

Лабораторная работа No 5

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

Бекназарова Виктория Тиграновна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12
4	Контрольные вопросы	13

Список иллюстраций

2.1	Выполнение примеров	6
2.2	Выполнение примеров	6
2.3	Выполнение примеров	6
2.4	Работа с каталогами	7
2.5	Настройка прав доступа	8
2.6	файла /etc/password	8
2.7	Работа с файлами и правами доступа	9
2.8	Команда mount	9
2.9	Команда fsck	10
2.10	Команда mkfs	10
2.11	Команда kill	11

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 Выполнение лабораторной работы

1. Выполним все примеры, приведённые в первой части описания лабораторной работы. (рис. 2.1) (рис. 2.2). (рис. 2.3)

```
vtbeknazarova@dk3n65 ~ $ cd
vtbeknazarova@dk3n65 ~ $ touch abc1
vtbeknazarova@dk3n65 ~ $ cp abc1 april
vtbeknazarova@dk3n65 ~ $ cp abc1 may
vtbeknazarova@dk3n65 ~ $ mkdir monthly
vtbeknazarova@dk3n65 ~ $ cp april may monthly
vtbeknazarova@dk3n65 ~ $ cp monthly/may monthly/june
vtbeknazarova@dk3n65 ~ $ ls monthly
april  june  may
vtbeknazarova@dk3n65 ~ $ mkdir monthly.00
vtbeknazarova@dk3n65 ~ $ cp -r monthly monthly.00
vtbeknazarova@dk3n65 ~ $ cp -r monthly.00 /tmp
vtbeknazarova@dk3n65 ~ $
```

Рис. 2.1: Выполнение примеров

```
vtbeknazarova@dk3n65 ~ $ mv april july
vtbeknazarova@dk3n65 ~ $ mv july monthly.00
vtbeknazarova@dk3n65 ~ $ ls monthly.00
july  monthly
vtbeknazarova@dk3n65 ~ $ mv monthly.00 monthly.01
vtbeknazarova@dk3n65 ~ $ mkdir reports
vtbeknazarova@dk3n65 ~ $ mv monthly.01 reports
vtbeknazarova@dk3n65 ~ $ mv reports/monthly.01 reports/monthly
vtbeknazarova@dk3n65 ~ $
vtbeknazarova@dk3n65 ~ $ cd
```

Рис. 2.2: Выполнение примеров

```
vtbeknazarova@dk3n65 ~ $ cd
vtbeknazarova@dk3n65 ~ $ touch may
vtbeknazarova@dk3n65 ~ $ ls -l may
-rw-r--r-- 1 vtbeknazarova studsci 0 map 9 15:41 may
vtbeknazarova@dk3n65 ~ $ chmod u+x may
vtbeknazarova@dk3n65 ~ $ ls -l may
-rwxr--r-- 1 vtbeknazarova studsci 0 map 9 15:41 may
vtbeknazarova@dk3n65 ~ $ chmod u-x may
vtbeknazarova@dk3n65 ~ $ ls -l may
-rw-r--r-- 1 vtbeknazarova studsci 0 map 9 15:41 may
vtbeknazarova@dk3n65 ~ $ cd
vtbeknazarova@dk3n65 ~ $ mkdir monthly
```

Рис. 2.3: Выполнение примеров

2. Скопируем файл `/usr/include/sys/io.h` в домашний каталог и назовите его `equipment`. Такого файла нет, взяли другой. В домашнем каталоге создаем директорию `~/ski.plases`. Переместим файл `equipment` в каталог `~/ski.plases`. Переименуем файл `~/ski.plases/equipment` в `~/ski.plases/equiplist`. Создаем в домашнем каталоге файл `abc1` и скопируем его в каталог `~/ski.plases`, назовем его `equiplist2`. Создаем каталог с именем `equipment` в каталоге `~/ski.plases`. Переместим файлы `~/ski.plases/equiplist` и `equiplist2` в каталог `~/ski.plases/equipment`. Создаем и переместим каталог `~/newdir` в каталог `~/ski.plases` и назовем его `plans`. (рис. 2.4)

```
vtbeknazarova@dk3n65 ~ $ cp /usr/include/linux/sysinfo.h ~
vtbeknazarova@dk3n65 ~ $ mv sysinfo/h equipment
mv: не удалось выполнить stat для 'sysinfo/h': Нет такого файла или каталога
vtbeknazarova@dk3n65 ~ $ mv sysinfo.h equipment
vtbeknazarova@dk3n65 ~ $ mkdir ski.plases
vtbeknazarova@dk3n65 ~ $ mv equipment ski.plases/
vtbeknazarova@dk3n65 ~ $ mv ski.plases/equipment ski.plases/equiplist.
vtbeknazarova@dk3n65 ~ $ touch abc1
vtbeknazarova@dk3n65 ~ $ cp abc1
cp: после 'abc1' пропущен операнд, задающий целевой файл
По команде «cp --help» можно получить дополнительную информацию.
vtbeknazarova@dk3n65 ~ $ cp abc1 ~
cp: 'abc1' и '/afs/.dk.sci.pfu.edu.ru/home/v/t/vtbeknazarova/abc1' - один и тот же файл
vtbeknazarova@dk3n65 ~ $ cp abc1 ski.plases/equiplist2
vtbeknazarova@dk3n65 ~ $ cd ski.plases/
vtbeknazarova@dk3n65 ~/ski.plases $ mkdir equipment
vtbeknazarova@dk3n65 ~/ski.plases $ mv equiplist equipment/
mv: не удалось выполнить stat для 'equiplist': Нет такого файла или каталога
vtbeknazarova@dk3n65 ~/ski.plases $ mv equiplist equipment/
mv: не удалось выполнить stat для 'equiplist': Нет такого файла или каталога
vtbeknazarova@dk3n65 ~/ski.plases $ mv equiplist. equipment/
vtbeknazarova@dk3n65 ~/ski.plases $ mv equiplist2 equipment/
vtbeknazarova@dk3n65 ~/ski.plases $ cd
vtbeknazarova@dk3n65 ~ $ mkdir newdir
vtbeknazarova@dk3n65 ~ $ mv newdir/ ski.plases/
vtbeknazarova@dk3n65 ~ $ mv ski.plases/newdir/ ski.plases/plans
vtbeknazarova@dk3n65 ~ $
```

Рис. 2.4: Работа с каталогами

3. Определим опции команды `chmod`, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав (рис. 2.5) нет: `(drwxr-r-) australia (drwx-x-x) play (-r-xr-r-)` `my_os (-rw-rw-r-) feathers`

```

vtbeknazarova@dk3n65 ~ $ cd /tmp/udc/learn/1/ - cd /tmp/udc/learn/1/
vtbeknazarova@dk3n65 ~ $
vtbeknazarova@dk3n65 ~ $ mkdir australia play
vtbeknazarova@dk3n65 ~ $ touch my_os feathers
vtbeknazarova@dk3n65 ~ $ chmod 744 australia
vtbeknazarova@dk3n65 ~ $ chmod 711 play/
vtbeknazarova@dk3n65 ~ $ chmod 544 my_os
vtbeknazarova@dk3n65 ~ $ chmod 664 feathers
vtbeknazarova@dk3n65 ~ $ ls -l
итого 81
-rwxr-xr-x 1 vtbeknazarova studsci 16072 ноя 16 11:54 1
-rw-r--r-- 1 vtbeknazarova studsci 552 ноя 16 11:02 1.cpp
drwxr-xr-x 2 vtbeknazarova studsci 2048 фев 5 00:01 2109
-rw-rw-r-- 1 vtbeknazarova studsci 0 мар 9 15:59 abc1
drwxr--r-- 2 vtbeknazarova studsci 2048 мар 9 16:11 australia
drwxr-xr-x 2 vtbeknazarova studsci 2048 фев 25 10:44 bin
-rw-r--r-- 1 vtbeknazarova studsci 8 окт 7 14:37 COURSE
-rw-rw-r-- 1 vtbeknazarova studsci 0 мар 9 16:12 feathers
drwxr-xr-x 3 vtbeknazarova studsci 2048 сен 28 10:36 GNUstep
-rwxr-xr-x 1 vtbeknazarova studsci 8668 ноя 21 15:13 hello
-rw-r--r-- 1 vtbeknazarova studsci 1488 ноя 21 15:11 list.lst
-rwxr-xr-x 1 vtbeknazarova studsci 9060 ноя 21 15:13 main
-rw-r--r-- 1 vtbeknazarova studsci 0 мар 9 15:41 may
drwx--x--x 2 vtbeknazarova studsci 2048 мар 9 15:36 monthly
-r-xr--r-- 1 vtbeknazarova studsci 0 мар 9 16:12 my_os
drwx--x--x 2 vtbeknazarova studsci 2048 мар 9 16:11 play
drwxr-xr-x 3 vtbeknazarova root 2048 сен 2 2022 public

```

Рис. 2.5: Настройка прав доступа

4. Просмотрим содержимое файла /etc/passwd (рис. 2.6)

```

vtbeknazarova@dk3n65 ~ $ cat /etc/passwd
root:x:0:0:System user; root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
daemon:x:2:2:daemon:/sbin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
lp:x:4:7:lp:/var/spool/lpd:/bin/false
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 program user:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/bin/false
uucp:x:10:14:uucp:/var/spool/uucppublic:/bin/false
operator:x:11:0:operator:/root:/bin/bash
man:x:13:15:System user; man:/dev/null:/sbin/nologin
postmaster:x:14:12:Postmaster user:/var/spool/mail:/sbin/nologin
cron:x:16:16:A user for sys-process/cronbase:/var/spool/cron:/sbin/nologin
ftp:x:21:21:ftp:/home/ftp:/bin/false
sshd:x:22:22:User for ssh:/var/empty:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/bin/false
squid:x:31:31:Squid:/var/cache/squid:/bin/false
gdm:x:32:32:User for running GDM:/var/lib/gdm:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/bin/false
games:x:35:35:games:/usr/games:/bin/bash
x:x:40:40:bind:/var/bind:/bin/false
mysql:x:60:60:MySQL program user:/dev/null:/sbin/nologin
postgres:x:70:70:PostgreSQL program user:/var/lib/postgresql:/bin/sh

```

Рис. 2.6: файла /etc/passwd

5. Скопируем файл ~/feathers в файл ~/file.old. Переместим файл ~/file.old в каталог ~/play. Скопируем каталог ~/play в каталог ~/fun. Переместим каталог ~/fun в каталог ~/play и назовем его games. Лишим владельца файла

~/feathers права на чтение.. Дайдим владельцу файла ~/feathers право на чтение.. Лишим владельца каталога ~/play права на выполнение. Перейдите в каталог ~/play. Дайте владельцу каталога ~/play право на выполнение.(рис. 2.7)

```
strelaysrv:x:496:499:added by portage for strelaysrv:/var/lib/syncthing-relaysrv:/sbin/nologin
lightdm:x:997:134:A user for x11-misc/lightdm:/var/lib/lightdm:/sbin/nologin
libreoffice:x:512:970:A user for the headless LibreOffice server:/var/lib/libreoffice:/sbin/nologin
flatpak:x:313:313:System user; flatpak:/dev/null:/sbin/nologin
tbeknazarova@dk3n65 ~ $ cp feathers file.old.
tbeknazarova@dk3n65 ~ $ mv file.old play/
mv: не удалось выполнить stat для 'file.old': Нет такого файла или каталога
tbeknazarova@dk3n65 ~ $ mv file.old. play/
tbeknazarova@dk3n65 ~ $ mkdir fun
tbeknazarova@dk3n65 ~ $ mv fun/ play/games
tbeknazarova@dk3n65 ~ $ chmod u-r feathers
tbeknazarova@dk3n65 ~ $ cat feathers
cat: feathers: Отказано в доступе
tbeknazarova@dk3n65 ~ $ cp feathers2
cp: невозможно открыть 'feathers' для чтения: Отказано в доступе
tbeknazarova@dk3n65 ~ $ chmod u+r feathers
tbeknazarova@dk3n65 ~ $ chmod u-x play
tbeknazarova@dk3n65 ~ $ chmod u-x play/
tbeknazarova@dk3n65 ~ $ cd play/
tbeknazarova@dk3n65 ~/play $ cd
tbeknazarova@dk3n65 ~ $ chmod +x play/
tbeknazarova@dk3n65 ~ $ man mount
tbeknazarova@dk3n65 ~ $ man fsck
tbeknazarova@dk3n65 ~ $ man mkfs
tbeknazarova@dk3n65 ~ $ man kill
tbeknazarova@dk3n65 ~ $
```

Рис. 2.7: Работа с файлами и правами доступа

6. Прочитаем ман по командам mount, fsck, mkfs, kill. (рис. 2.8) (рис. 2.9) (рис. 2.10) (рис. 2.11)

```
MOUNT(8)                                     System Administration                                     MOUNT(8)

NAME
    mount - mount a filesystem

SYNOPSIS
    mount [-h|-V]

    mount [-l] [-t fstype]

    mount -a [-fFnrsvw] [-t fstype] [-O optlist]

    mount [-fnrsvw] [-o options] device|mountpoint

    mount [-fnrsvw] [-t fstype] [-o options] device mountpoint

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

    mount --make-[shared|slave|private|unbindable|rshared|rslave|rprivate|runbindable] mountpoint

DESCRIPTION
    All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files can be spread out over several devices. The mount command serves to attach the filesystem found on
```

Рис. 2.8: Команда mount

```
FSCK(8)                                     System Administration                                     FSCK(8)

NAME
    fsck - check and repair a Linux filesystem

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

DESCRIPTION
    fsck is used to check and optionally repair one or more Linux filesystems. filesystem can be a device name (e.g., /dev/hdc1, /dev/sdb2), a mount point (e.g., /, /usr, /home), or an filesystem label or UUID specifier (e.g., UUID=8868abf6-88c5-4a83-98b8-bfc24057f7bd or LABEL=root). Normally, the fsck program will try to handle filesystems on different physical disk drives in parallel to reduce the total amount of time needed to check all of them.

    If no filesystems are specified on the command line, and the -A option is not specified, fsck will default to checking filesystems in /etc/fstab serially. This is equivalent to the -As options.

    The exit status returned by fsck is the sum of the following conditions:

    0
        No errors
```

Рис. 2.9: Команда fsck

```
MKFS(8)                                     System Administration                                     MKFS(8)

NAME
    mkfs - build a Linux filesystem

SYNOPSIS
    mkfs [options] [-t type] [fs-options] device [size]

DESCRIPTION
    This mkfs frontend is deprecated in favour of filesystem specific mkfs.<type> utils.

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

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

    In actuality, mkfs is simply a front-end for the various filesystem builders (mkfs.fstype) available under Linux. The filesystem-specific builder is searched for via your PATH environment setting only. Please see the filesystem-specific builder manual pages for further details.

OPTIONS
    -t, --type=type
```

Рис. 2.10: Команда mkfs

```
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>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is explained in signal(7) manual page.
```

Рис. 2.11: Команда kill

3 Выводы

В ходе данной работы мы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Научились совершать базовые операции с файлами, управлять правами их доступа для пользователя и групп. Ознакомились с Анализом файловой системы. А также получили базовые навыки по проверке использования диска и обслуживанию файловой системы.

4 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу.

Ответ: Ext2FS (расширенная файловая система номер два). Многие годы ext2 была файловой системой по умолчанию в GNU/Linux. Ext2 заменила собой Extended File System (вот откуда появилось “Second” в названии). В “новой” файловой системе были исправлены некоторые проблемы, а также убраны ограничения. Отличная стабильность, комплексные инструментальные средства для спасения удаленных файлов, очень долгое время перезагрузки после аварии, есть вероятность частичной или полной потери данных после аварии. Одним из главных недостатков “традиционных” файловых систем, подобных Ext2FS, является низкая сопротивляемость к резким системным сбоям (сбой питания или авария программного обеспечения)

Ext3 (Расширенная файловая система номер три) - является наследником файловой системы Ext2FS. Ext3 совместима с Ext2, но обладает одной новой и очень интересной особенностью – запись. Процесс сохранения объекта происходит прежде чем запись в журнал. В результате мы получаем всегда последовательную файловую систему. Это приводит к тому, что при появлении проблем, проверка и восстановление происходят очень быстро. Время, потраченное на то, чтобы проверить файловую систему таким образом, пропорционально его фактическому использованию и не больше его размера.

ReiserFS (Это тоже журналируемая файловая система подобно Ext3FS, но их внутренняя структура радикально отличается. В ReiserFS используется концепция

бинарных деревьев (binary-tree), позаимствованная из программного обеспечения баз данных.

JFS (журналируемая файловая система). JFS была разработана и использовалась IBM. Вначале JFS была закрытой системой, но недавно IBM решила открыть доступ для движения свободного программного обеспечения. Внутренняя структура JFS близка к ReiserFS. Средняя стабильность, нет комплексных инструментальных средств для спасения удаленных файлов, очень быстрая перезагрузка после аварии, очень хорошее восстановление данных после аварии.

2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры. Ответ:

- Загрузочный блок занимает первый блок файловой системы. Только корневая файловая система имеет активный загрузочный блок, хотя место для него резервируется в каждой файловой системе.
- Суперблок располагается непосредственно за загрузочным блоком и содержит самую общую информацию о ФС (размер ФС, размер области индексных дескрипторов, их число, список свободных блоков, свободные индексные дескрипторы и т. д.). Суперблок всегда находится в оперативной памяти. Различные версии ОС Unix способны поддерживать разные типы файловых систем. Поэтому у структуры суперблока могут быть варианты (сведения о свободных блоках, например, часто хранятся не как список, а как шкала бит), но суперблок всегда располагается за загрузочным блоком. При монтировании файловой системы в оперативной памяти создается копия ее суперблока. Все последующие операции по созданию и удалению файлов влекут изменения копии суперблока в оперативной памяти. Эта копия периодически записывается на магнитный диск. Обычно причиной повреждения файловой системы является отключение электропитания (или зависание ОС) в тот момент, когда система производит копирование суперблока из оперативной памяти на магнитный диск.

- Область индексных дескрипторов содержит описатели файлов (inode). С каждым файлом связан один inode, но одному inode может соответствовать несколько файлов. Binode хранится вся информация о файле, кроме его имени. Область индексных дескрипторов имеет фиксированный формат и располагается непосредственно за суперблоком. Общее число описателей и, следовательно, максимальное число файлов задается в момент создания файловой системы. Описатели нумеруются натуральными числами. Первый описатель используется ОС для описания специального файла (файла «Плохих блоков»). То есть поврежденные блоки раздела рассматриваются ОС как принадлежащие к специальному файлу и поэтому считаются «занятыми». Вторым описывает корневой каталог файловой системы.
 - В области данных расположены как обычные файлы, так и файлы каталогов (в том числе корневой каталог). Специальные файлы представлены в ФС только записями в соответствующих каталогах и индексными дескрипторами специального формата, т. е. места в области памяти не занимают.
3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе? Ответ: Команда `cat` - позволяет вывести на экран содержимое любого файла, однако в таком виде эта команда практически не используется. Если файл слишком большой, то его содержимое пролистается на экране, а Вы увидите только последние строки файла. С помощью этой команды можно комбинировать и объединять копии файлов, а также создавать новые файлы. Если набрать просто в командной строке `cat` и нажать `Enter`, то можно вводить (и соответственно видеть) текст на экране. Повторное нажатие клавиши `Enter` удвоит строку и позволит начать следующую. Когда текст набран, следует одновременно нажать клавиши `Ctrl` и `d`.

4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы? Ответ: Некорректность файловой системы может возникать:

- В результате насильственного прерывания операций ввода-вывода, выполняемых непосредственно с диском.
- 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? Ответ: Общее дерево файлов и каталогов системы Linux формируется из отдельных “ветвей”, соответствующих различным физическим носителям. В UNIX нет понятия “форматирования диска” (и команды форматирования), а используется понятие “создание файловой системы”. Когда мы получаем новый носитель, например, жесткий диск, мы должны создать на нем файловую систему. То есть каждому носителю ставится в соответствие отдельная файловая система. Чтобы эту файловую систему использовать для записи в нее файлов, надо ее вначале подключить в общее дерево каталогов (“смонтировать”). Вот и получается, что можно говорить о монтировании файловых систем или о монтировании носителей (с созданными на них файловыми системами). Например, создается файловая система типа ext2fs. Создание файловой системы типа ext2fs подразумевает создание в данном разделе на диске суперблока, таблицы индексных дескрипторов и совокупности блоков данных. Делается все это все с помощью команды mkfs. В простейшем случае достаточно дать эту команду в следующем формате:

[root]# mkfs -t ext2 /dev/hda5, где /dev/hda5 надо заменить указанием на соответствующее устройство или раздел. Например, если вы хотите создать файловую систему на дискете, то команда примет вид:

```
[root]# mkfs -t ext2 /dev/fd0
```

После выполнения команды `mkfs` в указанном разделе будет создана файловая система `ext2fs`. В новой файловой системе автоматически создается один каталог с именем `lost+found`. Он используется в экстренных случаях программой `fsck`, поэтому не удаляйте его. Для того, чтобы начать работать с новой файловой системой, необходимо подключить ее в общее дерево каталогов, что делается с помощью команды `mount`. В качестве параметров команде `mount` надо, как минимум, указать устройство и “точку монтирования”. Точкой монтирования называется тот каталог в уже существующем и известном системе дереве каталогов, который будет теперь служить корневым каталогом для подключаемой файловой системы. После монтирования файловой системы в каталог `/mnt/disk2` прежнее содержимое этого каталога станет для вас недоступно до тех пор, пока вы не размонтируете вновь подключенную файловую систему. Прежнее содержимое не уничтожается, а просто становится временно недоступным. Поэтому в качестве точек монтирования лучше использовать пустые каталоги (заранее заготовленные).

6. Дайте характеристику командам, которые позволяют просмотреть текстовые файлы. Ответ: 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 можно использовать следующие управляющие клавиши: - `Space` — 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 можно воспользоваться командой `head`. По умолчанию она выводит первые 10 строк файла. Формат команды: `head [-n] имя-файла`, где `n` — количество выводимых строк.

Команда `tail` выводит несколько (по умолчанию 10) последних строк файла. Формат команды: `tail [-n] имя-файла`, где `n` — количество выводимых строк.

7. Приведите основные возможности команды `cp` в Linux. Ответ: Копирование отдельных файлов Для копирования файла следует использовать утилиту `cp` с аргументами, представленными путями к исходному и целевому файлам.

Копирование файлов в другую директорию В том случае, если в качестве пути к целевому файлу используется путь к директории, исходные файлы будут скопированы в эту целевую директорию.

Команда `cp -r` Для копирования директорий целиком следует использовать команду `cp -r` (параметр `-r` позволяет осуществлять рекурсивное копирование всех файлов из всех поддиректорий).

Копирование множества файлов в директорию Вы также можете использовать утилиту `cp` для копирования множества файлов в одну директорию. В этом случае последний аргумент (аргумент, указывающий на цель) должен быть представлен путем к директории.

Команда `cp -i` Для предотвращения перезаписи существующих файлов в ходе использования утилиты `cp` следует использовать параметр `-i` (для активации интерактивного режима копирования).

8. Назовите и дайте характеристику командам перемещения и переименования файлов и каталогов. Ответ: Команды `mv` и `mkdir` предназначены для перемещения и переименования файлов и каталогов. Формат команды `mv`: `mv [-опции] старый_файл новый_файл` Примеры:

- Переименование файлов в текущем каталоге. Изменить название файла `april` на `july` в домашнем каталоге: `cd mv april july`

- Перемещение файлов в другой каталог. Переместить файл july в каталог monthly.00: `mv july monthly.00` `ls monthly.00` Результат: april july june may. Если необходим запрос подтверждения о перезаписи файла, то нужно использовать опцию `i`.
- 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`

9. Что такое права доступа? Как они могут быть изменены? Ответ: Права доступа — совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам). 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` Режим (в формате команды) имеет следующие компоненты структуры 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 `o` (others) все остальные В работе с правами доступа можно использовать их цифровую запись (восьмеричное значение) вместо символьной