

Лабораторная работа №5

Поляков Глеб Сергеевич

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	7
3.0.1 Команды для работы с файлами и каталогами	7
4 Выполнение лабораторной работы	9
4.0.1 Контрольные вопросы: .	14
5 Выводы	22
Список литературы	23

Список иллюстраций

4.1 Задание 2	12
4.2 Задание 3	13

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. Выполнить все примеры, приведённые в первой части описания лабораторной работы.
2. Выполнить действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения.
3. Определить опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет При необходимости создать нужные файлы.
4. Проделай упражнения, записывая в отчёт по лабораторной работе используемые при этом команды.
5. Прочитать man по командам mount, fsck, mkfs, kill и кратко их охарактеризовать, приведя примеры.

3 Теоретическое введение

3.0.1 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 умолчанию 10 последних строк файла. Формат команды:

`tail [-n] имя-файла,`

где n — количество выводимых строк. ### Копирование файлов и каталогов

Команда `cр` используется для копирования файлов и каталогов. Формат команды:

`cp [-опции] исходный_файл целевой_файл`

Здесь описываются теоретические аспекты, связанные с выполнением работы.

4 Выполнение лабораторной работы

1. Выполните все примеры, приведённые в первой части описания лабораторной работы. (рис. fig. ??) - (рис. fig. ??).

The image displays four separate screenshots of a terminal window titled "Terminal" from the Unity desktop environment. Each screenshot shows a different step in a sequence of file operations:

- Top-left:** The user has run several commands: "cd", "touch abc1", "cp abc1 april", and "cp abc1 may". The terminal prompt ends with a dollar sign.
- Top-right:** The user has run "mkdir monthly", "cp april may monthly", and then pressed the Enter key again.
- Bottom-left:** The user has run "cp monthly/may monthly/june" and "ls monthly". The output shows files named "april", "june", and "may".
- Bottom-right:** The user has run "mkdir monthly.00", "cp -r monthly monthly.00", "cp -r monthly.00 /tmp", and then pressed the Enter key again.

The image shows four terminal windows side-by-side, each with a dark background and light-colored text. The windows are part of a desktop environment with a top bar labeled "Activities" and "Terminal".

- Top-left Terminal:** Shows commands for renaming files. The user navigates to the current directory (~), moves "april" to "july", moves "july" to "monthly.00", lists "monthly.00", moves "monthly.00" to "monthly.01", and then exits.
- Top-right Terminal:** Shows commands for creating a directory and moving files. The user creates a directory "reports", moves "monthly.01" to it, and then moves "reports" to "monthly.01".
- Bottom-left Terminal:** Shows commands for creating and modifying a file. The user changes to the current directory (~), creates a file "may", lists it, changes its mode to u+x, lists it again, and then exits.
- Bottom-right Terminal:** Shows commands for changing file modes. The user changes to the current directory (~), changes the mode of "may" to u-x, lists it, and then exits.

The screenshot shows two terminal windows side-by-side. The top terminal window has a light gray header bar with the text 'Activities' and 'Terminal'. The bottom terminal window has a dark header bar with the text 'Activities' and 'Terminal'. Both windows have a small '+' icon in the top-left corner. The left terminal window contains the following command-line session:

```
gspolyakov@ubuntu:~$ chmod g-r monthly
gspolyakov@ubuntu:~$ chmod o-r monthly
gspolyakov@ubuntu:~$ ls -l monthly
total 0
-rw-rw-r-- 1 gspolyakov gspolyakov 0 мэр 9 13:35 april
-rw-rw-r-- 1 gspolyakov gspolyakov 0 мэр 9 13:35 june
-rw-rw-r-- 1 gspolyakov gspolyakov 0 мэр 9 13:35 may
gspolyakov@ubuntu:~$ cd
gspolyakov@ubuntu:~$
```

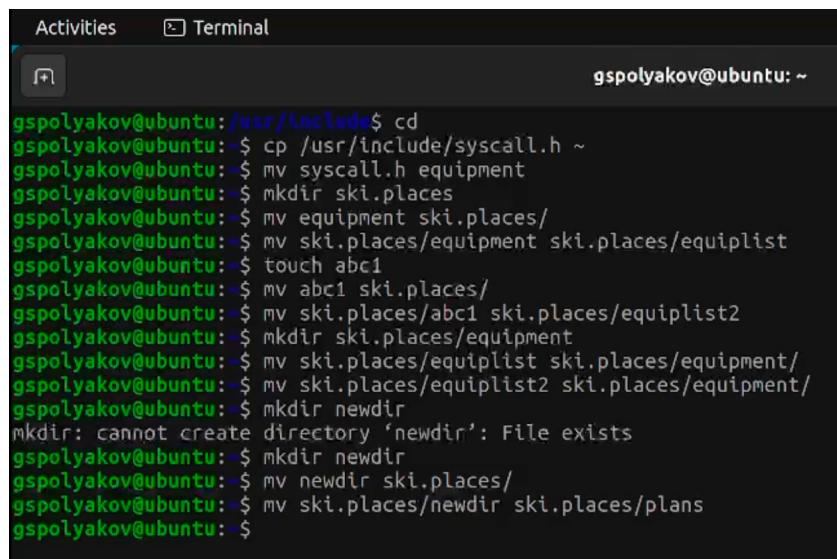
The right terminal window also has a light gray header bar with the text 'Activities' and 'Terminal'. It contains the following command-line session:

```
gspolyakov@ubuntu:~$ cd
gspolyakov@ubuntu:~$ touch abc1
gspolyakov@ubuntu:~$ chmod g+w abc1
gspolyakov@ubuntu:~$ ls -l abc1
-rw-rw-r-- 1 gspolyakov gspolyakov 0 мэр 9 13:35 abc1
gspolyakov@ubuntu:~$
```

2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
2. 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.
 3. В домашнем каталоге создайте директорию ~/ski.plases.
 4. Переместите файл equipment в каталог ~/ski.plases.
 5. Переименуйте файл ~/ski.plases/equipment в ~/ski.plases/equiplist.

6. Создайте в домашнем каталоге файл abc1 и скопируйте его в каталог ~/ski.places, назовите его equiplist2.
7. Создайте каталог с именем equipment в каталоге ~/ski.places.
8. Переместите файлы ~/ski.places/equipelist и equipelist2 в каталог ~/ski.places/equipment.
9. Создайте и переместите каталог ~/newdir в каталог ~/ski.places и назовите его plans.

(рис. fig. 4.1)



```

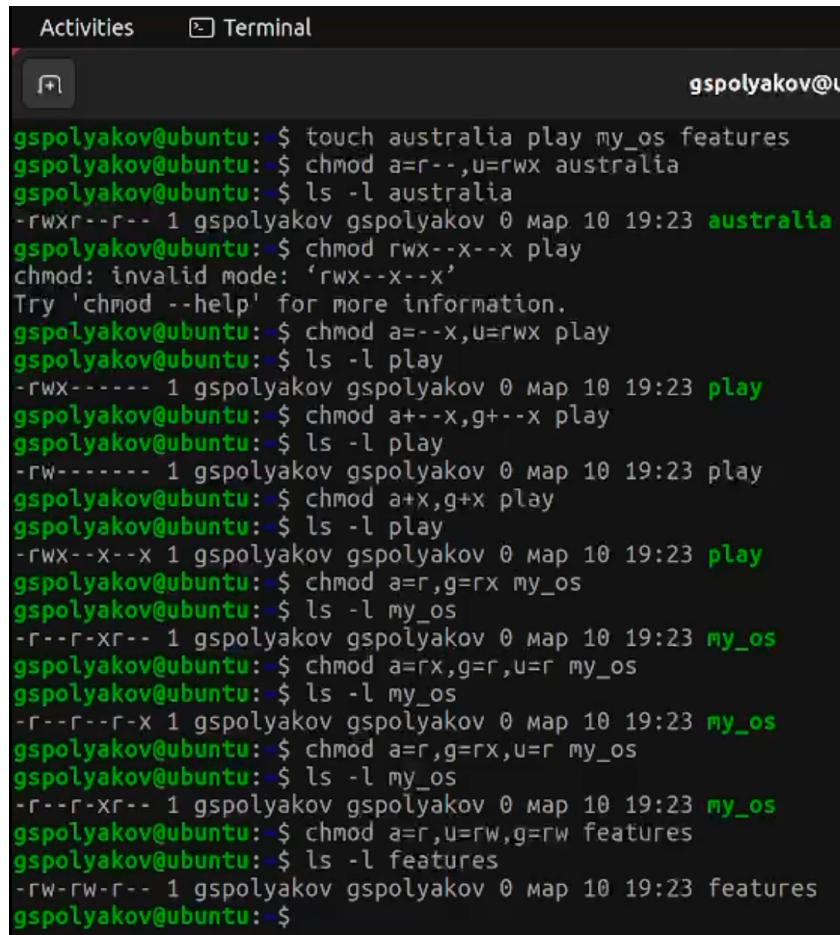
Activities Terminal gspolyakov@ubuntu: ~
gspolyakov@ubuntu: /usr/include$ cd
gspolyakov@ubuntu: $ cp /usr/include/syscall.h ~
gspolyakov@ubuntu: $ mv syscall.h equipment
gspolyakov@ubuntu: $ mkdir ski.places
gspolyakov@ubuntu: $ mv equipment ski.places/
gspolyakov@ubuntu: $ mv ski.places/equipment ski.places/equipelist
gspolyakov@ubuntu: $ touch abc1
gspolyakov@ubuntu: $ mv abc1 ski.places/
gspolyakov@ubuntu: $ mv ski.places/abc1 ski.places/equipelist2
gspolyakov@ubuntu: $ mkdir ski.places/equipment
gspolyakov@ubuntu: $ mv ski.places/equipelist ski.places/equipment/
gspolyakov@ubuntu: $ mv ski.places/equipelist2 ski.places/equipment/
gspolyakov@ubuntu: $ mkdir newdir
mkdir: cannot create directory 'newdir': File exists
gspolyakov@ubuntu: $ mkdir newdir
gspolyakov@ubuntu: $ mv newdir ski.places/
gspolyakov@ubuntu: $ mv ski.places/newdir ski.places/plans
gspolyakov@ubuntu: $

```

Рис. 4.1: Задание 2

3. Определите опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:

3. drwxr-r- ... australia
4. drwx-x-x ... play
5. -r-xr-r- ... my_os
6. -rw-rw-r- ... feathers При необходимости создайте нужные файлы.(рис. fig. 4.1)



The screenshot shows a terminal window titled "Terminal" with the user "gspolyakov@ubuntu". The terminal displays a series of commands related to file permissions:

```
gspolyakov@ubuntu:~$ touch australia play my_os features
gspolyakov@ubuntu:~$ chmod a=r--,u=rwx australia
gspolyakov@ubuntu:~$ ls -l australia
-rwxr--r-- 1 gspolyakov gspolyakov 0 мар 10 19:23 australia
gspolyakov@ubuntu:~$ chmod rwx--x--x play
chmod: invalid mode: 'rwx--x--x'
Try 'chmod --help' for more information.
gspolyakov@ubuntu:~$ chmod a=--x,u=rwx play
gspolyakov@ubuntu:~$ ls -l play
-rwx----- 1 gspolyakov gspolyakov 0 мар 10 19:23 play
gspolyakov@ubuntu:~$ chmod a+--x,g+--x play
gspolyakov@ubuntu:~$ ls -l play
-rw----- 1 gspolyakov gspolyakov 0 мар 10 19:23 play
gspolyakov@ubuntu:~$ chmod a+x,g+x play
gspolyakov@ubuntu:~$ ls -l play
-rwx--x--x 1 gspolyakov gspolyakov 0 мар 10 19:23 play
gspolyakov@ubuntu:~$ chmod a=r,g=rx my_os
gspolyakov@ubuntu:~$ ls -l my_os
-r--r-xr-- 1 gspolyakov gspolyakov 0 мар 10 19:23 my_os
gspolyakov@ubuntu:~$ chmod a=rx,g=r,u=r my_os
gspolyakov@ubuntu:~$ ls -l my_os
-r--r--r-x 1 gspolyakov gspolyakov 0 мар 10 19:23 my_os
gspolyakov@ubuntu:~$ chmod a=r,g=rx,u=r my_os
gspolyakov@ubuntu:~$ ls -l my_os
-r--r--r-x 1 gspolyakov gspolyakov 0 мар 10 19:23 my_os
gspolyakov@ubuntu:~$ chmod a=r,u=rw,g=rw features
gspolyakov@ubuntu:~$ ls -l features
-rw-rw-r-- 1 gspolyakov gspolyakov 0 мар 10 19:23 features
gspolyakov@ubuntu:~$
```

Рис. 4.2: Задание 3

4. Проделайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
 4. Просмотрите содержимое файла /etc/password.
 5. Скопируйте файл ~/feathers в файл ~/file.old.
 6. Переместите файл ~/file.old в каталог ~/play.
 7. Скопируйте каталог ~/play в каталог ~/fun.
 8. Переместите каталог ~/fun в каталог ~/play и назовите его games.
 9. Лишите владельца файла ~/feathers права на чтение.
10. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?

11. Что произойдёт, если вы попытаетесь скопировать файл ~/feathers?
 12. Дайте владельцу файла ~/feathers право на чтение.
 13. Лишите владельца каталога ~/play права на выполнение.
 14. Перейдите в каталог ~/play. Что произошло?
 15. Дайте владельцу каталога ~/play право на выполнение.
5. Прочитайте man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, приведя примеры.(рис. fig. ??),(рис. fig. ??).

```

Activities Terminal gspolyakov@ubuntu: ~
Activities Terminal gspolyakov@ubuntu: ~

gspolyakov@ubuntu: $ cp features file.old
gspolyakov@ubuntu: $ ls
australia documents features may Music Pictures Public ski_places Templates work
Desktop Downloads file.old games play reports snap Videos
gspolyakov@ubuntu: $ mv file.old /play
mv: cannot move 'file.old' to '/play': Permission denied
gspolyakov@ubuntu: $ mvdir file.old ~/play
Command 'mvdir' not found, did you mean:
  command 'svdir' from deb svtools (0.6-5)
  command 'mvt' from deb mtools (4.0.33-1+really4.0.32-1build1)
  command 'mvdir' from deb camna-utils (3.7p3-17)
  command 'Mvdir' from deb stmm (3.8.1-6.1)
  command 'mkdr' from deb coreutils (8.32-4.1ubuntu1)
  command 'vdrl' from deb coreutils (8.32-4.1ubuntu1)
Try: sudo apt install <deb name>
gspolyakov@ubuntu: $ mkdir play | mv file.old /play
mv: cannot move 'file.old' to '/play': Permission denied
mkdir: cannot create directory 'play': File exists
gspolyakov@ubuntu: $ mv file.old play
gspolyakov@ubuntu: $ ls play
play
gspolyakov@ubuntu: $ ls play/play
ls: cannot access 'play/play': Not a directory
gspolyakov@ubuntu: $ mv play file.old
gspolyakov@ubuntu: $ mkdir play
gspolyakov@ubuntu: $ mv file.old ~/play/
gspolyakov@ubuntu: $ ls play
file.old
gspolyakov@ubuntu: $ mkdir fun
gspolyakov@ubuntu: $ mv play/ fun/
gspolyakov@ubuntu: $ mv fun/play/ fun/games
gspolyakov@ubuntu: $ ls fun
games
gspolyakov@ubuntu: $ mv fun/games/ fun/play/
gspolyakov@ubuntu: $ mv fun/play/
mv: missing destination file operand after 'fun/play/'
Try 'mv --help' for more information.
gspolyakov@ubuntu: $ mv fun/play/ -
gspolyakov@ubuntu: $ mv fun/ play/
gspolyakov@ubuntu: $ ls play
ls: cannot access 'play': No such file or directory
gspolyakov@ubuntu: $ ls play/
file.old
gspolyakov@ubuntu: $ mv play/fun/ play/games
gspolyakov@ubuntu: $ chmod o-r features
gspolyakov@ubuntu: $ ls -l features
-rw-rw--- 1 gspolyakov gspolyakov 0 мэр 10 19:23 features
gspolyakov@ubuntu: $ chmod o-x play/
gspolyakov@ubuntu: $ cd play/
gspolyakov@ubuntu: $ rm *
cd
gspolyakov@ubuntu: $ chmod u-x play/
gspolyakov@ubuntu: $ chmod o+r,u-r features
gspolyakov@ubuntu: $ ls -l features
--w-rw--- 1 gspolyakov gspolyakov 0 мэр 10 19:23 features
gspolyakov@ubuntu: $ ls -l play/
ls: cannot access 'play/games': Permission denied
ls: cannot access 'play/file.old': Permission denied
total 0
???????? ? ? ? ? ? file.old
d????????? ? ? ? ? ? games
gspolyakov@ubuntu: $ cd play/
bash: cd: play/: Permission denied
gspolyakov@ubuntu: $ chmod u+x play/
gspolyakov@ubuntu: $ man mount
gspolyakov@ubuntu: $ man fsck
gspolyakov@ubuntu: $ man mkfs
gspolyakov@ubuntu: $ man kill
gspolyakov@ubuntu: $
```

4.0.1 Контрольные вопросы:

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

На жестком диске компьютера, который я использую для выполнения лабораторных работ, установлены две операционные системы - MacOS и Linux

Ubuntu. Каждая из этих операционных систем использует свою собственную файловую систему.

1. MacOS использует файловую систему HFS+ (Hierarchical File System Plus) или APFS (Apple File System): HFS+ была введена Apple в 1998 году и она поддерживает многие функции, такие как журналирование и кодирование имен файлов. Она была разработана для использования на жестких дисках, но также может работать на других устройствах, таких как оптические диски и USB-накопители. APFS была выпущена Apple в 2017 году как замена HFS+. Эта файловая система также поддерживает журналирование и кодирование имен файлов, но также имеет ряд новых функций, таких как копирование на основе снимков, шифрование на уровне файлов и поддержку метаданных. Она также предназначена для использования на жестких дисках и других устройствах.
2. Linux Ubuntu использует файловую систему ext4 (Fourth Extended Filesystem): ext4 является последней версией файловой системы ext, разработанной для использования в Linux. Она была выпущена в 2008 году и является улучшенной версией предыдущей файловой системы ext3. Она поддерживает функции, такие как журналирование, кодирование имен файлов, поддержку файлов большого размера и расширяемость, позволяющую увеличивать размер файловой системы по мере необходимости.
2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры.

Файловая система - это способ организации файлов и каталогов на компьютере или сервере. Общая структура файловой системы на Unix-подобных системах имеет следующую форму: \$ /

xxx bin

xxx boot

xxx dev

xxx etc

xxx home

xxx lib

xxx media

xxx mnt

xxx opt

xxx proc

xxx root

xxx run

xxx sbin

xxx srv

xxx sys

xxx tmp

xxx usr\$

Каждая директория первого уровня имеет следующие характеристики: **/bin** содержит основные исполняемые файлы, необходимые для работы системы и доступных для всех пользователей. **/boot** содержит ядро операционной системы и файлы загрузчика, необходимые для запуска системы. **/dev** содержит файлы, представляющие устройства и периферийные устройства, подключенные к системе. **/etc** содержит конфигурационные файлы для системных приложений и служб, таких как файлы настройки сети, файлы настройки пользователя, файлы настройки сервисов и т.д. **/home** содержит домашние каталоги для всех пользователей. **/lib** содержит разделяемые библиотеки, которые используются различными программами на системе. **/media** содержит точки монтирования для сменных носителей,

таких как USB-накопители, CD / DVD и т. д. /mnt содержит временные точки монтирования для файловых систем. /opt содержит дополнительные приложения, установленные на системе. /proc представляет системную информацию в виде файловой системы. Эта директория не содержит фактических файлов, но содержит информацию о текущих процессах, загрузке ядра и другой системной информации. /root содержит домашний каталог для суперпользователя root. /run содержит временные файлы, созданные различными системными приложениями во время работы системы. /sbin содержит основные исполняемые файлы для административных задач, таких как настройка сети и других системных сервисов. /srv содержит данные для служб, работающих на системе. /sys предоставляет доступ к системной информации о ядре и устройствах в виде файловой системы. /tmp содержит временные файлы, созданные приложениями и процессами во время работы системы. /usr содержит большую часть программного обеспечения.

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

Чтобы содержимое некоторой файловой системы было доступно операционной системе, необходимо произвести операцию “монтирования” (mounting) файловой системы.

Монтирование - это процесс, при котором операционная система связывает файловую систему (обычно представленную в виде устройства хранения данных, например, жесткого диска) с точкой монтирования в файловой системе операционной системы.

После успешного монтирования содержимое файловой системы становится доступным для чтения и записи через файловую систему операционной системы в месте, где была создана точка монтирования.

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

Основные причины нарушения целостности файловой системы могут включать в себя:

- Сбои питания или выключение компьютера без надлежащего выключения операционной системы.
- Ошибки в работе операционной системы или файловой системы, включая ошибки записи, чтения, перемещения или удаления файлов.
- Вредоносное программное обеспечение, такое как вирусы, троянские программы и шпионское программное обеспечение.
- Физические повреждения жесткого диска или другого устройства хранения данных.
- Недостаток свободного места на жестком диске, что может привести к фрагментации файловой системы и повреждению файлов.

Чтобы устранить повреждения файловой системы, можно попробовать следующие действия:

- Проверить жесткий диск на наличие повреждений с помощью утилиты проверки диска, которая может быть встроена в операционную систему.
- Использовать утилиту восстановления файловой системы, которая может быть встроена в операционную систему.
- Использовать сторонние утилиты, которые могут помочь восстановить поврежденные файлы и файловую систему.
- Переустановить операционную систему, если другие методы не помогли устраниить повреждения файловой системы.
- Регулярно создавать резервные копии важных данных, чтобы восстановить их в случае повреждения файловой системы.

5. Как создаётся файловая система?

Файловая система - это способ организации данных на компьютере или другом устройстве хранения информации. Создание файловой системы

включает в себя несколько шагов.

Выбор типа файловой системы: Первым шагом является выбор типа файловой системы, который будет использоваться для хранения данных. Наиболее распространенными типами файловых систем являются NTFS, FAT32, exFAT, HFS + и ext4, но их существует намного больше.

Разметка диска: После выбора типа файловой системы следует разметить диск или раздел диска, на котором будет создана файловая система. Разметка диска может быть выполнена с помощью специальных утилит, таких как Disk Management в Windows или Disk Utility в macOS.

Создание файловой системы: Создание файловой системы выполняется с помощью утилиты форматирования диска. Эта утилита создает корневую директорию файловой системы и устанавливает начальные параметры, такие как размер кластера, который определяет минимальный размер файла, и количество файловых дескрипторов, которые можно использовать.

Монтирование файловой системы: После создания файловой системы ее нужно примонтировать (подключить) к системе. Это позволяет операционной системе обращаться к файлам и директориям, которые хранятся на этой файловой системе.

Настройка параметров файловой системы: Последний шаг - настройка параметров файловой системы. Например, можно настроить различные параметры безопасности, такие как права доступа к файлам и директориям, а также установить ограничения на размер файлов или директорий.

6. Дайте характеристику командам для просмотра текстовых файлов.

1. Команда “cat” (Unix, Linux): Команда “cat” отображает содержимое текстового файла на экране. Если файл содержит больше строк, чем может поместиться на одном экране, то вывод будет продолжаться до тех пор, пока все строки не будут выведены.

2. Команда “more” (Unix, Linux): Команда “more” отображает содержимое текстового файла на экране постранично. Она выводит первую страницу файла и ожидает ввода пользователя, после чего выводит следующую страницу.
 3. Команда “less” (Unix, Linux): Команда “less” отображает содержимое текстового файла на экране постранично, как и “more”. Однако “less” имеет дополнительные функции, такие как поиск, прокрутка вверх и вниз, а также возможность просмотра больших файлов без загрузки всего содержимого в память.
 4. Команда “type” (Windows): Команда “type” отображает содержимое текстового файла на экране. Она работает аналогично команде “cat” в Unix и Linux.
 5. Команда “more” (Windows): Команда “more” отображает содержимое текстового файла на экране постранично, как и в Unix и Linux.
 6. Команда “notepad” (Windows): Команда “notepad” запускает стандартный текстовый редактор в Windows и открывает содержимое файла в нём.
 7. Команда “vim” (Unix, Linux): Команда “vim” запускает текстовый редактор Vim в Unix и Linux, который может использоваться для просмотра и редактирования текстовых файлов.
7. Приведите основные возможности команды cp в Linux.
1. Копирование файлов и директорий
 2. Копирование с сохранением атрибутов
 3. Копирование рекурсивно
 4. Копирование с перезаписью
 5. Копирование с использованием шаблонов
 6. Копирование с использованием буфера обмена

7. Отображение прогресса копирования
8. Приведите основные возможности команды mv в Linux
 1. Перемещение файла или папки в другую директорию
 2. Переименование файла или папки
 3. Перемещение и переименование файла или папки
 4. Перезапись файла в случае, если файл с таким именем уже существует
 5. Рекурсивное перемещение папки со всем ее содержимым
 6. Создание символьической ссылки на файл или папку вместо перемещения или копирования
 7. Отображение информации о перемещении файлов или папок
9. Что такое права доступа? Как они могут быть изменены?

Права доступа - это правила, которые определяют, какой доступ разрешен к определенному ресурсу или функции в компьютерной системе или сети. Эти права определяют, какие пользователи или группы пользователей имеют доступ к определенной информации, файлам или программам, а также какой уровень доступа им разрешен.

Права доступа могут быть изменены различными способами, в зависимости от системы. Например, в ОС Windows для изменения прав доступа к файлам и папкам можно использовать свойства файла или папки. В свойствах файлов и папок можно настроить, кто может просматривать, изменять, копировать или удалять файлы и папки. В Linux и других Unix-подобных системах, права доступа управляются через команду chmod. Команда chmod изменяет права доступа к файлам и папкам на основе числовых или символьных параметров.

5 Выводы

Выполняя данную лабораторную работу, я 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