Отчёт по лабораторной работе №6

Дисциплина: Операционные системы

Аветисян Давид Артурович

Содержание

# 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.

# Задание

1. Выполните все примеры, приведённые в первой части описания лабораторной работы.
2. Выполните следующие действия, зафиксировав в отчёте по лабораторной работе используемые при этом команды и результаты их выполнения:
   1. 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.
   2. В домашнем каталоге создайте директорию ~/ski.plases.
   3. Переместите файл equipment в каталог ~/ski.plases.
   4. Переименуйте файл ~/ski.plases/equipment в ~/ski.plases/equiplist.
   5. Создайте в домашнем каталоге файл abc1 и скопируйте его в каталог ~/ski.plases, назовите его equiplist2.
   6. Создайте каталог с именем equipment в каталоге ~/ski.plases.
   7. Переместите файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment.
   8. Создайте и переместите каталог ~/newdir в каталог ~/ski.plases и назовите его plans.
3. Определите опции команды chmod, необходимые для того, чтобы присвоить перечисленным ниже файлам выделенные права доступа, считая, что в начале таких прав нет:
4. drwxr–r– … australia
5. drwx–x–x … play
6. -r-xr–r– … my\_os
7. -rw-rw-r– … feathers  
   При необходимости создайте нужные файлы.
8. Проделайте приведённые ниже упражнения, записывая в отчёт по лабораторной работе используемые при этом команды:
   1. Просмотрите содержимое файла /etc/password.
   2. Скопируйте файл ~/feathers в файл ~/file.old.
   3. Переместите файл ~/file.old в каталог ~/play.
   4. Скопируйте каталог ~/play в каталог ~/fun.
   5. Переместите каталог ~/fun в каталог ~/play и назовите его games.
   6. Лишите владельца файла ~/feathers права на чтение.
   7. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?
   8. Что произойдёт, если вы попытаетесь скопировать файл ~/feathers?
   9. Дайте владельцу файла ~/feathers право на чтение.
   10. Лишите владельца каталога ~/play права на выполнение.
   11. Перейдите в каталог ~/play. Что произошло?
   12. Дайте владельцу каталога ~/play право на выполнение.
9. Прочитайте man по командам mount, fsck, mkfs, kill и кратко их охарактеризуйте, приведя примеры.

# Выполнение лабораторной работы

1. Для начала выполним примеры, описанные в первой части описания лабораторной работы (рис. -fig. 1):
   1. Скопируем файл ~/abc1 в файл april и в файл may. Для этого создадим файл abc1, используя команду «touch abc1», далее осуществим копирование с помощью команд «cp abc1 april» и «cp abc1 may».
   2. Скопируем файлы april и may в каталог monthly, используя команды «mkdir monthly» − для создания каталога monthly и «cp april may monthly» − для копирования.
   3. Скопируем файл monthly/may в файл с именем june. Выполним команды «cp monthly/may monthly/june» и «ls monthly» (для просмотра содержимого каталога).
   4. Скопируем каталог monthly в каталог monthly.00. Для этого создадим каталог monthly.00 командой «mkdir monthly.00» и осуществим копирование, используя команду «cp -r monthly monthly.00» (команда cp с опцией r (recursive) позволяет копировать каталоги вместе с входящими в них файлами и каталогами).
   5. Скопируем каталог monthly.00 в каталог /tmp, используя команду «cp -r monthly.00 /tmp».

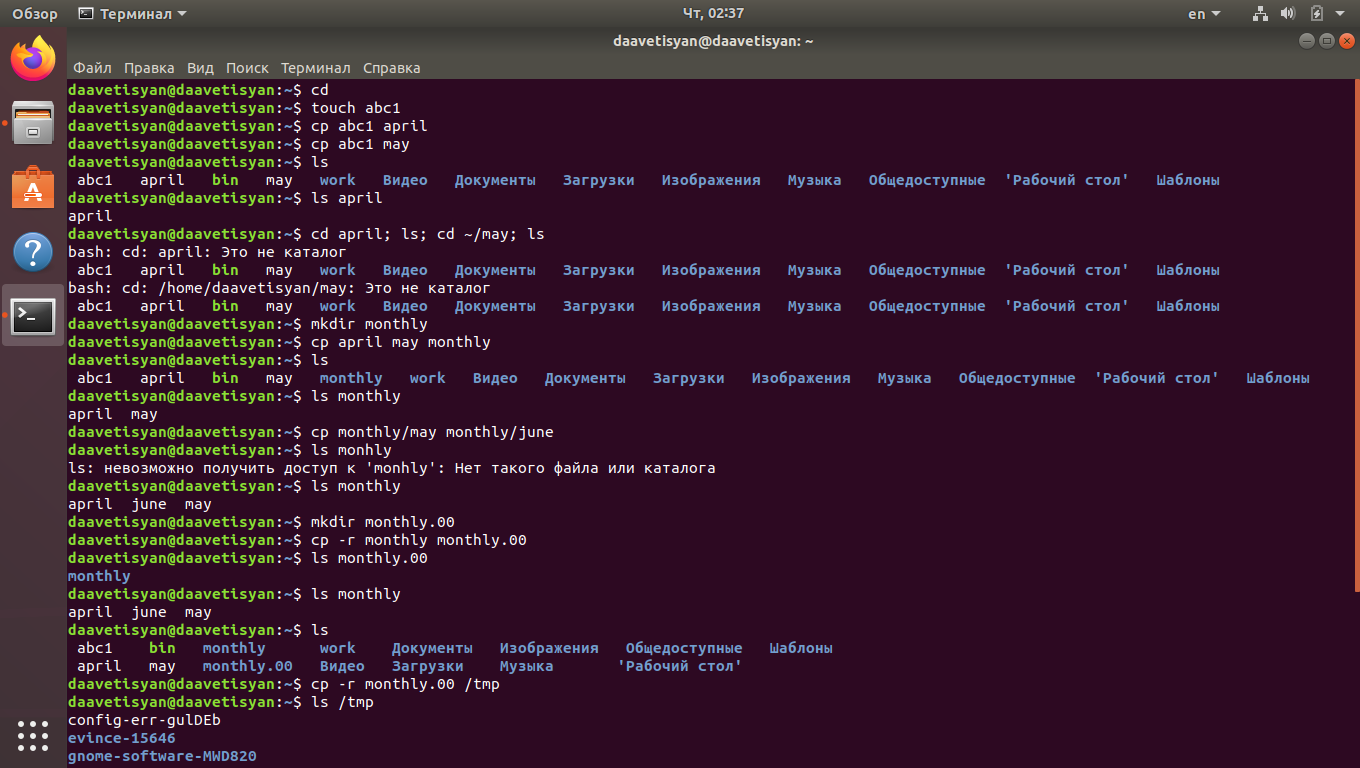


Figure 1: Выполним примеры, описанные в лабораторной работе

(рис. -fig. 2):  
1. Изменим название файла april на july в домашнем каталоге, используя команду «mv april july». 2. Переместим файл july в каталог monthly.00 с помощью команды «mv july monthly.00». Проверим результат командой «ls monthly.00». 3. Переименуем каталог monthly.00 в monthly.01, используя команду «mv monthly.00 monthly.01». 4. Переместим каталог monthly.01 в каталог reports. Для этого создадим каталог reports с помощью команды «mkdir reports» и выполним перемещение командой «mv monthly.01 reports». 5. Переименуем каталог reports/monthly.01 в reports/monthly командой «mv reports/monthly.01 reports/monthly».

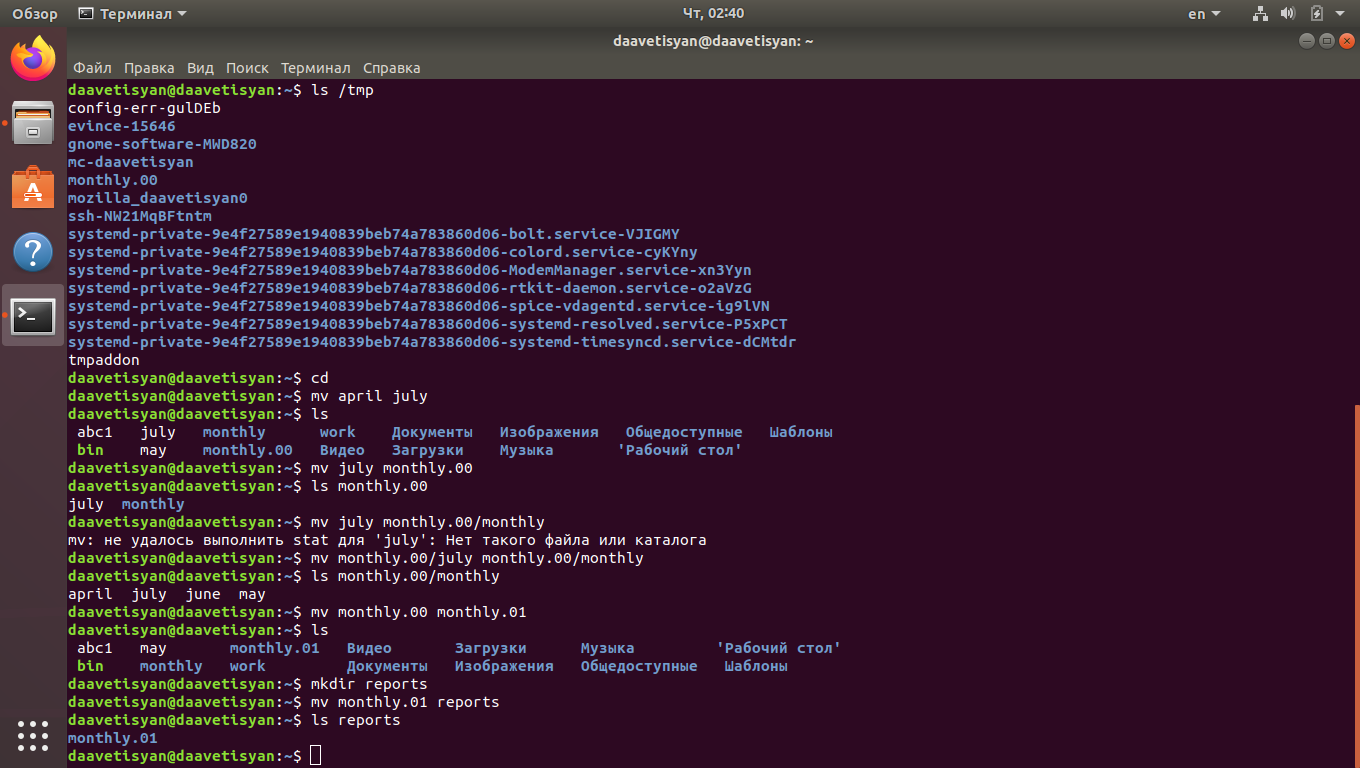


Figure 2: Выполним примеры, описанные в лабораторной работе

(рис. -fig. 3) (рис. -fig. 4):  
1. Создадим файл ~/may с правом выполнения для владельца. Для этого выполним следующие команды: «touch may» (создание файла), «ls -l may» (просмотр сведений о файле), «chmod u+x may» (изменение прав), «ls -l may». 2. Лишаем владельца файла ~/may права на выполнение, используя команды: «chmod u-x may» (изменение прав), «ls -l may» (просмотр сведений о файле). 3. Создаем каталог monthly с запретом на чтение для членов группы и всех остальных пользователей. Выполняем команды: «mkdir monthly» (создание каталога), «chmod go-r monthly» (изменение прав). 4. Создаем файл ~/abc1 с правом записи для членов группы, используя команды: «touch abc1» (создание файла), «chmod g+w abc1» (изменение прав).

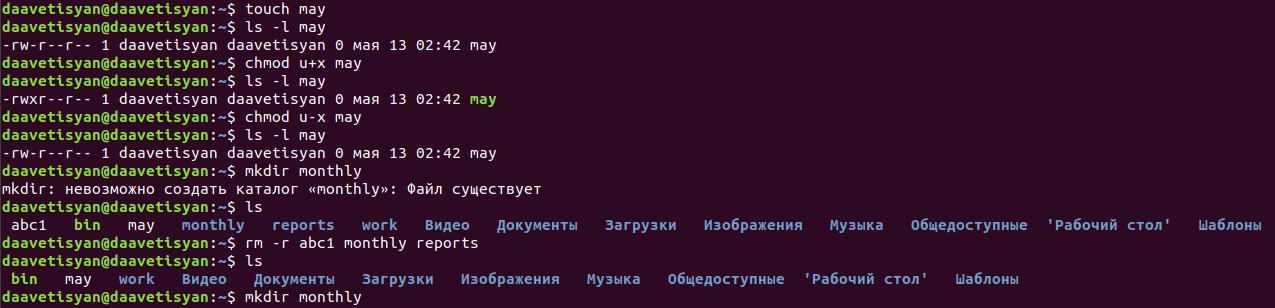


Figure 3: Выполним примеры, описанные в лабораторной работе

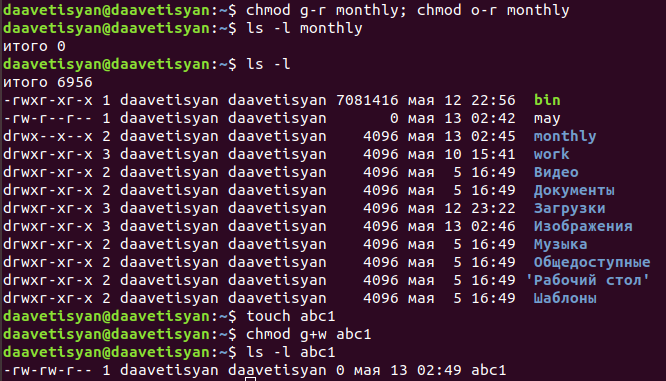


Figure 4: Выполним примеры, описанные в лабораторной работе

1. Выполняем следующие действия, отображенные на (рис. -fig. 5) (рис. -fig. 6) (рис. -fig. 7):
   1. Копируем файл /usr/include/aio.h (т.к. у меня нет каталога /usr/include/sys/, то беру произвольный файл из каталога /usr/include/) в домашний каталог (команда «cp /usr/include/aio.h ~») и назоваем его equipment (команда «mv aio.h equipment»).
   2. В домашнем каталоге создаем директорию ~/ski.plases (команда «mkdir ski.plases»).
   3. Перемещаем файл equipment в каталог ~/ski.plases (команда «mv equipment ski.plases»).
   4. Переименовываем файл ~/ski.plases/equipment в ~/ski.plases/equiplist (команда «mv ski.plases/equipment ski.plases/equiplist»).
   5. Создаем в домашнем каталоге файл abc1 (команда «touch abc1») и копируем его в каталог ~/ski.plases (команда «cp abc1 ski.plases»), называем его equiplist2 (команда «mv ski.plases/abc1 ski.plases/equiplist2»).
   6. Создаем каталог с именем equipment в каталоге ~/ski.plases (команда «mkdir ski.plases/equipment»).
   7. Перемещаем файлы ~/ski.plases/equiplist и equiplist2 в каталог ~/ski.plases/equipment (команда «mv ski.plases/equiolist ski.plases/equiplist2 ski.plases/equipment»).
   8. Создаем (команда «mkdir newdir») и перемещаем каталог ~/newdir в каталог ~/ski.plases (команда «mv newdir ski.plases») и называем его plans (команда «mv ski.plases/newdir ski.plases/plans»).

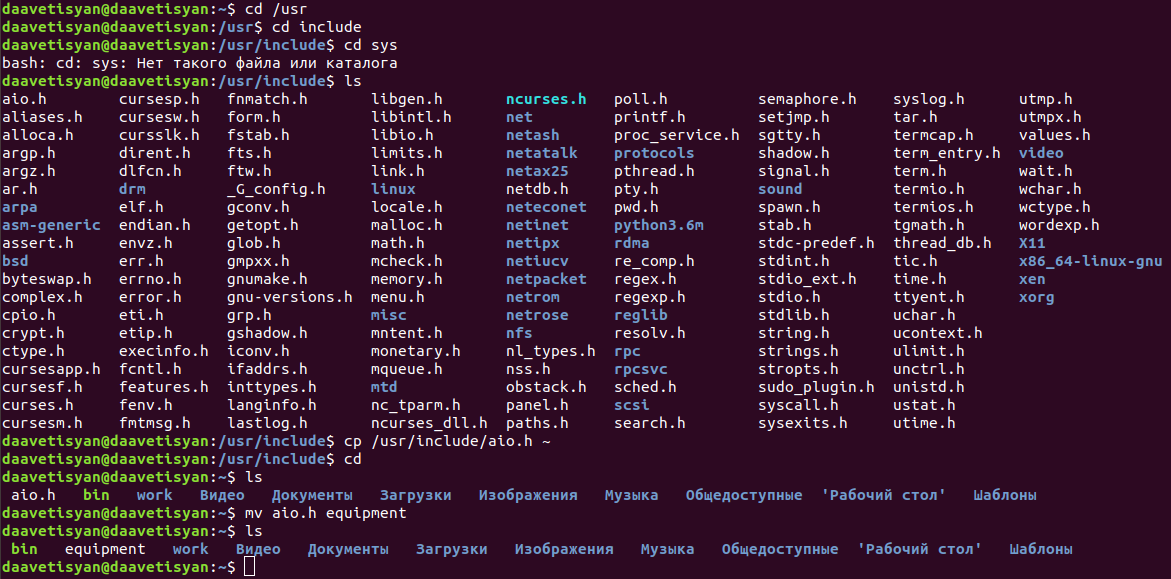


Figure 5: Выполним действия из пункта 2 задания

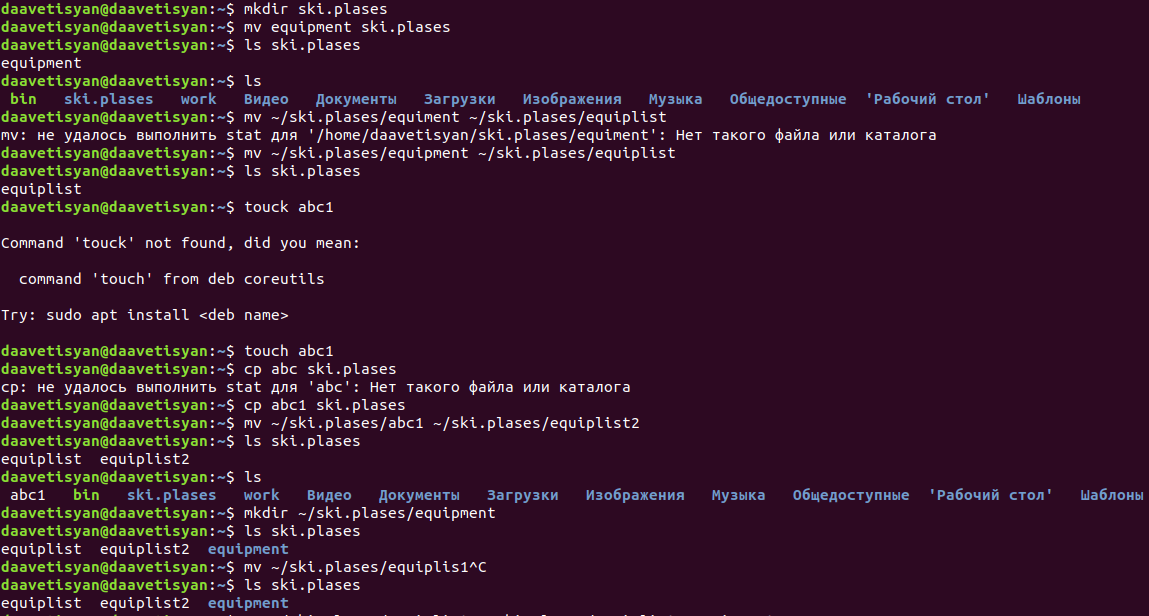


Figure 6: Выполним действия из пункта 2 задания

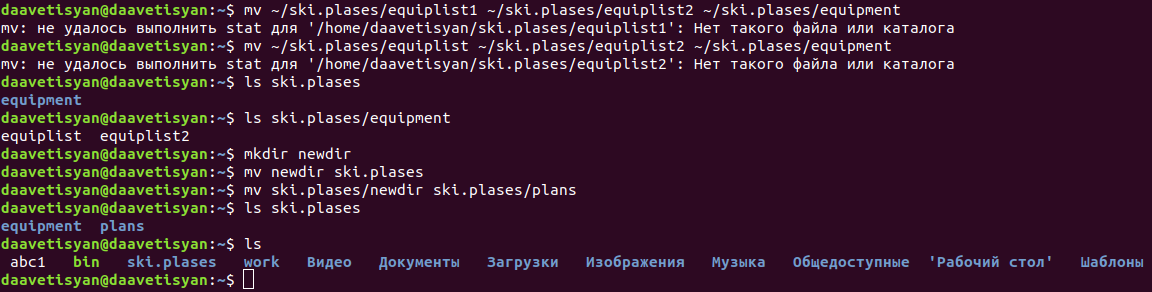


Figure 7: Выполним действия из пункта 2 задания

1. Определяем опции команды chmod, необходимые для того, чтобы присвоить соответствующим файлам выделенные права доступа, считая, что в начале таких прав нет. Предварительно создаем необходимые файлы, используя команды: «mkdir australia play», «touch my\_os feathers».

* drwxr–r– … australia: команда «chmod 744 australia» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только чтение)
* drwx–x–x … play: команда «chmod 711 play» (это каталог, владелец имеет право на чтение, запись и выполнение, группа владельца и остальные – только выполнение)
* -r-xr–r– … my\_os: команда «chmod 544 my\_os» (это файл, владелец имеет право на чтение и выполнение, группа владельца и остальные – только чтение)
* -rw-rw-r– … feathers: команда «chmod 664 feathers» (это файл,владелец и группа владельца имеют право на чтение и запись, остальные – только чтение)  
  Командой «ls -l» проверяем правильность выполненных действий (рис. -fig. 8).

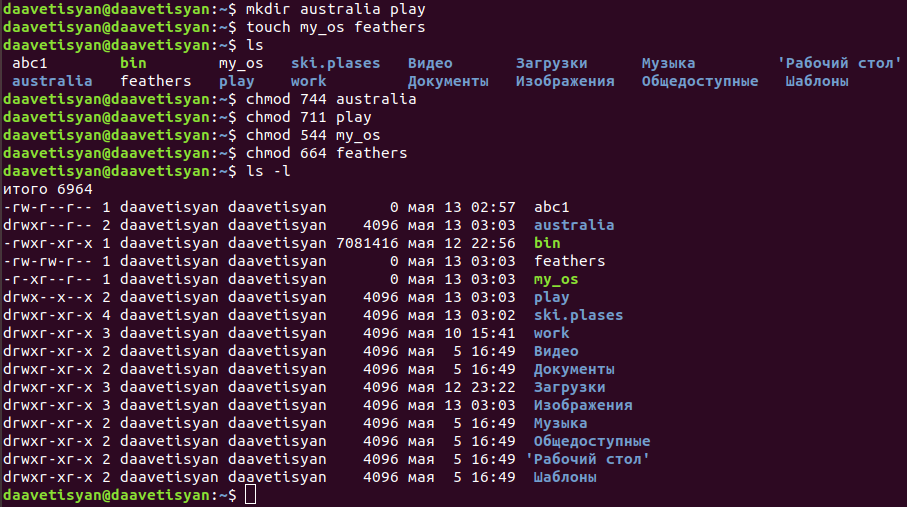


Figure 8: Определяем опции команды chmod

1. Выполняем следующие действия, отображенные на (рис. -fig. 9) (рис. -fig. 10) (рис. -fig. 11):
   1. Просмотрим содержимое файла /etc/passwd (команда «cat /etc/passwd»).
   2. Копируем файл ~/feathers в файл ~/file.old (команда «cp feathers file.old»).
   3. Переместим файл ~/file.old в каталог ~/play (команда «mv file.ord play»).
   4. Скопируем каталог ~/play в каталог ~/fun (команда «cp -r play fun»).
   5. Переместим каталог ~/fun в каталог ~/play (команда «mv fun play») и назовем его games (команда «mv play/fun play/games»).
   6. Лишим владельца файла ~/feathers права на чтение (команда «chmodu-r feathers»).
   7. Если мы попытаемся просмотреть файл ~/feathers командой cat, то получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на чтение данного файла.
   8. Если мы попытаемся скопировать файл ~/feathers, например, в каталог monthly, то получим отказ в доступе, по причине, описанной в предыдущем пункте.
   9. Дадим владельцу файла ~/feathers право на чтение (команда «chmod u+r feathers»).
   10. Лишим владельца каталога ~/play права на выполнение (команда «chmod u-x play»).
   11. Перейдем в каталог ~/play (команда «cd play»). Получим отказ в доступе, т.к. в предыдущем пункте лишили владельца права на выполнение данного каталога.
   12. Дадим владельцу каталога ~/play право на выполнение (команда «chmod u+x play»).

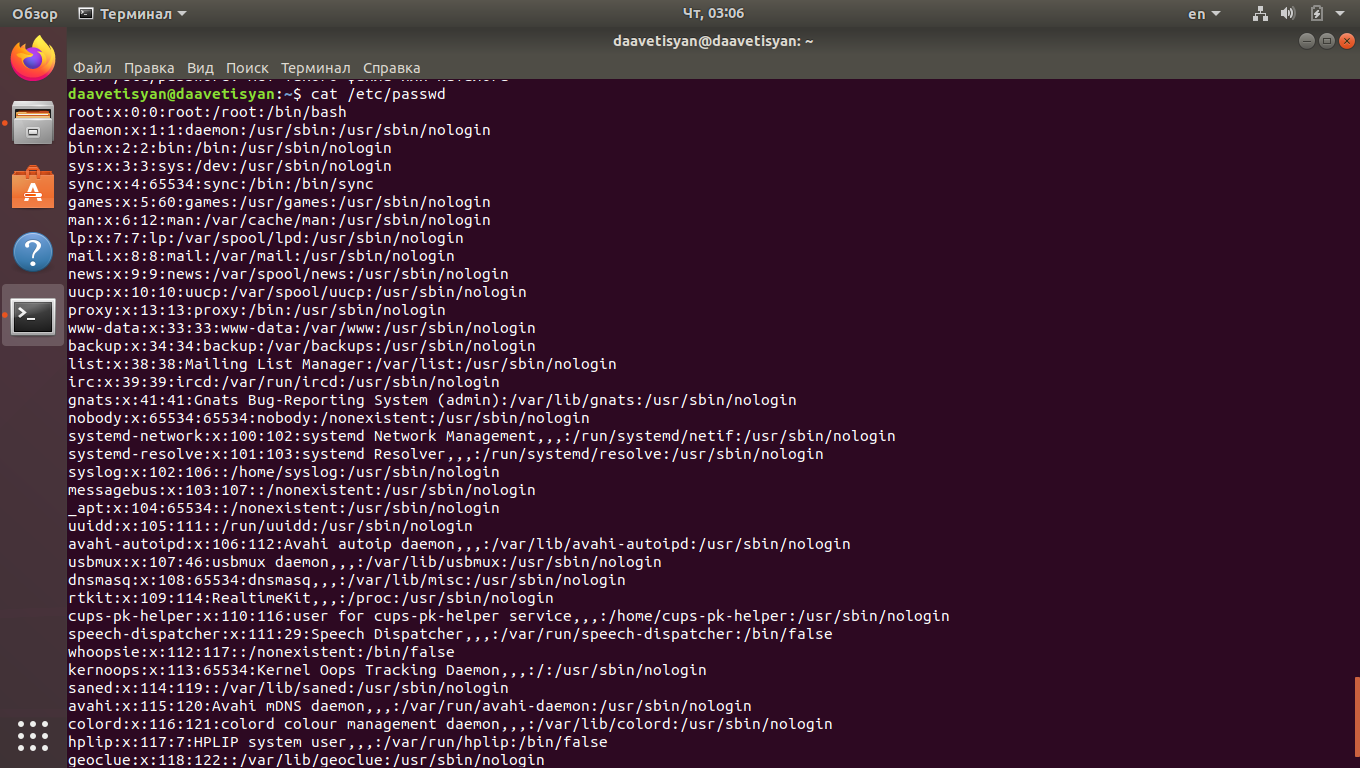


Figure 9: Просмотрим содержимое файла /etc/passwd

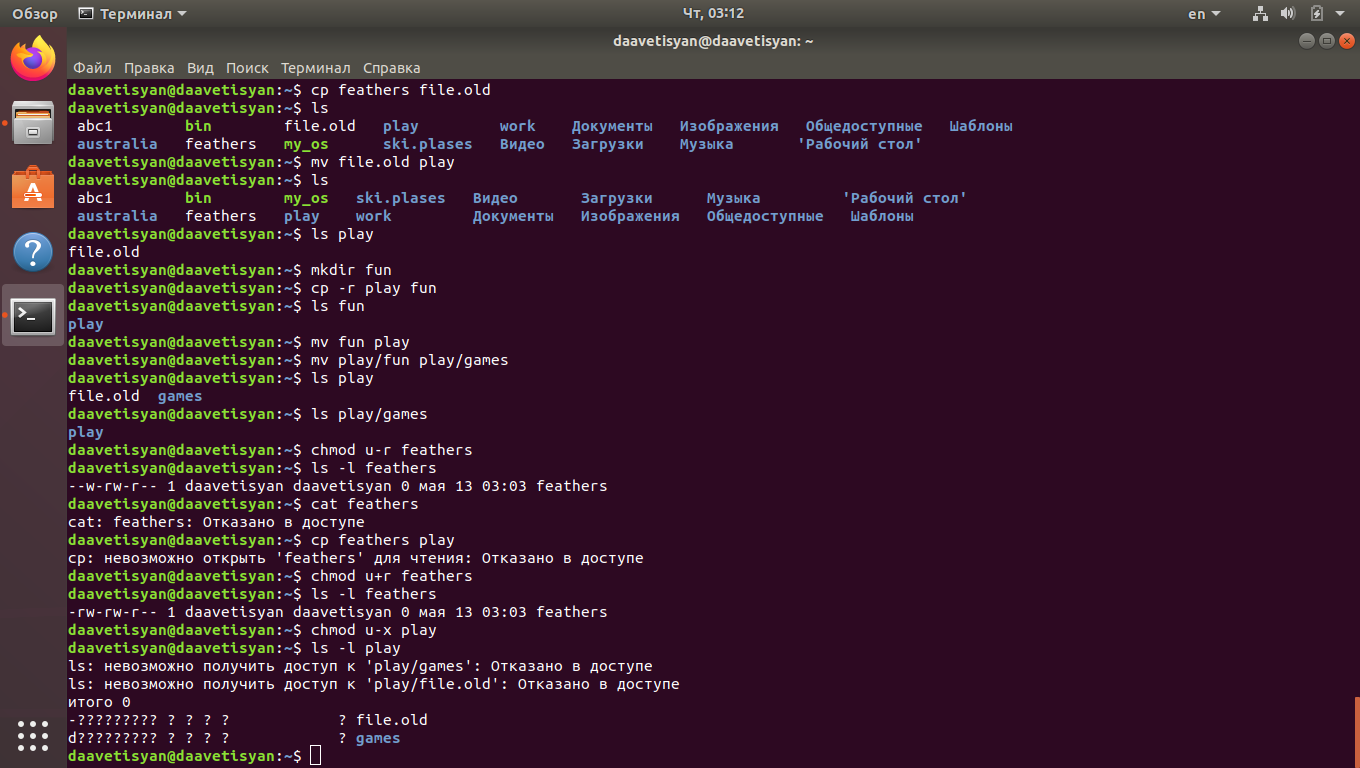


Figure 10: Выполним действия из пункта 4 задания

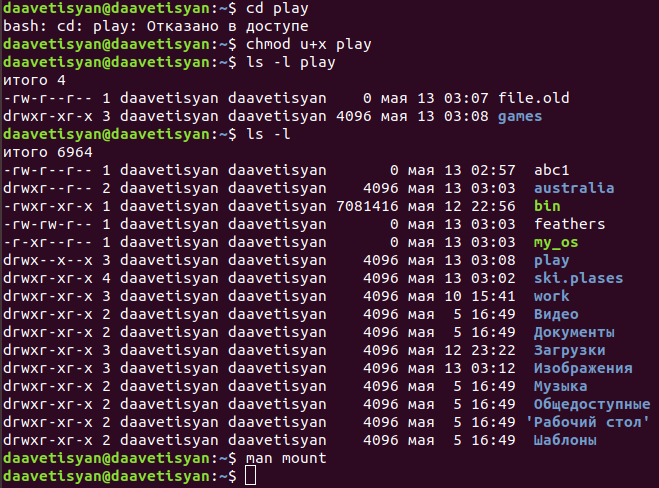


Figure 11: Выполним действия из пункта 4 задания

1. Используя команды «man mount», «man fsck», «man mkfs», «man kill», получим информацию о соответствующих командах.  
   Команда mount (рис. -fig. 12):  
   Предназначена для монтирования файловой системы. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого дерева обозначается как /. Физически файлы могут располагаться на различных устройствах. Команда mount служит для подключения файловых систем разных устройств к этому большому дереву.  
   Наиболее часто встречающаяся форма команды mount выглядит следующим образом:  
   «mount -t vfstype device dir»  
   Такая команда предлагает ядру смонтировать (подключить) файловую систему указанного типа vfstype, расположенную на устройстве device, к заданному каталогу dir, который часто называют точкой монтирования.

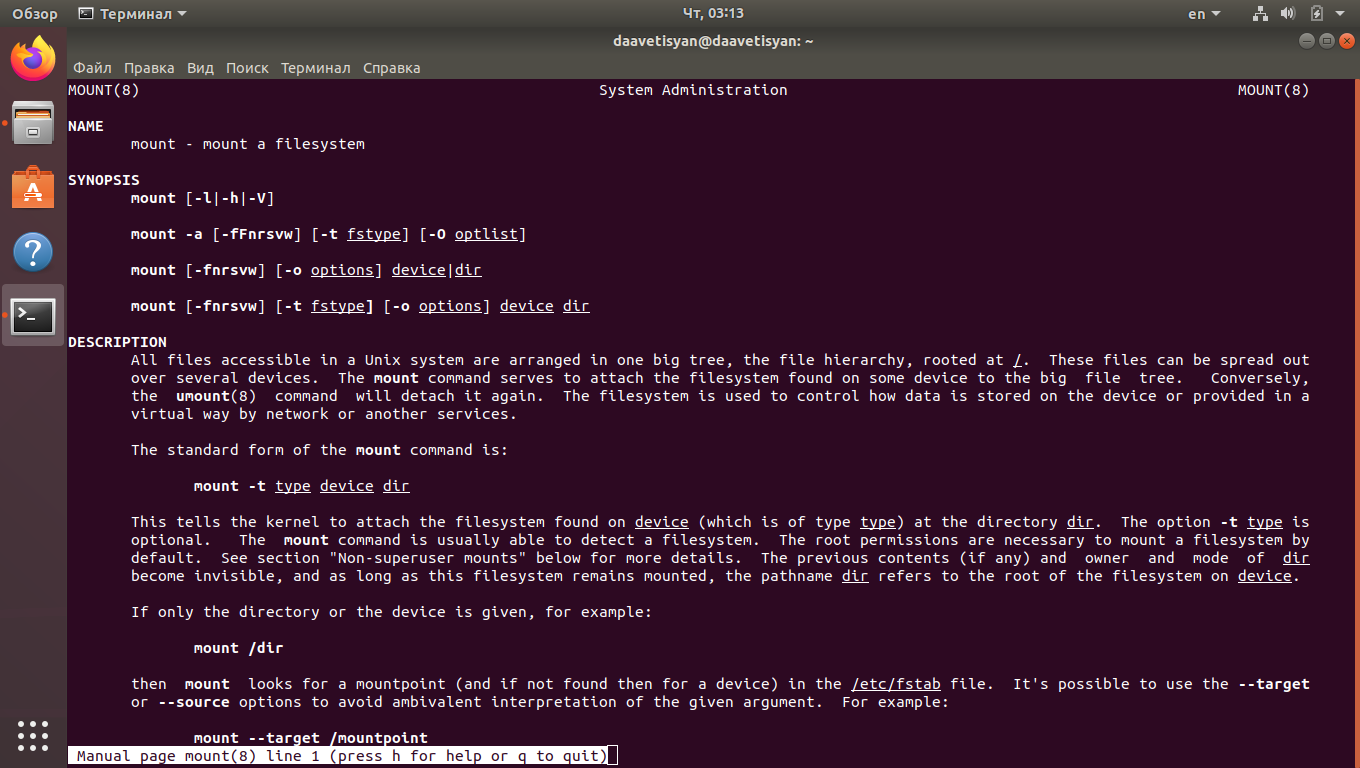


Figure 12: Команда mount

Команда fsck (рис. -fig. 13):  
Это утилита командной строки, которая позволяет выполнять проверки согласованности и интерактивное исправление в одной или нескольких файловых системах Linux. Он использует программы, специфичные для типа файловой системы, которую он проверяет.  
У команды fsck следующий синтаксис:  
fsck параметр – параметры ФС . . .  
Например, если нужно восстановить («починить») файловую систему на некотором устройстве /dev/sdb2, следует воспользоваться командой:  
«sudo fsck -y /dev/sdb2»  
Опция -y необходима, т. к. при её отсутствии придётся слишком часто давать подтверждение.

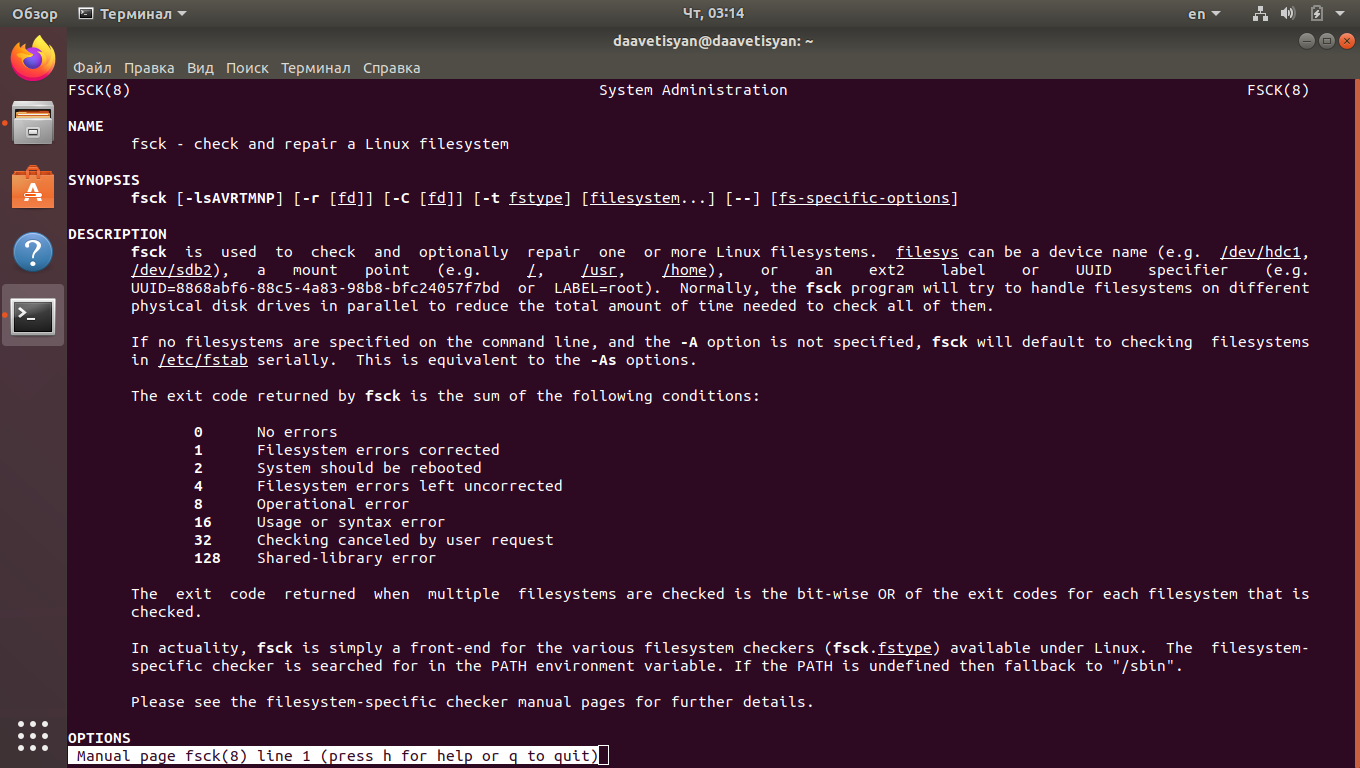


Figure 13: Команда fsck:

Команда mkfs (рис. -fig. 14):  
Создаёт новую файловую систему Linux.  
Имеет следующий синтаксис:  
mkfs -V -t fstype fs-options filesys blocks  
mkfs используется для создания файловой системы Linux на некотором устройстве, обычно в разделе жёсткого диска. В качестве аргумента filesys для файловой системы может выступать или название устройства (например, /dev/hda1, /dev/sdb2) или точка монтирования (например, /, /usr, /home).  
Аргументом blocks указывается количество блоков, которые выделяются для использования этой файловой системой.  
По окончании работы mkfs возвращает 0 - в случае успеха, а 1 - при неудачной операции.  
Например, команда «mkfs -t ext2 /dev/hdb1» создаёт файловую систему типа ext2 в разделе /dev/hdb1 (второй жёсткий диск).

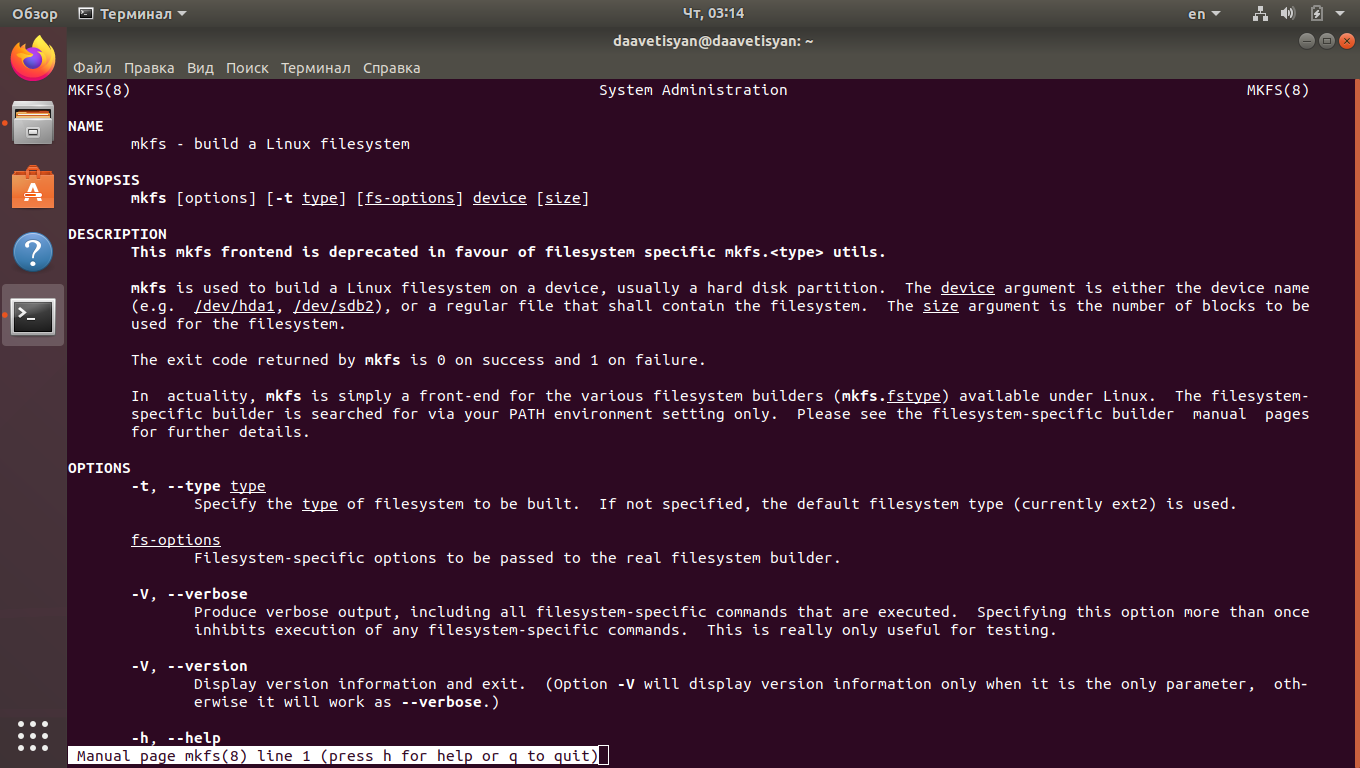


Figure 14: Команда mkfs

Команда kill (рис. -fig. 15):  
Посылает сигнал процессу или выводит список допустимых сигналов.  
Имеет следующий синтаксис:  
kill опции PID, где PID – это PID (числовой идентификатор) процесса или несколько PID процессов, если требуется послать сигнал сразу нескольким процессам.  
Например, команда «kill -KILL 3121» посылает сигнал KILL процессу с PID 3121, чтобы принудительно завершить процесс.

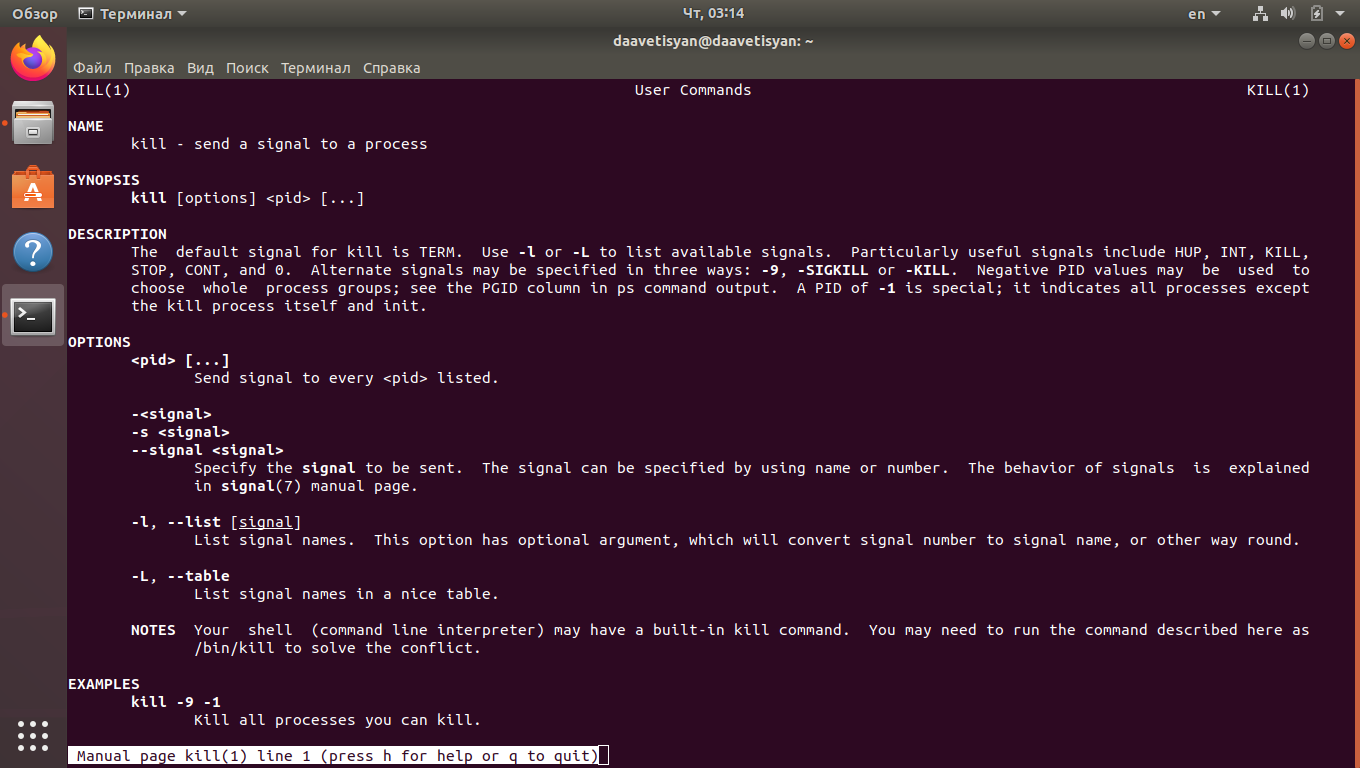


Figure 15: Команда kill

# Контрольные вопросы

1. Чтобы узнать, какие файловые системы существуют на жёстком диске моего компьютера, использую команду «df -Th». На моем компьютере есть следующие файловые системы: devtmpfs, tmpfs, ext4, iso9660. devtmpfs позволяет ядру создать экземпляр tmpfs с именем devtmpfs при инициализации ядра, прежде чем регистрируется какое-либо устройство с драйверами. Каждое устройство с майором / минором будет предоставлять узел устройства в devtmpfs. devtmpfs монтируется на /dev и содержит специальные файлы устройств для всех устройств. tmpfs − временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо ПЗУ. Подобная конструкция является RAM диском. Данная файловая система также предназначенная для быстрого и ненадёжного хранения временных данных. Хорошо подходит для /tmp и массовой сборки пакетов/образов. Предполагает наличие достаточного объёма виртуальной памяти. Файловая система tmpfs предназначена для того, чтобы использовать часть физической памяти сервера как обычный дисковый раздел, в котором можно сохранять данные (чтение и запись). Поскольку данные размещены в памяти, то чтение или запись происходят во много раз быстрее, чем с обычного HDD диска. ext4 − имеет обратную совместимость с предыдущими версиями ФС. Эта версия была выпущена в 2008 году. Является первой ФС из «семейства» Ext, использующая механизм «extent file system», который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность файловой системы. Кроме того, в Ext4 реализован механизм отложенной записи (delayed allocation − delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны, хотя механизм отложенной записи и используется во многих ФС, но в силу сложности своей реализации он повышает вероятность утери данных.  
   Характеристики:

* максимальный размер файла: 16 TB;
* максимальный размер раздела: 16 TB;
* максимальный размер имени файла: 255 символов.  
  Рекомендации по использованию:
* наилучший выбор для SSD;
* наилучшая производительность по сравнению с предыдущими Etx-системами;
* она так же отлично подходит в качестве файловой системы для серверов баз данных, хотя сама система и моложе Ext3.  
  ISO 9660 − стандарт, выпущенный Международной организацией по стандартизации, описывающий файловую систему для дисков CD-ROM. Также известен как CDFS (Compact Disc File System). Целью стандарта является обеспечить совместимость носителей под разными операционными системами, такими, как Unix, Mac OS, Windows.

1. Файловая система Linux/UNIX физически представляет собой пространство раздела диска разбитое на блоки фиксированного размера, кратные размеру сектора − 1024, 2048, 4096 или 8120 байт. Размер блока указывается при создании файловой системы. В файловой структуре Linux имеется один корневой раздел − / (он же root, корень). Все разделы жесткого диска (если их несколько) представляют собой структуру подкаталогов, “примонтированных” к определенным каталогам.

* / − корень  
  Это главный каталог в системе Linux. По сути, это и есть файловая система Linux. Адреса всех файлов начинаются с корня, а дополнительные разделы, флешки или оптические диски подключаются в папки корневого каталога.Только пользователь root имеет право читать и изменять файлы в этом каталоге.
* /BIN – бинарные файлы пользователя  
  Этот каталог содержит исполняемые файлы. Здесь расположены программы, которые можно использовать в однопользовательском режиме или режиме восстановления.
* /SBIN – системные испольняемые файлы  
  Так же как и /bin, содержит двоичные исполняемые файлы, которые доступны на ранних этапах загрузки, когда не примонтирован каталог /usr. Но здесь находятся программы, которые можно выполнять только с правами суперпользователя.
* /ETC – конфигурационные файлы  
  В этой папке содержатся конфигурационные файлы всех программ, установленных в системе. Кроме конфигурационных файлов, в системе инициализации Init Scripts, здесь находятся скрипты запуска и завершения системных демонов, монтирования файловых систем и автозагрузки программ.
* /DEV – файлы устройств  
  В Linux все, в том числе внешние устройства являются файлами. Таким образом, все подключенные флешки, клавиатуры, микрофоны, камеры − это просто файлы в каталоге /dev/. Выполняется сканирование всех подключенных устройств и создание для них специальных файлов.
* /PROC – информация о процессах  
  По сути, это псевдофайловая система, содержащая подробную информацию о каждом процессе, его Pid, имя исполняемого файла, параметры запуска, доступ к оперативной памяти и так далее. Также здесь можно найти информацию об использовании системных ресурсов.
* /VAR – переменные файлы  
  Название каталога /var говорит само за себя, он должен содержать файлы, которые часто изменяются. Размер этих файлов постоянно увеличивается. Здесь содержатся файлы системных журналов, различные кеши, базы данных и так далее.
* /TMP – временные файлы  
  В этом каталоге содержатся временные файлы, созданные системой, любыми программами или пользователями. Все пользователи имеют право записи в эту директорию.
* /USR – программы пользователя  
  Это самый большой каталог с большим количеством функций. Здесь находятся исполняемые файлы, исходники программ, различные ресурсы приложений, картинки, музыку и документацию.
* /HOME – домашняя папка  
  В этой папке хранятся домашние каталоги всех пользователей. В них они могут хранить свои личные файлы, настройки программ и т.д.
* /BOOT – файлы загрузчика  
  Содержит все файлы, связанные с загрузчиком системы. Это ядро vmlinuz, образ initrd, а также файлы загрузчика, находящие в каталоге /boot/grub.
* /LIB – системные библиотеки  
  Содержит файлы системных библиотек, которые используются исполняемыми файлами в каталогах /bin и /sbin.
* /OPT – дополнительные программы  
  В эту папку устанавливаются проприетарные программы, игры или драйвера. Это программы созданные в виде отдельных исполняемых файлов самими производителями.
* /MNT – монтирование  
  В этот каталог системные администраторы могут монтировать внешние или дополнительные файловые системы.
* /MEDIA – съемные носители  
  В этот каталог система монтирует все подключаемые внешние накопители –USB флешки, оптические диски и другие носители информации.
* /SRV – сервер В этом каталоге содержатся файлы серверов и сервисов.
* /RUN - процессы Каталог, содержащий PID файлы процессов, похожий на /var/run, но в отличие от него, он размещен в TMPFS, а поэтому после перезагрузки все файлы теряются.

1. Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount.
2. Целостность файловой системы может быть нарушена из-за перебоев в питании, неполадок в оборудовании или из-за некорректного/внезапного выключения компьютера. Чтобы устранить повреждения файловой системы необходимо использовать команду fsck.
3. Файловую систему можно создать, используя команду mkfs. Ее краткое описание дано в пункте 5 в ходе выполнения заданий лабораторной работы.
4. Для просмотра текстовых файлов существуют следующие команды:

* сat  
  Задача команды cat очень проста − она читает данные из файла или стандартного ввода и выводит их на экран.  
  Синтаксис утилиты:  
  cat опции файл1 файл2 …  
  Основные опции:  
  -b – нумеровать только непустые строки  
  -E – показывать символ $ в конце каждой строки  
  -n – нумеровать все строки  
  -s – удалять пустые повторяющиеся строки-T – отображать табуляции в виде ^I  
  -h – отобразить справку  
  -v – версия утилиты
* nl  
  Команда nl действует аналогично команде cat, но выводит еще и номера строк в столбце слева.
* less  
  Cущественно более развитая команда для пролистывания текста.  
  При чтении данных со стандартного ввода она создает буфер, который позволяет листать текст как вперед, так и назад, а также искать как по направлению к концу, так и по направлению к началу текста.  
  Синтаксис аналогичный синтаксису команды cat.  
  Некоторые опции:  
  -g – при поиске подсвечивать только текущее найденное слово (по умолчанию подсвечиваются все вхождения)  
  -N – показывать номера строк
* head  
  Команда head выводит начальные строки (по умолчанию − 10) из одного или нескольких документов. Также она может показывать данные, которые передает на вывод другая утилита.  
  Синтаксис аналогичный синтаксису команды cat.  
  Основные опции:  
  -c (–bytes) − позволяет задавать количество текста не в строках, а в байтах  
  -n (–lines) − показывает заданное количество строк вместо 10, которые выводятся по умолчанию  
  -q (–quiet, –silent) − выводит только текст, не добавляя к нему название файла  
  -v (–verbose) − перед текстом выводит название файла-z (–zero-terminated) − символы перехода на новую строку заменяет символами завершения строк
* tail  
  Эта команда позволяет выводить заданное количество строк с конца файла, а также выводить новые строки в интерактивном режиме.  
  Синтаксис аналогичный синтаксису команды cat.  
  Основные опции:  
  -c − выводить указанное количество байт с конца файла  
  -f − обновлять информацию по мере появления новых строк в файле  
  -n − выводить указанное количество строк из конца файла  
  –pid − используется с опцией -f, позволяет завершить работу утилиты, когда завершится указанный процесс  
  -q − не выводить имена файлов  
  –retry − повторять попытки открыть файл, если он недоступен  
  -v − выводить подробную информацию о файле

1. Утилита cp позволяет полностью копировать файлы и директории.  
   Cинтаксис:  
   cp опции файл-источник файл-приемник  
   После выполнения команды файл-источник будет полностью перенесен в файл-приемник. Если в конце указан слэш, файл будет записан в заданную директорию с оригинальным именем.  
   Основные опции:  
   –attributes-only − не копировать содержимое файла, а только флаги доступа и владельца  
   -f, –force − перезаписывать существующие файлы  
   -i, –interactive − спрашивать, нужно ли перезаписывать существующие файлы  
   -L − копировать не символические ссылки, а то, на что они указывают-n − не перезаписывать существующие файлы  
   -P − не следовать символическим ссылкам  
   -r − копировать папку Linux рекурсивно  
   -s − не выполнять копирование файлов в Linux, а создавать символические ссылки  
   -u − скопировать файл, только если он был изменён  
   -x − не выходить за пределы этой файловой системы  
   -p − сохранять владельца, временные метки и флаги доступа при копировании  
   -t − считать файл-приемник директорией и копировать файл-источник в эту директорию
2. Команда mv используется для перемещения одного или нескольких файлов (или директорий) в другую директорию, а также для переименования файлов и директорий.  
   Синтаксис:  
   mv -опции старый\_файл новый\_файл  
   Основные опции:  
   –help − выводит на экран официальную документацию об утилите  
   –version − отображает версию mv  
   -b − создает копию файлов, которые были перемещены или перезаписаны  
   -f − при активации не будет спрашивать разрешение у владельца файла, если речь идет о перемещении или переименовании файла -i − наоборот, будет спрашивать разрешение у владельца  
   -n − отключает перезапись уже существующих объектов  
   –strip-trailing-slashes — удаляет завершающий символ / у файла при его наличии  
   -t директория — перемещает все файлы в указанную директорию  
   -u − осуществляет перемещение только в том случае, если исходный файл новее объекта назначения  
   -v − отображает сведения о каждом элементе во время обработки команды  
   Команда rename также предназначена, чтобы переименовать файл.  
   Синтаксис:  
   rename опции старое\_имя новое\_имя файлы  
   Основные опции:  
   -v − вывести список обработанных файлов  
   -n − тестовый режим, на самом деле никакие действия выполнены не будут  
   -f − принудительно перезаписывать существующие файлы
3. Права доступа − совокупность правил, регламентирующих порядок и условия доступа субъекта к объектам информационной системы (информации, её носителям, процессам и другим ресурсам) установленных правовыми документами или собственником, владельцем информации.  
   Права доступа к файлу или каталогу можно изменить, воспользовавшись командой chmod. Сделать это может владелец файла (или каталога) или пользователь с правами администратора.  
   Синтаксис команды:  
   chmod режим имя\_файла  
   Режим 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

# Выводы

В ходе выполнения данной лабораторной работы я ознакомился с файловой системой Linux, её структурой, именами и содержанием каталогов, получил навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.