

Отчет о выполнении лабораторной работы №6

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

Российский Университет Дружбы Народов

Факультет Физико-Математических и Естественных Наук

Дисциплина: *Операционные системы*

Студент: Алхатиб Осама

Группа: НПИбд-02-20

Москва, 2021г.

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.

1. Выполнил все примеры, приведённые в первой части описания лабораторной работы.
 2. Выполнил следующие действия:
 - Скопировал файл `/usr/include/sys/io.txt` в домашний каталог и назвал его `equipment`.
 - В домашнем каталоге создал директорию `~/ski.places`.
 - Переместил файл `equipment` в каталог `~/ski.places`.
 - Переименовал файл `~/ski.places/equipment` в `~/ski.places/equiplist`.
 - В домашнем каталоге создал файл `abc1` и скопировал его в каталог `~/ski.places`, назвал его `equiplist2`.
 - Создал каталог с именем `equipment` в каталоге `~/ski.places`.
 - Переместил файлы `~/ski.places/equiplist` и `equiplist2` в каталог `~/ski.places/equipment`.
 - Создал и переместил каталог `~/newdir` в каталог `~/ski.places`, назвав его `plans`.
 3. Определил опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:
-

- `drwxr--r-- ... australia`
- `drwx-x--x ... play`
- `-r-xr--r-- ... my_os`
- `-rw-rw-r-- ... feathers`

4. Проделал приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
-

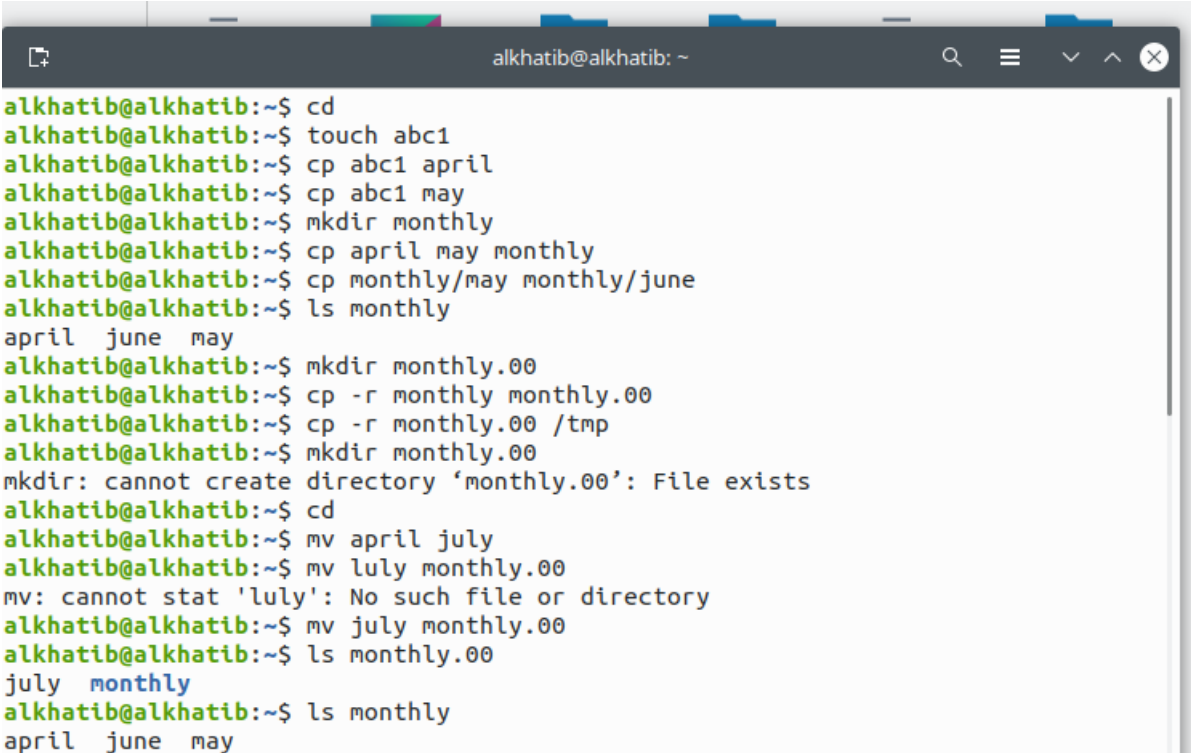
- Просмотрел содержимое файла `/etc/passwd`.
-

- Скопировал файл `~/feathers` в файл `~/file.old`.
 - Переместил файл `~/file.old` в каталог `~/play`.
 - Скопировал каталог `~/play` в каталог `~/fun`.
 - Переместил каталог `~/fun` в каталог `~/play` и назвала его `games`.
-

- Лишил владельца файла `~/feathers` права на чтение.
 - Не удалось просмотреть файл `~/feathers` командой `cat`.
 - Не получилось скопировать файл `~/feathers`.
 - Дал владельцу файла `~/feathers` право на чтение
-

- Лишил владельца каталога `~/play` права на выполнение.
- НЕ получилось перейти в каталог `~/play`.
- Дал владельцу каталога `~/play` право на выполнение

- **mount** - нужна для просмотра смонтированных файловых систем, а также для монтирования любых локальных или удаленных файловых систем. Например, при вызове команды «`mount /dev/cdrom /mnt/cdrom`» устройство `/dev/cdrom` монтируется в каталог `/mnt/cdrom`, если он существует. Начиная от момента монтирования и пока пользователь не отмонтирует файловую систему (или туда не будет смонтировано что-то иное) в каталоге `/mnt/cdrom` будет содержаться дерево каталогов устройства `/dev/cdrom`; те файлы, и подкаталоги, которые раньше находились в `/mnt/cdrom`, сохранятся, но будут недоступны до размонтирования устройства `/dev/cdrom`. Для размонтирования достаточно указать точку монтирования или имя устройства, команда «`umount /dev/cdrom`». При запуске команды `mount` без параметров выводится список смонтированных
 - файловых систем.
- **fsck** - проверяет и устраняет ошибки в файловой системе. Например, `fsck -fy -t ext4 /dev/sda1`. Опция `-f` (force) используется для принудительного выполнения проверки. Опция `-y` (yes) позволяет программе автоматически отвечать "да" на все вопросы в ходе работы.
- **mkfs** - действие заключается в создании указанной файловой системы на выбранном диске или разделе. Например, команда «`mkfs-text2 /dev/hda1`» создает файловую систему `ext2` на разделе `hda1`.
- **kill** - завершает некорректно работающее приложение. Например, чтобы послать сигнал `SIGKILL` (он имеет номер 9) процессу 2811, необходимо вызвать команду «`kill -9 2811`».



```
alkhatib@alkhatib: ~  
alkhatib@alkhatib:~$ cd  
alkhatib@alkhatib:~$ touch abc1  
alkhatib@alkhatib:~$ cp abc1 april  
alkhatib@alkhatib:~$ cp abc1 may  
alkhatib@alkhatib:~$ mkdir monthly  
alkhatib@alkhatib:~$ cp april may monthly  
alkhatib@alkhatib:~$ cp monthly/may monthly/june  
alkhatib@alkhatib:~$ ls monthly  
april  june  may  
alkhatib@alkhatib:~$ mkdir monthly.00  
alkhatib@alkhatib:~$ cp -r monthly monthly.00  
alkhatib@alkhatib:~$ cp -r monthly.00 /tmp  
alkhatib@alkhatib:~$ mkdir monthly.00  
mkdir: cannot create directory 'monthly.00': File exists  
alkhatib@alkhatib:~$ cd  
alkhatib@alkhatib:~$ mv april july  
alkhatib@alkhatib:~$ mv luly monthly.00  
mv: cannot stat 'luly': No such file or directory  
alkhatib@alkhatib:~$ mv july monthly.00  
alkhatib@alkhatib:~$ ls monthly.00  
july  monthly  
alkhatib@alkhatib:~$ ls monthly  
1. april  june  may
```

```
alkhatib@alkhatib: ~  
alkhatib@alkhatib:~$ ls monthly  
april  june  may  
alkhatib@alkhatib:~$ mv monthly.00 monthly.01  
alkhatib@alkhatib:~$ mkdir reports  
alkhatib@alkhatib:~$ mv monthly.01 reports  
alkhatib@alkhatib:~$ mv reports/monthly.01 reports/monthly  
alkhatib@alkhatib:~$ cd  
alkhatib@alkhatib:~$ touch may  
alkhatib@alkhatib:~$ ls -l may  
-rw-rw-r-- 1 alkhatib alkhatib 0 мая 10 17:36 may  
alkhatib@alkhatib:~$ chmod u+x may  
alkhatib@alkhatib:~$ ls -l may  
-rwxrw-r-- 1 alkhatib alkhatib 0 мая 10 17:36 may  
alkhatib@alkhatib:~$ chmod u-x may  
alkhatib@alkhatib:~$ ls -l may  
-rw-rw-r-- 1 alkhatib alkhatib 0 мая 10 17:36 may  
alkhatib@alkhatib:~$ cd  
alkhatib@alkhatib:~$ mkdir monthly  
mkdir: cannot create directory 'monthly': File exists  
alkhatib@alkhatib:~$ chmod 0-r monthly  
chmod: invalid mode: '0-r'  
Try 'chmod --help' for more information.  
alkhatib@alkhatib:~$ chmod o-r monthly  
alkhatib@alkhatib:~$ chmod g-r monthly  
2. -rw-rw-r-- 1 alkhatib alkhatib 0 мая 10 17:36 may  
alkhatib@alkhatib:~$ cd  
alkhatib@alkhatib:~$ mkdir monthly  
mkdir: cannot create directory 'monthly': File exists  
alkhatib@alkhatib:~$ chmod 0-r monthly  
chmod: invalid mode: '0-r'  
Try 'chmod --help' for more information.  
alkhatib@alkhatib:~$ chmod o-r monthly  
alkhatib@alkhatib:~$ chmod g-r monthly  
alkhatib@alkhatib:~$ cd  
alkhatib@alkhatib:~$ touch abc1  
alkhatib@alkhatib:~$ chmod g+w abc1  
3. alkhatib@alkhatib:~$ cp /usr/include/sys/io.h equipment  
alkhatib@alkhatib:~$ cp /usr/include/xorg/isdv4.h /home/alkhatib  
alkhatib@alkhatib:~$ mkdir ski.plases  
alkhatib@alkhatib:~$ mv isdv4.h equipment  
alkhatib@alkhatib:~$ mv equipment ~/ski.plases  
alkhatib@alkhatib:~$ mv ~/ski.plases/equipment ~/ski.plases/equiplist  
alkhatib@alkhatib:~$ cp ~/ski.plases  
cp: missing destination file operand after '/home/alkhatib/ski.plases'  
Try 'cp --help' for more information.  
alkhatib@alkhatib:~$ cp abc1 ~/ski.plases  
alkhatib@alkhatib:~$ mv ~/ski.plases/abc1 ~/ski.plases/equiplist2  
alkhatib@alkhatib:~$ cp ski.plases  
cp: missing destination file operand after 'ski.plases'  
Try 'cp --help' for more information.  
alkhatib@alkhatib:~$ cd ski.plases  
alkhatib@alkhatib:~/ski.plases$ mkdir equipment  
alkhatib@alkhatib:~/ski.plases$ cd  
4. alkhatib@alkhatib:~$
```

```
alkhatib@alkhatib:~$ mkdir australia
alkhatib@alkhatib:~$ mkdir play
alkhatib@alkhatib:~$ touch my_os
alkhatib@alkhatib:~$ touch feathers
alkhatib@alkhatib:~$ ls -ld australia play my_os feathers
drwxrwxr-x 2 alkhatib alkhatib 4096 мая 10 18:07 australia
-rw-rw-r-- 1 alkhatib alkhatib   0 мая 10 18:08 feathers
-rw-rw-r-- 1 alkhatib alkhatib   0 мая 10 18:08 my_os
drwxrwxr-x 2 alkhatib alkhatib 4096 мая 10 18:08 play
alkhatib@alkhatib:~$ chmod u+rw,g+r,o+r australia
alkhatib@alkhatib:~$ chmod u+rw,g+x,o+x play
alkhatib@alkhatib:~$ chmod u+rw,g+r,o+r my_os
alkhatib@alkhatib:~$ chmod u+rx,g+r,o+r my_os
alkhatib@alkhatib:~$ chmod u+rw,g+rw,o+r feathers
alkhatib@alkhatib:~$ ls -ld australia play my_os feathers
drwxrwxr-x 2 alkhatib alkhatib 4096 мая 10 18:07 australia
-rw-rw-r-- 1 alkhatib alkhatib   0 мая 10 18:08 feathers
-rwxrw-r-- 1 alkhatib alkhatib   0 мая 10 18:08 my_os
drwxrwxr-x 2 alkhatib alkhatib 4096 мая 10 18:08 play
alkhatib@alkhatib:~$
```

```
5. alkhatib@alkhatib:~$ chmod u+rx,g+r,o+r my_os
alkhatib@alkhatib:~$ chmod u+rw,g+rw,o+r feathers
alkhatib@alkhatib:~$ ls -ld australia play my_os feathers
drwxrwxr-x 2 alkhatib alkhatib 4096 мая 10 18:07 australia
-rw-rw-r-- 1 alkhatib alkhatib   0 мая 10 18:08 feathers
-rwxrw-r-- 1 alkhatib alkhatib   0 мая 10 18:08 my_os
drwxrwxr-x 2 alkhatib alkhatib 4096 мая 10 18:08 play
alkhatib@alkhatib:~$ ls /etc/password
ls: cannot access '/etc/password': No such file or directory
alkhatib@alkhatib:~$ cp feathers file.old
alkhatib@alkhatib:~$ mv file.old play
alkhatib@alkhatib:~$ cp -r play fun
alkhatib@alkhatib:~$ mv fun play/games
alkhatib@alkhatib:~$ chmod u-r feathers
alkhatib@alkhatib:~$ cat feathers
cat: feathers: Permission denied
alkhatib@alkhatib:~$ cp feathers feathers2
cp: cannot open 'feathers' for reading: Permission denied
alkhatib@alkhatib:~$ chmod u+r feathers
alkhatib@alkhatib:~$ chmod u-x play
alkhatib@alkhatib:~$ cd play
bash: cd: play: Permission denied
alkhatib@alkhatib:~$ chmod u+x play
alkhatib@alkhatib:~$
```

```
6. alkhatib@alkhatib:~$ chmod u+r feathers
alkhatib@alkhatib:~$ chmod u-x play
alkhatib@alkhatib:~$ cd play
bash: cd: play: Permission denied
alkhatib@alkhatib:~$ chmod u+x play
alkhatib@alkhatib:~$ man mount
alkhatib@alkhatib:~$ man fsck
alkhatib@alkhatib:~$ man mkfs
alkhatib@alkhatib:~$ man kill
7. alkhatib@alkhatib:~$
```

8.
9.

```
alkhatib@alkhatib: ~  
MOUNT(8) System Administration MOUNT(8)  
  
NAME  
    mount - mount a filesystem  
  
SYNOPSIS  
    mount [-l|-h|-V]  
  
    mount -a [-fFnrsvw] [-t fstype] [-O optlist]  
  
    mount [-fnrsvw] [-o options] device|dir  
  
    mount [-fnrsvw] [-t fstype] [-o options] device dir  
  
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 sev-  
    eral 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 another  
    services.  
  
Manual page mount(8) line 1 (press h for help or q to quit)
```

10.

```
alkhatib@alkhatib: ~  
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 code 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)
```



```
alkhatib@alkhatib: ~  
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>
```

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

```
alkhatib@alkhatib: ~  
alkhatib@alkhatib:~$ man mkfs  
alkhatib@alkhatib:~$ man kill  
alkhatib@alkhatib:~$ mount  
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)  
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)  
udev on /dev type devtmpfs (rw,nosuid,noexec,relatime,size=2533068k,nr_inodes=633267,mode=755)  
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)  
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=515712k,mode=755)  
/dev/sda5 on / type ext4 (rw,relatime,errors=remount-ro)  
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)  
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)  
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)  
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)  
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)  
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)  
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)  
none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)  
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
```

12.

```
alkhatib@alkhatib:~$ fsck  
fsck from util-linux 2.34  
e2fsck 1.45.5 (07-Jan-2020)  
/dev/sda5 is mounted.  
  
WARNING!!! The filesystem is mounted. If you continue you ***WILL***  
cause ***SEVERE*** filesystem damage.  
  
Do you really want to continue<n>?
```

13.

```
open: permission denied  
alkhatib@alkhatib:~$ mkfs -t ext2 /home
```

14.

Вывод

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.

Контрольные вопросы

1. На моем ноутбуке:

Ext4 - журналируемая файловая система, используемая в ОС на ядре Linux. Основана на файловой системе Ext3, но отличается тем, что в ней представлен механизм записи файлов в непрерывные участки блоков (екстенты), уменьшающий фрагментацию и повышающий производительность.

2. *Файловая система в дисплейном классе содержит следующие каталоги первого уровня:*

- /bin - Основные программы, необходимые для работы в системе: командные оболочки shell, основные утилиты.
- /boot - Каталог, который содержит ядро системы— главную программу, загружающую и исполняющую все остальные.
- /dev - Каталог, в котором содержатся псевдофайлы устройств. С точки зрения Linux все физические устройства, как главные, так и периферийные, представляют собой файлы особого типа, в которые система может записывать данные и из которых она может их считывать. Пользователь не должен работать с этими файлами, поскольку запись неправильных данных в файл устройства может повредить устройство или хранящиеся на нём данные.
- /etc - В этом каталоге содержатся системные конфигурационные файлы — текстовые файлы, которые считываются при загрузке системы и запуске программ и определяют их поведение. Настройка и администрирование Linux в конечном итоге сводится к редактированию этих файлов, даже если оно выполняется при помощи графических средств конфигурирования системы.
- /home - В структуре файловой системы Linux каждый пользователь имеет отдельный личный каталог для своих данных (т.н. домашний каталог), и все пользовательские каталоги выделены в отдельный общий каталог /home.
- /mnt - Каталоги для монтирования файловых систем сменных устройств и внешних файловых систем.
- /proc - Файловая система на виртуальном устройстве, её файлы содержат информацию о текущем состоянии системы.
- /root - Каталог администратора системы.
- /sbin - Системные утилиты.
- /usr - Программы и библиотеки, доступные пользователю.
- /var - Рабочие файлы программ, различные временные данные: очереди (письма на отправку, файлы на печать и др.), системные журналы (файлы, в которые записывается информация о происходящих в системе событиях).
- /tmp - Временные файлы.

3. *Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе?*

Для того чтобы содержимое некоторой файловой системы было доступно операционной системе, необходимо выполнить команду mount.

4. *Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы?*

Некорректность файловой системы может возникать не только в результате насильственного прерывания операций ввода-вывода, выполняемых непосредственно с диском, но и в результате нарушения работы дискового кэша. Кэширование данных с диска предполагает, что в течение некоторого времени результаты операций ввода-вывода никак не сказываются на содержимом диска — все изменения происходят с копиями блоков диска, временно хранящихся в буферах оперативной памяти. В этих буферах оседают данные из пользовательских файлов и служебная информация файловой системы, такая как каталоги, индексные дескрипторы, списки свободных, занятых и поврежденных блоков и т. п.

5. *Как создаётся файловая система?*

Разбитие диска на разделы и создание ФС в Linux делается при помощи специальных утилит – cfdisk fdisk sfdisk mke2fs mkfs mkfs.ext2 mkfs.ext3 mkfs.ext4 mkswap partimage parted указывая им в качестве аргумента конкретное блочное устройство (/dev/***). Блочные устройства HDD вида /dev/sda можно использовать целиком для единственной ФС, но это редко применяется на практике. Лучше разделить все пространство на разделы меньшего размера и использовать их под разные задачи.

6. *Дайте характеристику командам, которые позволяют просмотреть текстовые файлы.*

Для просмотра небольших файлов удобно пользоваться командой cat.

Формат команды: cat имя-файла Для просмотра больших файлов используйте команду less — она позволяет осуществлять постраничный просмотр файлов (длина страницы соответствует размеру экрана). Формат команды: less имя-файла Для управления процессом просмотра вы можете использовать следующие управляющие клавиши: Пробел → переход на следующую страницу, ENTER → сдвиг вперед на одну строку, b → возврат на предыдущую страницу, h → обращение за подсказкой, q → выход в режим командной строки. Для просмотра начала файла вы можете воспользоваться командой head. По умолчанию она выводит первые 10 строк файла.

7. *Приведите основные возможности команды cp в Linux.*

Копирование файлов и каталогов осуществляется при помощи команды cp. Формат команды: cp[-опции] исходный_файл_целевой_файл. Опция i в команде cp поможет избежать уничтожения информации в случае, если на место целевого файла вы поставите имя уже существующего файла: система попросит подтвердить, что вы хотите перезаписать этот файл. Команда cp с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами

8. *Назовите и дайте характеристику командам перемещения и переименования файлов и каталогов.*

Команды mv и mvdir предназначены для перемещения и переименования файлов и каталогов. Формат команды: mv [-опции] старый_файл_новый_файл. Для получения предупреждения перед переписыванием файла стоит использовать опцию i.

9. *Что такое права доступа? Как они могут быть изменены? При ответах на вопросы используйте дополнительные источники информации по теме.*

Права доступа определяют, кто и что может делать с содержимым файла. Существуют три группы прав доступа: для владельца файла, для членов

группы, для всех остальных. Для изменения прав доступа к файлу или каталогу используется команда `chmod`. Права доступа к файлу может поменять только владелец или суперпользователь (администратор). Формат команды: `chmod режим имя_файла`. Режим (в формате команды) имеет следующую структуру и способ записи:

- "=" установить право;
- "-" лишить права;
- "+" дать право;
- "r" чтение;
- "w" запись;
- "x" выполнение;
- "u" (user) владелец файла;
- "g" (group) группа, к которой принадлежит владелец файла;
- (others) все остальные.