# Титульный лист

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

**Лабораторная работа 6**

По дисциплине "Операционные системы"

Выполнил:

Студент группы НПВбм-01-19

Студенческий билет №: 1032193844

Саидов Ахият Магомадович

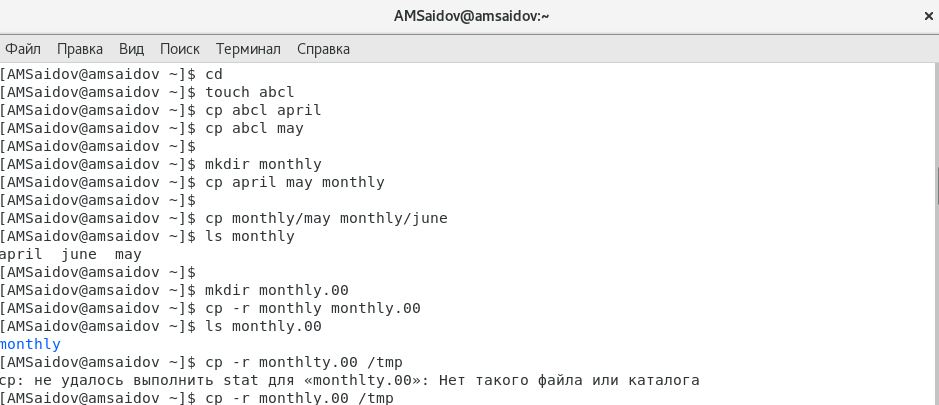
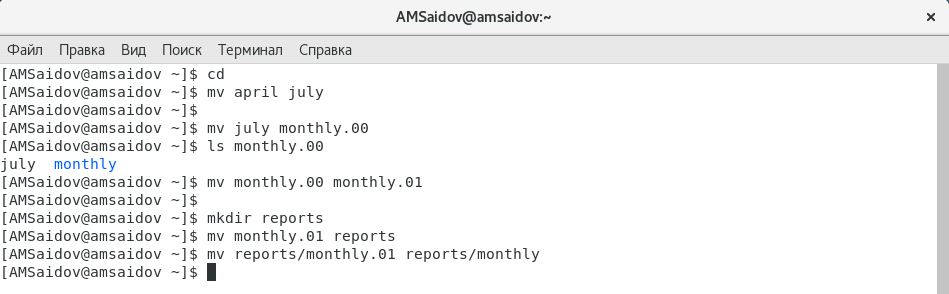
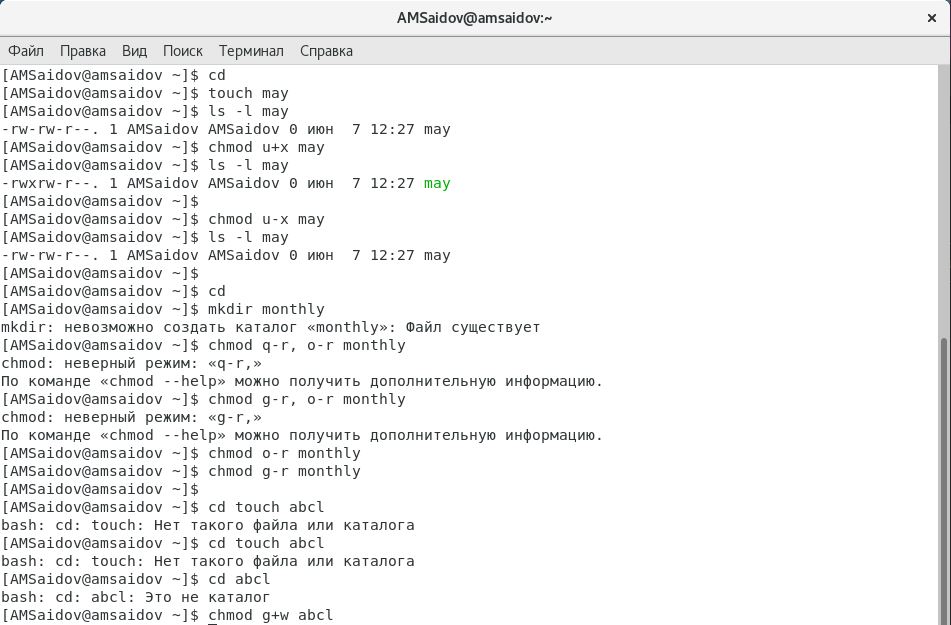
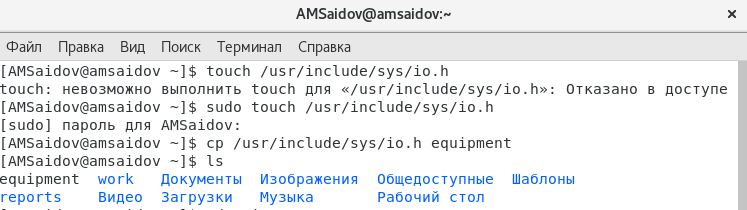
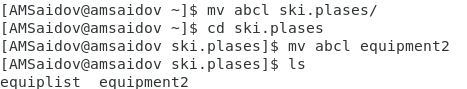
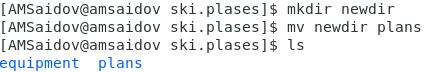
Руководитель: Валиева Татьяна Рефатовна

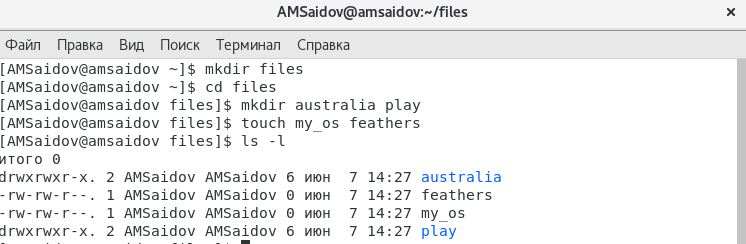
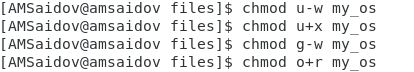
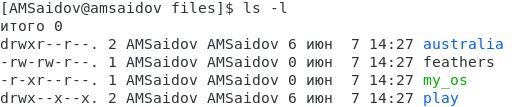
Москва 2023

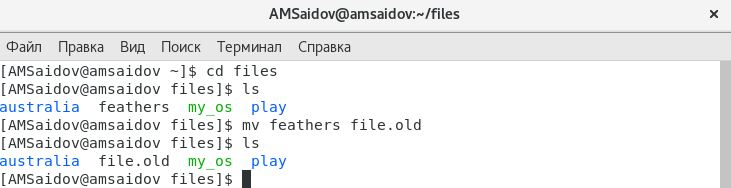
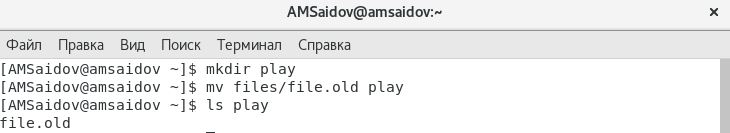
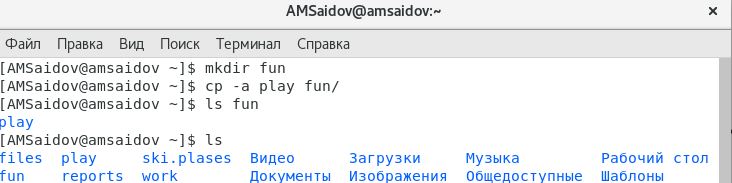
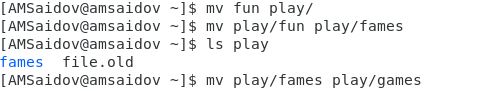
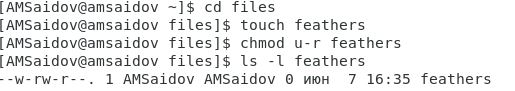
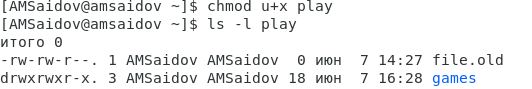
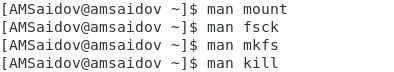
# Цель работы

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.

# Начало работы

1. Я выполнил все примеры, приведенные в первой части описания лабораторной работы.
   * Пример ЛБ 3.2.2.
   * 
   * *Рисунок 1*
   * Пример ЛБ 3.2.3.
   * 
   * *Рисунок 2*
   * Пример ЛБ 3.2.5.
   * 
   * *Рисунок 3*
2. Выполним следующие действия, зафиксировав в отчете по лабораторной работе, используемые при этом команды и результаты их выполнения:
   * Скопируем файл */usr/include/sys/io.h* в домашний каталог и назовет его *equipment*.
   * 
   * *Рисунок 4*
   * В домашнем каталоге создадим директорию */ski.plases*.
   * fig:
   * *Рисунок 5*
   * Переместим файл *equipment* в каталог */ski.plases*
   * fig:
   * *Рисунок 6*
   * Переименуем файл */ski.plases/equipment* в */ski.plases/equiplist*
   * fig:
   * *Рисунок 7*
   * Создадим в домашнем каталоге файл abcl
   * fig:
   * *Рисунок 8*
   * и скопируем его в каталог */ski.plases*, назовем его *equipment2*
   * 
   * *Рисунок 9*
   * Создадим каталог с именем *equipment* в каталоге */ski.plases*
   * fig:
   * *Рисунок 10*
   * Переместим файлы */ski.plases/equiplist* и */ski.plases/equiplist2* в каталог */ski.plases/equipment*
   * fig:
   * *Рисунок 11*
   * Проверим:
   * fig:
   * *Рисунок 12*
   * Создадим и переместим каталог */newdir* и назовем его *plans*
   * 
   * *Рисунок 13*
3. Определим опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет.

* 
* *Рисунок 14*
  + fig:
  + *Рисунок 15*
  + fig:
  + *Рисунок 16*
  + 
  + *Рисунок 17*
  + Каталог *feathers* оставляем без изменений.
  + Итог: 
  + *Рисунок 18*

1. Проделаем указанные в ЛР упражнения, записывая в отчет по ЛР, используемые при этом команды.
   * Посмотрим содержимое файла \_/etc/passwd.
   * Итог: 
   * *Рисунок 19*
   * Скопируем файл */feathers* в файл */file.old*.
   * Итог: 
   * *Рисунок 20*
   * Переместим файл */file.old* в каталог */play*.
   * Итог: 
   * *Рисунок 21*
   * Скопируем каталог */play* в каталог */fun*.
   * Итог: 
   * *Рисунок 22*
   * Переместим каталог */fun* в каталог */play* и назовите его *games*.
   * Итог: 
   * *Рисунок 23*
   * Лишим владельца файла */feathers* права на чтение.
   * Итог: 
   * *Рисунок 24*
   * Что произойдёт, если вы попытаетесь просмотреть файл *~/feathers* командой cat?
   * Итог: fig:
   * *Рисунок 25*
   * Что произойдёт, если вы попытаетесь скопировать файл */feathers*?
   * Итог: fig:
   * *Рисунок 26*
   * Дадим владельцу файла */feathers* право на чтение.
   * Итог: fig:
   * *Рисунок 27*
   * Лишим владельца каталога */play* права на выполнение.
   * Итог: 
   * *Рисунок 28*
   * Перейдем в каталог */play*. Что произошло?
   * Итог: fig:
   * *Рисунок 29*
   * Дадим владельцу каталога */play* право на выполнение.
   * Итог: 
   * *Рисунок 30*
2. Итог: 

* *Рисунок 31*
  + Смонтировать файловую систему man mount.
  + Проверка и восстановление файловой системы Linux man fsck.
  + Построить файловую систему man mkfs.
  + Завершить процесс man kill.

# Вывод

Мы ознакомились с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретели 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

* + **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 многопроцессорных 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.
  + **/ — root** каталог. Содержит в себе всю иерархию системы;
  + **/bin** — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: pwd, ls, cat, ps);
  + **/boot** — тут расположены файлы, используемые для загрузки системы (образ initrd, ядро vmlinuz);
  + **/dev** — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;
  + **/etc** — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;
  + **/home** — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;
  + **/lib** — содержит системные библиотеки, с которыми работают программы и модули ядра;
  + **/lost+found** — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;
  + **/media** — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;
  + **/mnt** — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;
  + **/opt** — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);
  + **/proc** — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;
  + **/root** — директория, которая содержит файлы и личные настройки суперпользователя;
  + **/run** — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;
  + **/sbin** — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;
  + **/srv** — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);
  + **/sys** — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;
  + **/tmp** — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;
  + **/usr** — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой;
  + **/var** — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.

1. Монтирование тома.
2. Отсутствие синхронизации между образом файловой системы в памяти и ее данными на диске в случае аварийного останова может привести к появлению следующих ошибок:
   * Один блок адресуется несколькими mode (принадлежит нескольким файлам).
   * Блок помечен как свободный, но в то же время занят (на него ссылается onode).
   * Блок помечен как занятый, но в то же время свободен (ни один inode на него не ссылается).
   * Неправильное число ссылок в inode (недостаток или избыток ссылающихся записей в каталогах).
   * Несовпадение между размером файла и суммарным размером адресуемых inode блоков.
   * Недопустимые адресуемые блоки (например, расположенные за пределами файловой системы).
   * "Потерянные" файлы (правильные inode, на которые не ссылаются записи каталогов).
   * Недопустимые или неразмещенные номера inode в записях каталогов.
3. mkfs - позволяет создать файловую систему Linux.
4. cat - выводит содержимое файла на стандартное устройство вывода
5. cp – копирует или перемещает директорию, файлы.
6. mv - переименовать или переместить файл или директорию
7. Права доступа к файлу или каталогу можно изменить, воспользовавшись командой chmod. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.