Отчёт по лабораторной работе №6

дисциплина: Операционные системы

Быстров Глеб Андреевич

Содержание

1 Цель работы

В данной лабораторной работе мне будет необходимо ознакомиться с файловой системой Linux, её структурой, именами и содержанием каталогов. Будет необходимо преобрести практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

2 Теория

- Команды: Создание текстового файла - команда touch. Формат команды: touch имя-файла Просмотр небольших файлов - команда cat. Формат команды: cat имя-файла Просмотр больших файлов - команда less (постраничный просмотр файлов). Формат команды: less имя-файла Копирование файлов и каталогов - команда cp. Формат команды: cp [-опции] исходный_файл целевой_файл Команды mv и mvdir - перемещение и переименование файлов и каталогов. Формат команды mv: mv [-опции] старый_файл новый_файл
- Права доступа: Чтение r (разрешены просмотр и копирование) Запись w (разрешены изменение и переименование) Выполнение x (разрешено выполнение файла (скриптов и/или программ)

3 Задание

Провести работу в домашнем каталоге согласно инструкции. Использовать команды для создания, изменения и перемещения файлов.

4 Выполнение лабораторной работы

1. Выполнил все примеры, приведённые в первой части описания лабораторной работы. (рис. 4.1) (рис. 4.2) (рис. 4.3) (рис. 4.4) (рис. 4.5) (рис. 4.6) (рис. 4.7) (рис. 4.8) (рис. 4.9) (рис. 4.10) (рис. 4.11) (рис. 4.12) (рис. 4.13) (рис. 4.14)

Figure 4.1: Скопировал файл ~/abc1 в файл april и в файл may

Figure 4.2: Скопировал файлы april и may в каталог monthly

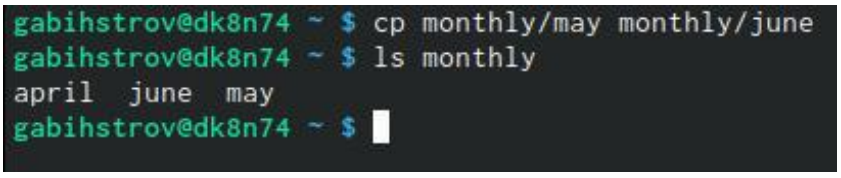

Figure 4.3: Скопировал файл monthly/may в файл с именем june

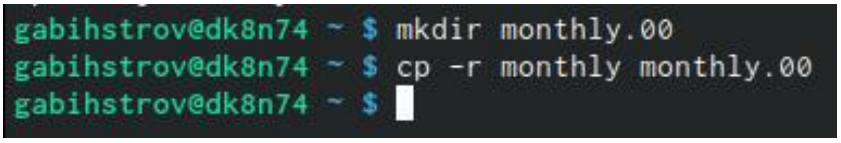

Figure 4.4: Скопировал каталог monthly в каталог monthly.00

Figure 4.5: Скопировал каталог monthly.00 в каталог /tmp

Figure 4.6: Изменил название файла april на july

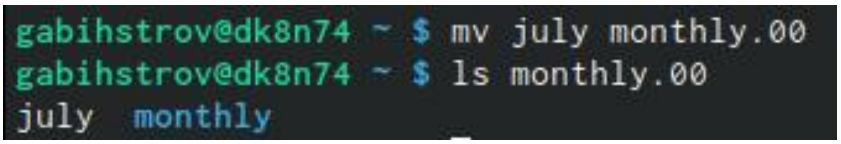

Figure 4.7: Переместить файл july в каталог monthly.00

Figure 4.8: Переименовал каталог monthly.00 в monthly.01

Figure 4.9: Переместил каталог monthly.01 в каталог reports

Figure 4.10: Переименовал каталог reports/monthly.01

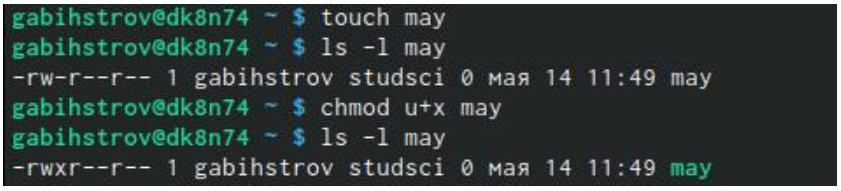

Figure 4.11: Создал файл ~/may с правом выполнения для владельца

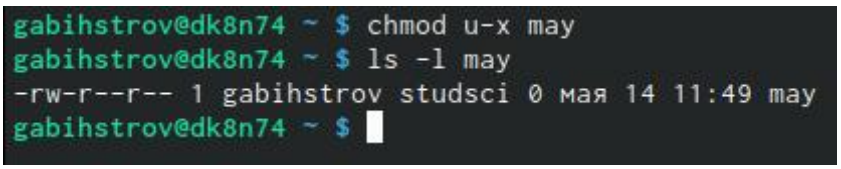

Figure 4.12: Лишил владельца файла ~/may права на выполнение

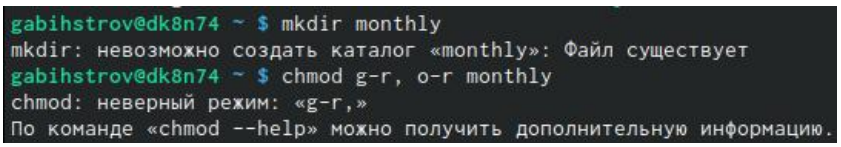

Figure 4.13: Создал каталог monthly с запретом на чтение

Figure 4.14: Создал файл ~/abc1 с правом записи для членов группы

2. Скопировал файл /usr/include/sys/io.h в домашний каталог и назвал его equipment. Использовал команды cp и mv. (рис. 4.15)

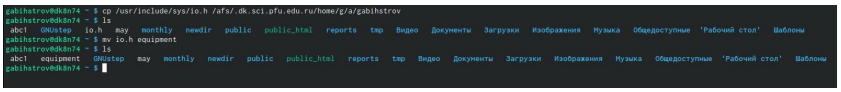

Figure 4.15: Работа с файлом io.h

3. В домашнем каталоге создайте директорию ~/ski.plases. Использовал команду mkdir. (рис. 4.16)

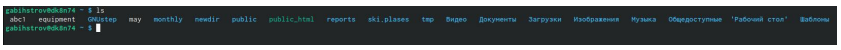

Figure 4.16: Создал директорию ~/ski.plases

4. Переместил файл equipment в директорию ~/ski.plases. Использовал команду mv. (рис. 4.17)

```
gabihstrov@dk8n74 ~ $ mv equipment ski.plases
gabihstrov@dk8n74 ~ $ cd ski.plases
gabihstrov@dk8n74 ~/ski.plases $ ls
equipment
gabihstrov@dk8n74 ~/ski.plases $
```

Figure 4.17: Переместил файл equipment

5. Переименовал файл equipment в equiplist. Использовал команду mv. (рис. 4.18)

```
gabihstrov@dk8n74 ~/ski.plases $ mv equipment equiplist
gabihstrov@dk8n74 ~/ski.plases $ ls
equiplist
gabihstrov@dk8n74 ~/ski.plases $
```

Figure 4.18: Переименовал файл в equiplist

6. Создал в домашнем каталоге файл abc1 и скопировал его в каталог ~/ski.plases, назвал его equiplist2. Использовал команды touch и cp. (рис. 4.19)

```
gabihstrov@dk8n74 ~ $ touch abc1
gabihstrov@dk8n74 ~ $ cp -r abc1 ski.plases/equiplist2
gabihstrov@dk8n74 ~ $ cd ski.plases
gabihstrov@dk8n74 ~/ski.plases $ ls
equiplist  equiplist2
gabihstrov@dk8n74 ~/ski.plases $
```

Figure 4.19: Работа с файлом abc1

7. Создал каталог с именем equipment в каталоге ~/ski.plases. Использовал команду mkdir. (рис. 4.20)

```
gabihstrov@dk8n74 ~/ski.plases $ mkdir equipment
gabihstrov@dk8n74 ~/ski.plases $
```

Figure 4.20: Создание каталога equipment

8. Переместил файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment. Использовал команду mv. (рис. 4.21)

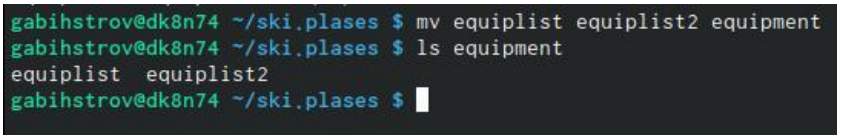

Figure 4.21: Перемещение файлов

9. Создал и переместил каталог ~/newdir в каталог ~/ski.plases и назвал его plans. Использовал команды mkdir и mv. (рис. 4.22)

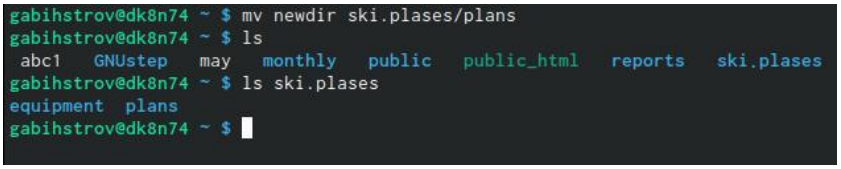

Figure 4.22: Работа с каталогом ~/newdir

10. Определил опции команды chmod для файла australia. (рис. 4.23)

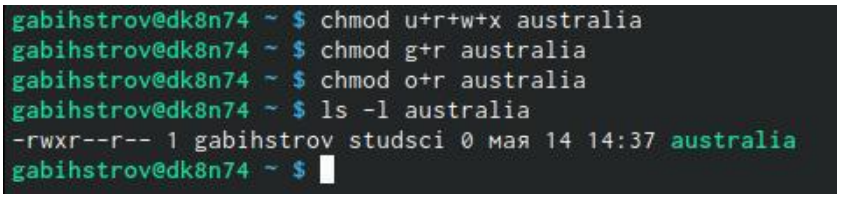

Figure 4.23: chmod для файла australia

11. Определил опции команды chmod для файла play. (рис. 4.24)

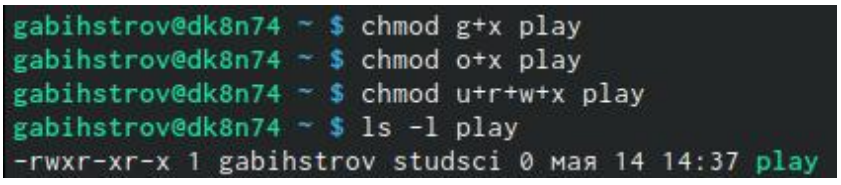

Figure 4.24: chmod для файла play

12. Определил опции команды chmod для файла my_os. (рис. 4.25)

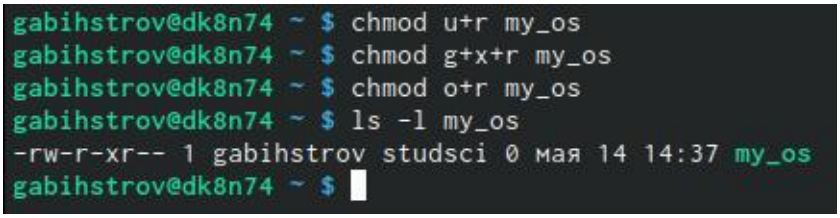

Figure 4.25: chmod для файла my_os

13. Определил опции команды chmod для файла feathers. (рис. 4.26)

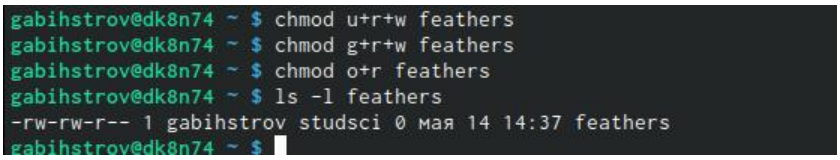

Figure 4.26: chmod для файла feathers

14. Просмотрел содержимое файла /etc/password. (рис. 4.27)

Figure 4.27: Содержимое файла /etc/password

15. Скопировал файл ~/feathers в файл ~/file.old. (рис. 4.28)

Figure 4.28: Скопировал файл ~/feathers

16. Переместил файл ~/file.old в каталог ~/play. (рис. 4.29)

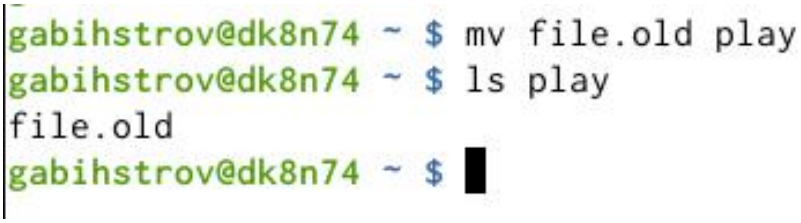

Figure 4.29: еместил файл ~/file.old

17. Скопировал каталог ~/play в каталог ~/fun. (рис. 4.30)

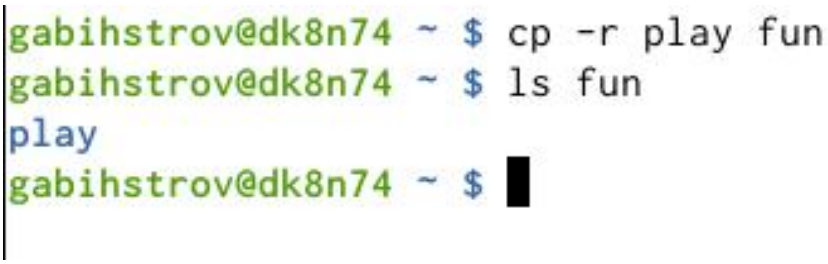

Figure 4.30: Скопировал каталог ~/play

18. Переместил каталог ~/fun в каталог ~/play и назвал его games. (рис. 4.31)

Figure 4.31: Переместил каталог ~/fun

19. Лишил владельца файла ~/feathers права на чтение. Попытался просмотреть файл ~/feathers командой cat. Было отказано в доступе. Попытался скопировать файл ~/feathers. Было отказано в доступе. (рис. 4.32)

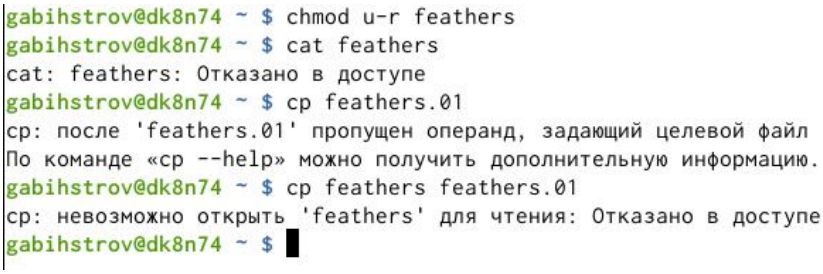

Figure 4.32: Работа с файлом ~/feathers

22. Дал владельцу файла ~/feathers права на чтение. (рис. 4.33)

Figure 4.33: Дал права на чтение

23. Лишил владельца каталога ~/play права на выполнение. Перешёл в каталог ~/play. Было отказано в доступе. Дал владельцу каталога ~/play право на выполнение. (рис. 4.34)

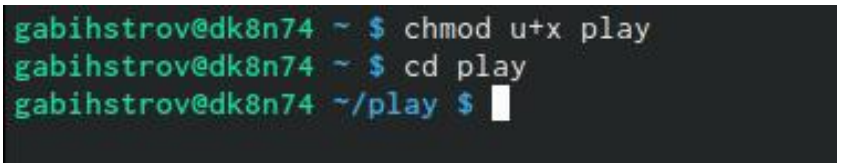

Figure 4.34: Работа с каталогом ~/play

26. Прочитал man по командам mount, fsck, mkfs, kill и кратко их охарактеризовал, приведя примеры. (рис. 4.35) (рис. 4.36) (рис. 4.37) (рис. 4.38)

- mount $ mount файл_устройства папка_назначения Основные опции: -V - вывести версию утилиты; -h - вывести справку; -v - подробный режим;
- fsck $ fsck [опции] [опции_файловой_системы] [раздел_диска] опции fsck: -C - показать прогресс проверки файловой системы; -M - не проверять, если файловая система смонтирована; -N - ничего не выполнять, показать, что проверка завершена успешно; -R - не проверять корневую файловую систему; -T - не показывать информацию об утилите; -V - максимально подробный вывод.
- mkfs $ mkfs -t -V - подробно информирует происходящее, включая все выполняемые специфические команды файловой системы. Если указать этот параметр более одного раза, то это запретит реальное выполнение любых специфических команд файловой системы. Использовать этот параметр

целесообразно во время тестирования. -t fstype - указывает тип создаваемой файловой системы. Если этот параметр не указан, тогда, по умолчанию, принимается тип файловой системы ext2. fs-options - передаёт модулю создания специфической файловой системы параметры в виде списка. Следует отметить, что нет гарантии в том, что следующие перечисленные параметры будут поддерживаться большинством модулей создания файловых систем. -c - перед созданием файловой системы проверяет наличие сбойных блоков на устройстве. -l filename - считывает список сбойных блоков из указанного файла filename. Для составления подобного списка, можно выполнить предварительную проверку, например, с помощью программы badblocks. -v - подробно комментирует происходящее.

- kill kill опции PID Чтобы “убить” процесс с указанным PID, нужно выполнить следующую команду: # kill -9 1684 Можно использовать не только номер, но и название сигнала/ Например, для запроса на прерывание, который можно обработать или проигнорировать: # kill -SIGTERM 1684 Аналогично, команда kill -SIGKILL PID эквивалентна команде # kill -9 PID Для прерывания нескольких процессов можно указать их идентификаторы через пробел: # kill PID1 PID2 PID3 или # kill -9 PID1 PID2 PID3 или # kill -SIGKILL PID1 PID2 PID3

Figure 4.35: Прочитал man по команде mount

Figure 4.36: Прочитал man по команде fsck

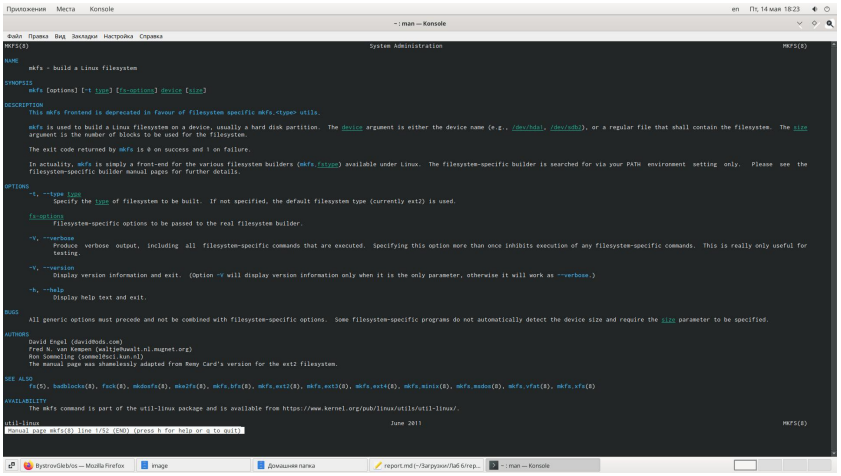

Figure 4.37: Прочитал man по команде mkfs

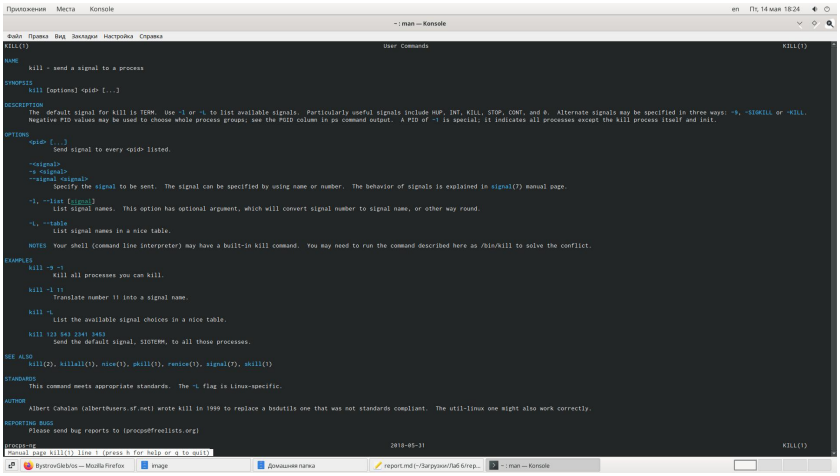

Figure 4.38: Прочитал man по команде kill

5 Контрольные вопросы

1. Дайте характеристику каждой файловой системе, существующей на жёстком диске компьютера, на котором вы выполняли лабораторную работу. Библиографический список (ссылка №5). Каждый дистрибутив Linux позволяет использовать одну из этих файловых систем, каждая из них имеет свои преимущества и недостатки. Все они включены в ядро и могут использоваться в качестве корневой файловой системы. Давайте рассмотрим каждую из них более подробно.

- Ext2, Ext3, Ext4 или Extended Filesystem - это стандартная файловая система для Linux. Она была разработана еще для Minix. Она самая стабильная из всех существующих, кодовая база изменяется очень редко и эта файловая система содержит больше всего функций. Версия ext2 была разработана уже именно для Linux и получила много улучшений. В 2001 году вышла ext3, которая добавила еще больше стабильности благодаря использованию журналирования. В 2006 была выпущена версия ext4, которая используется во всех дистрибутивах Linux до сегодняшнего дня. В ней было внесено много улучшений, в том числе увеличен максимальный размер раздела до одного экзабайта.
- JFS или Journaled File System была разработана в IBM для AIX UNIX и использовалась в качестве альтернативы для файловых систем ext. Сейчас она используется там, где необходима высокая стабильность и минимальное потребление ресурсов. При разработке файловой системы ставилась цель создать максимально эффективную файловую систему для многопро-

цессорных компьютеров. Также как и ext, это журналируемая файловая система, но в журнале хранятся только метаданные, что может привести к использованию старых версий файлов после сбоев.

- ReiserFS - была разработана намного позже, в качестве альтернативы ext3 с улучшенной производительностью и расширенными возможностями. Она была разработана под руководством Ганса Райзера и поддерживает только Linux. Из особенностей можно отметить динамический размер блока, что позволяет упаковывать несколько небольших файлов в один блок, что предотвращает фрагментацию и улучшает работу с небольшими файлами. Еще одно преимущество - в возможности изменять размеры разделов на лету. Но минус в некоторой нестабильности и риске потери данных при отключении энергии. Раньше ReiserFS применялась по умолчанию в SUSE Linux, но сейчас разработчики перешли на Btrfs.
- XFS - это высокопроизводительная файловая система, разработанная в Silicon Graphics для собственной операционной системы еще в 2001 году. Она изначально была рассчитана на файлы большого размера, и поддерживала диски до 2 Терабайт. Из преимуществ файловой системы можно отметить высокую скорость работы с большими файлами, отложенное выделение места, увеличение разделов на лету и незначительный размер служебной информации.
- XFS - журналируемая файловая система, однако в отличие от ext, в журнал записываются только изменения метаданных. Она используется по умолчанию в дистрибутивах на основе Red Hat. Из недостатков - это невозможность уменьшения размера, сложность восстановления данных и риск потери файлов при записи, если будет неожиданное отключение питания, поскольку большинство данных находится в памяти.
- Btrfs или B-Tree File System - это совершенно новая файловая система, которая сосредоточена на отказоустойчивости, легкости администрирования и восстановления данных. Файловая система объединяет в себе очень много

новых интересных возможностей, таких как размещение на нескольких разделах, поддержка подтомов, изменение размера не лету, создание мгновенных снимков, а также высокая производительность. Но многими пользователями файловая система Btrfs считается нестабильной. Тем не менее, она уже используется как файловая система по умолчанию в OpenSUSE и SUSE Linux.

2. Приведите общую структуру файловой системы и дайте характеристику каждой директории первого уровня этой структуры. Библиографический список (ссылка №4). Система хранения данных на дисках может быть структурирована следующим образом.

Нижний уровень - оборудование. Это в первую очередь, магнитные диски с подвижными головками - основные устройства внешней памяти, представляющие собой пакеты магнитных пластин (поверхностей), между которыми на одном рычаге двигается пакет магнитных головок. Шаг движения пакета головок является дискретным и каждому положению пакета головок логически соответствует цилиндр магнитного диска. Цилиндры делятся на дорожки (треки), а каждая дорожка размечается на одно и то же количество блоков (секторов), таким образом, что в каждый блок можно записать по максимуму одно и то же число байтов. Следовательно, для произведения обмена с магнитным диском на уровне аппаратуры нужно указать номер цилиндра, номер поверхности, номер блока на соответствующей дорожке и число байтов, которое нужно записать или прочитать от начала этого блока. Таким образом, диски могут быть разбиты на блоки фиксированного размера, и можно непосредственно получить доступ к любому блоку (организовать прямой доступ к файлам).

Непосредственно с устройствами (дисками) взаимодействует часть ОС, называемая система ввода-вывода (см. соответствующую главу). Система ввода-вывода (она состоит из драйверов устройств и обработчиков прерываний для передачи информации между памятью и дисковой системой) предоставляет в распоряжение более высокоуровневого компонента ОС - файловой системы используемое

дисковое пространство в виде непрерывной последовательности блоков фиксированного размера. Система ввода-вывода имеет дело с физическими блоками диска, которые характеризуются адресом, например, диск 2, цилиндр 75, сектор 11. Файловая система имеет дело с логическими блоками, каждый из которых имеет номер (от 0 или 1 до N). Размер этих логических блоков файла совпадает или кратен размеру физического блока диска и может быть задан равным размеру страницы виртуальной памяти, поддерживаемой аппаратурой компьютера совместно с операционной системой.

В структуре системы управления файлами можно выделить базисную подсистему, которая отвечает за выделение дискового пространства конкретным файлам, и более высокоуровневую логическую подсистему, которая использует структуру дерева директорий для предоставления модулю базисной подсистемы необходимой ей информации исходя из символического имени файла. Она также ответственна за авторизацию доступа к файлам.

3. Какая операция должна быть выполнена, чтобы содержимое некоторой файловой системы было доступно операционной системе? Для просмотра используемых в операционной системе файловых систем можно воспользоваться командой mount без параметров. В контексте команды mount устройство — специальный файл устройства, с помощью которого операционная система получает доступ к аппаратному устройству.

4. Назовите основные причины нарушения целостности файловой системы. Как устранить повреждения файловой системы? Библиографический список (ссылка №3). Важный аспект надежной работы файловой системы - контроль ее целостности. В результате файловых операций блоки диска могут считываться в память, модифицироваться и затем записываться на диск. Причем многие файловые операции затрагивают сразу несколько объектов файловой системы. Например, копирование файла предполагает выделение ему блоков диска, формирование индексного узла, изменение содержимого каталога и т. д. В течение короткого периода времени между

этими шагами информация в файловой системе оказывается несогласованной. И если вследствие непредсказуемой остановки системы на диске будут сохранены изменения только для части этих объектов (нарушена атомарность файловой операции), файловая система на диске может быть оставлена в несовместимом состоянии. В результате могут возникнуть нарушения логики работы с данными, например появиться “потерянные” блоки диска, которые не принадлежат ни одному файлу и в то же время помечены как занятые, или, наоборот, блоки, помеченные как свободные, но в то же время занятые (на них есть ссылка в индексном узле) или другие нарушения. В современных ОС предусмотрены меры, которые позволяют свести к минимуму ущерб от порчи файловой системы и затем полностью или частично восстановить ее целостность. Очевидно, что для правильного функционирования файловой системы значимость отдельных данных неравноценна. Искажение содержимого пользовательских файлов не приводит к серьезным (с точки зрения целостности файловой системы) последствиям, тогда как несоответствия в файлах, содержащих управляющую информацию (директории, индексные узлы, суперблок и т. п.), могут быть катастрофическими. Поэтому должен быть тщательно продуман порядок выполнения операций со структурами данных файловой системы. Рассмотрим пример создания жесткой связи для файла. Для этого файловой системе необходимо выполнить следующие операции:

- создать новую запись в каталоге, указывающую на индексный узел файла;
- увеличить счетчик связей в индексном узле. Если аварийный останов произошел между 1-й и 2-й операциями, то в каталогах файловой системы будут существовать два имени файла, адресующих индексный узел со значением счетчика связей, равному 1. Если теперь будет удалено одно из имен, это приведет к удалению файла как такового. Если же порядок операций изменен и, как прежде, останов произошел между первой и второй операциями, файл будет иметь несуществующую жесткую связь, но существующая

запись в каталоге будет правильной. Хотя это тоже является ошибкой, но ее последствия менее серьезны, чем в предыдущем случае.

5. Как создаётся файловая система? Библиографический список (ссылка №2). Создать файловую систему можно только в процессе форматирования. Создание файловой системы на жестком осуществляется непосредственно во время установки программного обеспечения. Как правило, форматирование диска производится на одном из этапов инсталляции, в котором пользователю предлагается выбрать диск, на который будет остановлена операционная система. Если данный диск еще никогда не использовался, то создать файловую систему на нем будет просто необходимо. Для ее создания при установке Windows выберите команду “Форматировать”, отрегулируйте объем диска и нажмите “Старт”. При необходимости жесткий диск можно разбить на несколько частей. Создать файловую систему на жестком диске также можно с помощью стандартных средств Windows. Для этого нужно зайти в папку “Мой компьютер” и нажать правой кнопкой мыши на диск, который необходимо отформатировать. При форматировании жесткого диска из-под Windows выберите необходимые опции, такие как тип файловой системы, которая должна быть создана после форматирования, размер кластера будущего логического диска и название тома, которое будет отображаться в программе “Проводник”. Помимо этого в опциях можно выбрать вид форматирования, которое бывает быстрым, в результате которого диск становится пригодным для записи новых данных, но физически не форматируется, и стандартным, в итоге которого все данные стираются. Помимо вышеперечисленных способов, создать файловую систему диска можно с помощью специального софта, работающего со структурой жесткого диска, например Acronic Disk Director Suite, а также специальными средствами Windows. Специальный инструмент управления дисками включается следующим образом: “Пуск” - “Мой компьютер” (правая кнопка мыши) - “Управление” - “Управление дисками”.

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

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 Копирование файла в текущем каталоге. Копирование нескольких файлов в каталог. Копирование файлов в произвольном каталоге. Опция i в команде cp выведет на экран запрос подтверждения о перезаписи файла, если на место целевого файла вы поставите имя уже существующего файла. 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. Библиографический список (ссылка №1). Команды mv и mvdir предназначены для перемещения и переименования файлов и каталогов. Формат команды mv: mv [-опции] старый_файл новый_файл Если необходим запрос подтверждения о перезаписи файла, то нужно использовать опцию i. mvdir - перемещение каталога /etc/mvdir исходный_каталог целевой_каталог Команда mvdir перемещает каталоги в пределах файловой системы. Если целевой_каталог не существует, он создается; в противном случае создается каталог с именем целевой_каталог/исходный_каталог. Исходный и целевой каталоги не могут лежать на одном маршруте, то есть ни один из них не может быть прямым или косвенным подкаталогом другого.

9. Что такое права доступа? Как они могут быть изменены? Каждый файл или каталог имеет права доступа. Права доступа к файлу или каталогу можно

изменить, воспользовавшись командой chmod. Сделать это может владелец файла (или каталога) или пользователь с правами администратора. Формат команды: chmod режим имя_файла

- Чтение r (разрешены просмотр и копирование)
- Запись w (разрешены изменение и переименование)
- Выполнение x (разрешено выполнение файла (скриптов и/или программ)

6 Выводы

В данной лабораторной работе мне успешно удалось ознакомиться с файловой системой Linux, её структурой, именами и содержанием каталогов. Получилось преобрести практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

7 Библиографический список

1. Команда mvdir (http://www.linuxlib.ru/manpages/MVDIR.1.shtml)
2. Создание файловой системы (https://www.kakprosto.ru/kak-48770-kak-sozdat-faylovuyu-sistemu#ixzz6up4lHV3x)
3. Нарушение целостности файловых систем (https://it.wikireading.ru/6624)
4. Общая структура файловой системы (https://studopedia.ru/15_909_obshchaya-struktura-faylovoy-sistemi.html)
5. Типы файловых систем для Linux (https://losst.ru/tipy-fajlovyh-sistem-dlya-linux)