это файл). Проверим его свойства и зададим новые права доступа командой `chmod` и следующими комбинациями: `u+r-w+x` - для пользователя, `g+r-w-x` / `o+r-w-x` - для группы и остальных пользователей.

Рисунок 25: файл `my_os`:

```
[ovgubina@localhost ~]$ touch my_os
[ovgubina@localhost ~]$ ls -l my_os
-rw-rw-r--. 1 ovgubina ovgubina 0 май 7 16:00 my_os
[ovgubina@localhost ~]$ chmod u+r-w+x my_os
[ovgubina@localhost ~]$ chmod g+r-w-x my_os
[ovgubina@localhost ~]$ chmod o+r-w-x my_os
[ovgubina@localhost ~]$ ls -l my_os
-r-xr--r--. 1 ovgubina ovgubina 0 май 7 16:00 my_os
[ovgubina@localhost ~]$
```

Проверим свойства командой `ls`. Видим, что права доступа стали именно такими, как требовалось (рисунок 25).

3.4. `-rw-rw-r-- ... feathers` (рисунок 26*)

Нам нужно обозначить права для владельца файла и группы на его чтение и запись и право только на чтение для других пользователей.

Для этого сначала создадим файл `feathers` командой `touch` (создаем файл, поскольку первый знаком является `-`, следовательно `feathers` - это файл). Проверим его свойства и зададим новые права доступа командой `chmod` и следующими комбинациями: `u+r+w-x` / `g+r+w-x` - для пользователя и группы, `o+r-w-x` - для остальных пользователей.

Рисунок 26: файл `feathers`:

```
[ovgubina@localhost ~]$ touch feathers
[ovgubina@localhost ~]$ ls -l feathers
-rw-rw-r--. 1 ovgubina ovgubina 0 май  7 16:03 feathers
[ovgubina@localhost ~]$ chmod u+r+w-x feathers
[ovgubina@localhost ~]$ chmod g+r+w-x feathers
[ovgubina@localhost ~]$ chmod o+r-w-x feathers
[ovgubina@localhost ~]$ ls -l feathers
-rw-rw-r--. 1 ovgubina ovgubina 0 май  7 16:03 feathers
[ovgubina@localhost ~]$
```

Проверим свойства командой `ls`. Видим, что права доступа стали именно такими, как требовалось (рисунок 26).

4.

4.1. Просмотрим содержимое файла `/etc/passwd` с помощью команды `cat` (рисунок 27).

Рисунок 27: просмотр содержимого файла:

```
[ovgubina@localhost ~]$ 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
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
libstoragemgmt:x:998:996:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
colord:x:997:995:User for colord:/var/lib/colord:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
saned:x:996:994:SANE scanner daemon user:/usr/share/sane:/sbin/nologin
```

4.2. Скопируем файл `~/feathers` в файл `~/file.old` командой `cp`, проверяем командой `ls`, что такой файл появился (рисунок 28).

Рисунок 28:

```
[ovgubina@localhost ~]$ cp ~/feathers ~/file.old
[ovgubina@localhost ~]$ ls ~/file.old
/home/ovgubina/file.old
```

4.3. Переместим файл `~/file.old` в каталог `~/play` с помощью команды `mv`, проверим правильность действий просмотром содержимого каталога, по выводу команды `ls` видим, что мы все сделали верно (рисунок 29).

Рисунок 29:

```
[ovgubina@localhost ~]$ mv ~/file.old ~/play
[ovgubina@localhost ~]$ ls ~/play
file.old
[ovgubina@localhost ~]$
```

4.4. Скопируем каталог `~/play` в каталог `~/fun` командой `cp`, проверим правильность действий просмотром содержимого каталога, по выводу команды `ls` видим, что мы все сделали верно (рисунок 30).

Рисунок 30:

```
[ovgubina@localhost ~]$ cp -r ~/play ~/fun
[ovgubina@localhost ~]$ ls ~/fun
file.old
[ovgubina@localhost ~]$
```

4.5. Переместим каталог `~/fun` в каталог `~/play` командой `mv` и назовите его `games` - переименуем его командой `mv` в `games` (рисунок 31). Проверим правильность действий просмотром содержимого каталога, по выводу команды `ls` видим, что мы все сделали верно (рисунок 31).

Рисунок 31:


```
[ovgubina@localhost ~]$ mv ~/fun ~/play
[ovgubina@localhost ~]$ ls ~/play
file.old  fun
[ovgubina@localhost ~]$ mv ~/play/fun ~/play/games
[ovgubina@localhost ~]$ ls ~/play
file.old  games
[ovgubina@localhost ~]$
```

4.6. Лишим владельца файла ~/feathers права на чтение командой `chmod u-r ...` (рисунок 32).

Рисунок 32:

```
[ovgubina@localhost ~]$ chmod u-r ~/feathers
[ovgubina@localhost ~]$ cat ~/feathers
cat: /home/ovgubina/feathers: Отказано в доступе
[ovgubina@localhost ~]$ cp ~/feathers ~/abc1
cp: невозможно открыть «/home/ovgubina/feathers» для чтения: Отказано в доступе
[ovgubina@localhost ~]$ chmod u+r ~/feathers
[ovgubina@localhost ~]$
```

4.7. Посмотрим, что произойдёт, если мы попытаемся просмотреть файл ~/feathers командой `cat` (рисунок 32). Видим, что мы не можем просмотреть этот файл, система выдает сообщение об отказе в доступе, поскольку ранее мы его ограничили, следовательно мы все сделали правильно.

4.8. Теперь проверим, что произойдёт, если мы попытаемся скопировать файл ~/feathers в файл, например, ~/abc1 (рисунок 32). Система выдает сообщение о том, что мы не можем открыть файл для чтения - отказано в доступе - все верно.

4.9. Возвращаем владельцу файла ~/feathers право на чтение командой `chmod` (рисунок 32).

4.10. Лишим владельца каталога ~/play права на выполнение командой `chmod u-x ...` (рисунок 33).

Рисунок 33:

```
[ovgubina@localhost ~]$ chmod u-x ~/play
[ovgubina@localhost ~]$ cd ~/play
bash: cd: /home/ovgubina/play: Отказано в доступе
[ovgubina@localhost ~]$ chmod u+x ~/play
[ovgubina@localhost ~]$ cd ~/play
[ovgubina@localhost play]$
```

4.11. Пробуем перейти в каталог ~/play (рисунок 33). У нас не получается этого сделать, поскольку мы ограничили права в пункте 4.10 - отказано в доступе.

4.12. Вернем владельцу каталога ~/play право на выполнение через команду `chmod` и снова попробуем перейти в него (рисунок 33). На этот раз это происходит успешно, поскольку мы вернули владельцу права на выполнение.

5. Прочитаем ман по командам `mount`, `fsck`, `mkfs`, `kill` (рисунок 34).

Рисунок 34: чтение `mount`, `fsck`, `mkfs`, `kill`:

```
[ovgubina@localhost ~]$
[ovgubina@localhost ~]$ man mount
[ovgubina@localhost ~]$ man fsck
[ovgubina@localhost ~]$ man mkfs
[ovgubina@localhost ~]$ man kill
[ovgubina@localhost ~]$
```

Теперь мы можем дать краткую характеристику по данным командам:

- `mount` - утилита командной строки в UNIX-подобных операционных системах, которая применяется для монтирования файловых систем:

```
$ mount <файл_устройства> <папка_назначения>
```

или

```
$ mount <опции> -t <файловая_система> -o <опции_монтирования> <файл_устройства> <папка_назначения>
```

- `fsck` - команда UNIX, которая проверяет и устраняет ошибки в файловой системе:

```
$ fsck [опции] [опции_файловой_системы] [раздел_диска]
```

- `mkfs` - позволяет создать файловую систему Linux:

```
$ sudo mkfs -t <тип> <устройство>
```

- `kill` - посылает указанный сигнал указанному процессу:

```
kill [ -s сигнал | -p ] [ -a ] pid...
```

или

```
kill -l [ сигнал ]
```

За более полной и понятной информацией можно обратиться на сайт [Losstf](#)^[2].

Вывод:

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.

Библиография:

[1]: [Материалы лаборторной работы №6](#)

[2]: [Losst - Linux Open Source Software Technologies](#)