Лабораторная работа №6

Бабков Дмитрий Николаевич, НПМбд-01-20

Содержание

[Цель работы: 1](#_Toc71989734)

[Задание: 1](#_Toc71989735)

[Ход работы: 2](#_Toc71989736)

[Вывод: 12](#_Toc71989737)

[Контрольные вопросы: 13](#_Toc71989738)

# Цель работы:

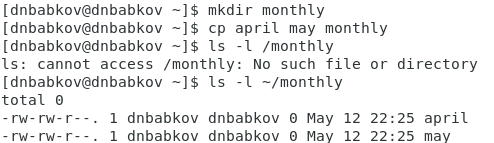
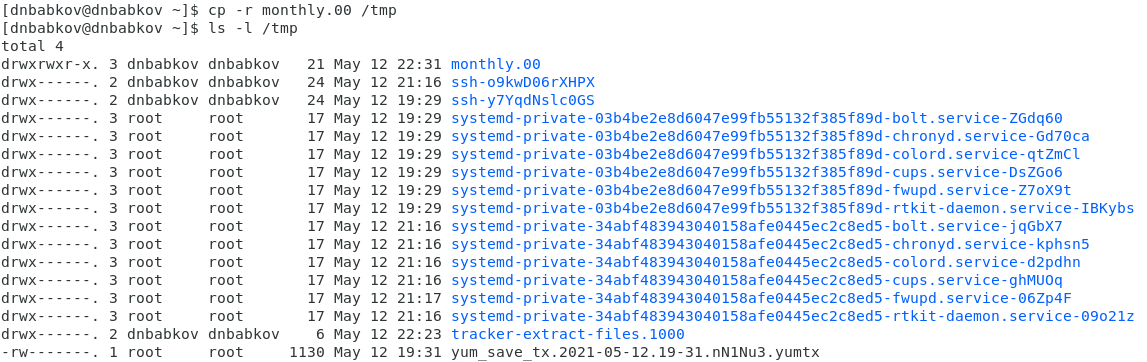
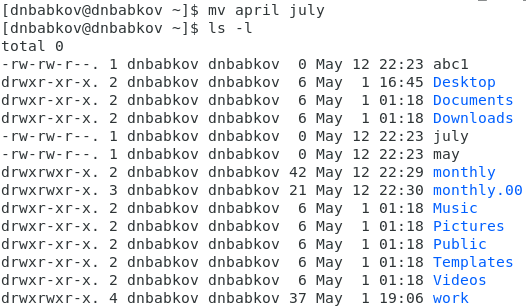
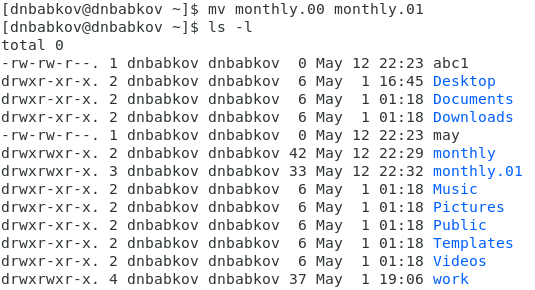
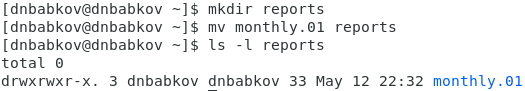
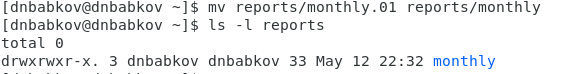
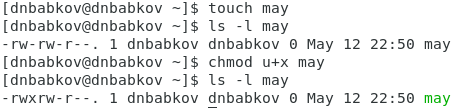
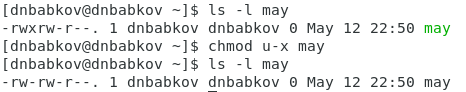
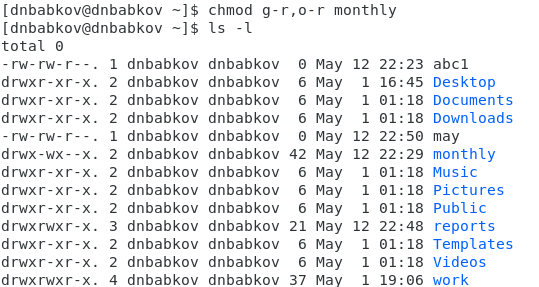
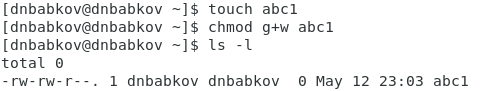
Ознакомление с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобретение практических навыков по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

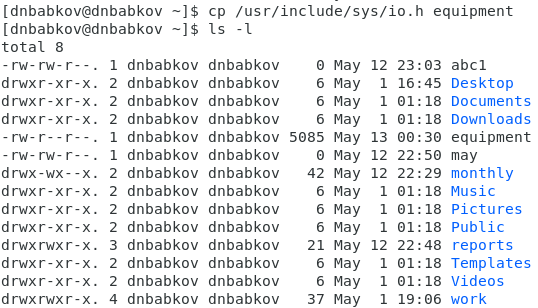
# Задание:

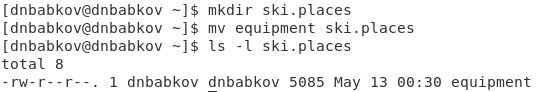
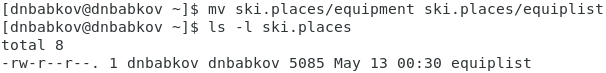
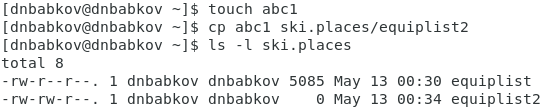
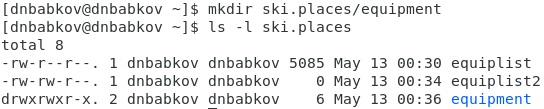
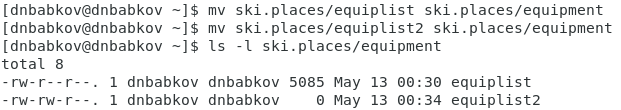
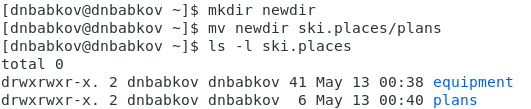
Ознакомиться с файловой системой Linux, её структурой, именами и содержанием каталогов. Приобрести практические навыки по применению команд для работы с файлами и каталогами, по управлению процессами (и работами), по проверке использования диска и обслуживанию файловой системы.

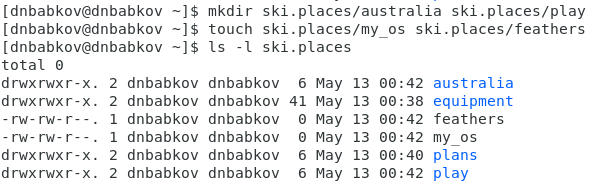
# Ход работы:

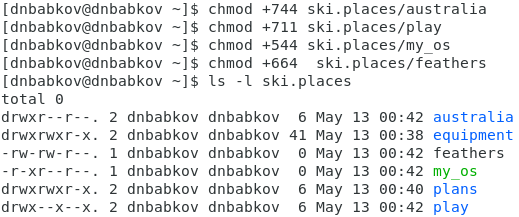
1. Выполним все примеры из первой части лабораторной работы:  
   1.1. Скопируем файл ~/abc1 в файл april и в файл may (-@Рис.1):  
   

* 1.2. Скопируем файлы april и may в каталог monthly (-@Рис.2):  
  
* 1.3. Скопируем файл monthly/may в файл с именем june (-@Рис.3):  
  Рис.3
* 1.4. Скопируем каталог monthly в каталог monthly.00 (-@Рис.4): 
* 1.5. Скопируем каталог monthly.00 в каталог /tmp (-@Рис.5): 
* 1.6. Изменим название файла april на july в домашнем каталоге (-@Рис.6):  
  
* 1.7. Переместим файл july в каталог monthly.00 (-@Рис.7):  
  Рис.7
* 1.8. Переименуем каталог monthly.00 в monthly.01 (-@Рис.8):  
  
* 1.9. Переместим каталог monthly.01 в каталог reports (-@Рис.9):  
  
* 1.10. Переименуем каталог reports/monthly.01 в reports/monthly:  
  
* 1.11. Требуется создать файл ~/may с правом выполнения для владельца:  
  
* 1.12. Требуется лишить владельца файла ~/may права на выполнение: 
* 1.13. Требуется создать каталог monthly с запретом на чтение для членов группы и всех остальных пользователей: 
* 1.14. Требуется создать файл ~/abc1 с правом записи для членов группы:  
  

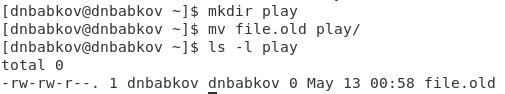
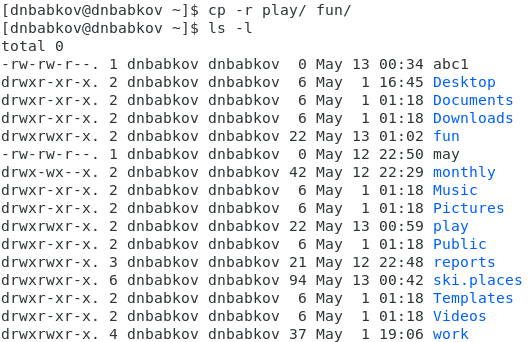
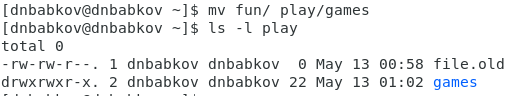
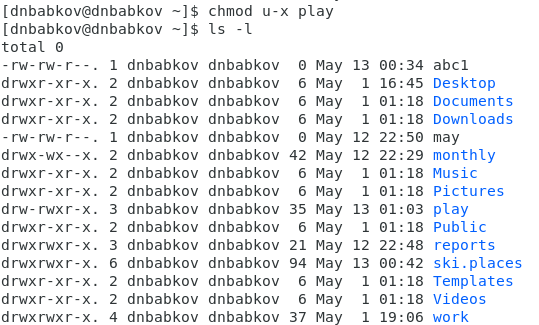
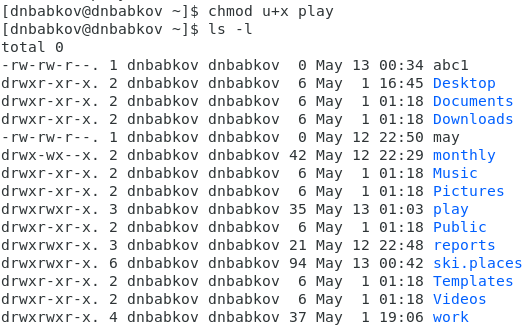
1. Выполним следующие действия:  
   2.1. C помощью команды **cp** скопируем файл **/usr/include/sys/io.h** в домашний каталог и назовем его **equipment** (-@Рис.15):  
   

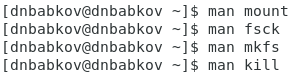
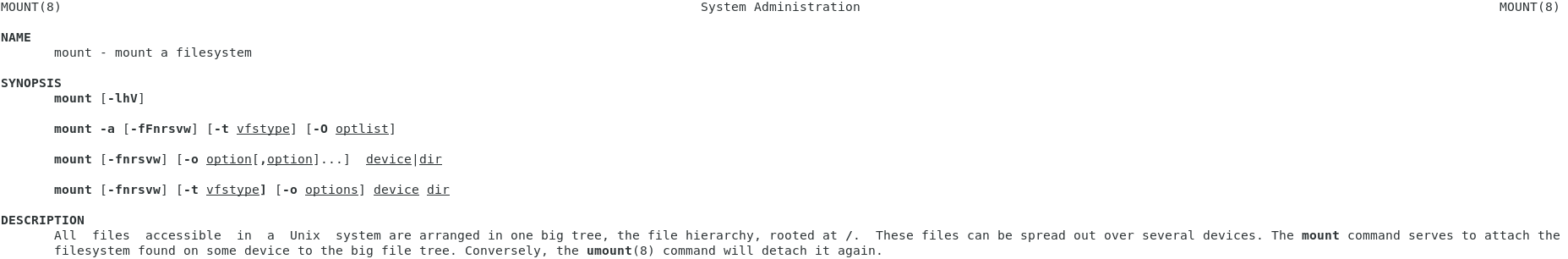
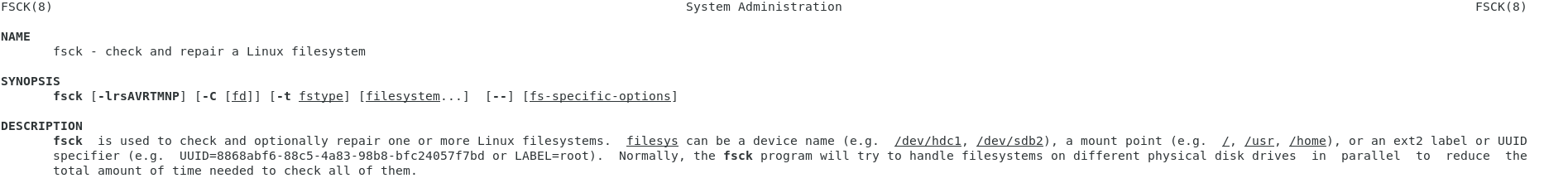
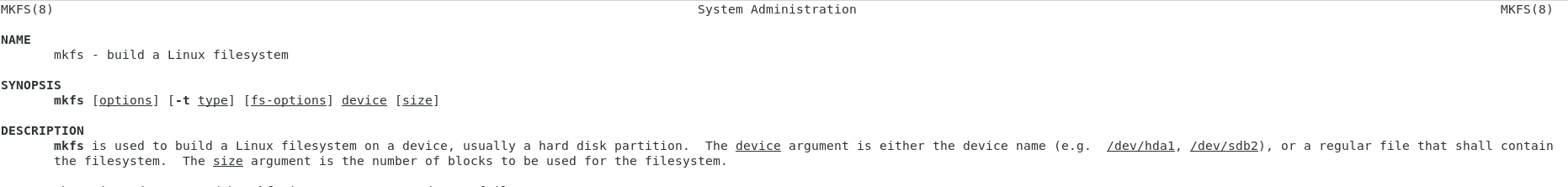
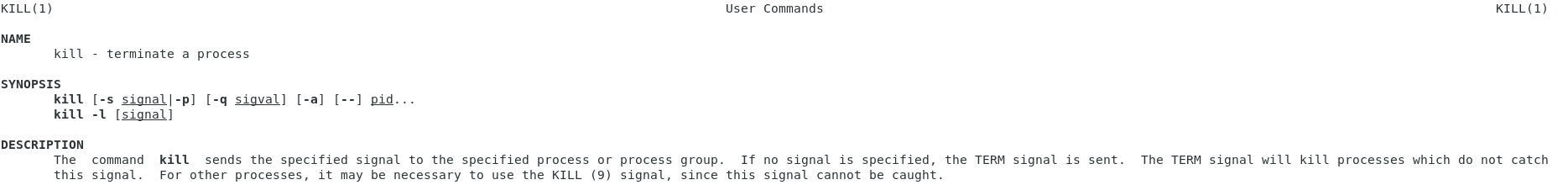
* 2.2. В домашнем каталоге создаем директорию **~/ski.places** и перемещаем туда файл **equipment** с помощью команды **mv** (-@Рис.16):  
  
* 2.3. Переименуем файл **~/ski.plases/equipment** в **~/ski.plases/equiplist** с помощью команды **mv** (-@Рис.17):  
  
* 2.4. Создадим в домашнем каталоге файл **abc1** и скопируем его в каталог **~/ski.plases**, назвав его equiplist2 (-@Рис.18):  
  
* 2.5. Создадим каталог **equipment** в каталоге **~/ski.places** и переместим в него файлы **equiplist** и **equiplist2** (-@Рис.19, -@Рис.20):  
    
  
* 2.6. Создадим и переместим каталог **~/newdir** в каталог **~/ski.plases** и назовём его **plans** (-@Рис.21):  
  

1. Присвоим перечисленным ниже файлам выделенные права доступа (предварительно создав эти файлы (-@Рис.22)):  
   

* drwxr–r– … australia; drwx–x–x … play; -r-xr–r– … my\_os; -rw-rw-r– … feathers (-@Рис.23)  
  

1. Проделаем приведенные ниже упражнения:  
   4.1. Просмотрите содержимое файла /etc/password (-@Рис.24).  
   

* 4.2. Скопируйте файл ~/feathers в файл ~/file.old (-@Рис.25).  
  
* 4.3. Переместите файл ~/file.old в каталог ~/play (-@Рис.26).  
  
* 4.4. Скопируйте каталог ~/play в каталог ~/fun (-@Рис.27).  
  
* 4.5. Переместите каталог ~/fun в каталог ~/play и назовите его games (-@Рис.28).  
  
* 4.6. Лишите владельца файла ~/feathers права на чтение (-@Рис.29).  
  Рис.29
* 4.7. Что произойдёт, если вы попытаетесь просмотреть файл ~/feathers командой cat?  
  - Выдаст сообщение о том, что доступ запрещен (-@Рис.30)  
  Рис.30
* 4.8. Что произойдёт, если вы попытаетесь скопировать файл ~/feathers?  
  - Выдаст сообщение о том, что скопировать невозможно, т.к. нет разрешения на чтение (-@Рис.31)  
  Рис.31
* 4.9. Дайте владельцу файла ~/feathers право на чтение (-@Рис.32).  
  Рис.32
* 4.10. Лишите владельца каталога ~/play права на выполнение (-@Рис.33).  
  
* 4.11. Перейдите в каталог ~/play. Что произошло?  
  - Выдаст сообщение о том, что нет доступа (-@Рис.34)  
  Рис.34
* 4.12. Дайте владельцу каталога ~/play право на выполнение (-@Рис.35)  
  

1. Используя команды «man mount», «man fsck», «man mkfs», «man kill», получим информацию о соответствующих командах (-@Рис.36)  
   
   * Команда mount:  
     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. (-@Рис.37)  
     
   * Команда fsck:  
     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. (-@Рис.38)  
     
   * Команда mkfs:  
     Cоздаёт 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). (-@Рис.39)  
     
   * Команда kill:  
     Посылает сигнал процессу или выводит список допустимых сигналов. Имеет следующий синтаксис: **kill [опции] PID**, где PID – это 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 (-@Рис.40).  
     

# Вывод:

В ходе выполнения этой лабораторной работы я ознакомился с файловой системой Linux и получил навыки по применению команд для работы с файлами, каталогами, по управлению процессами, по проверке использования диска и обслуживанию файловой системы.

# Контрольные вопросы:

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 исполняемые 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.
2. Чтобы содержимое некоторой файловой системы было доступно операционной системе необходимо воспользоваться командой mount.
3. 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.
4. Файловую систему можно создать, используя команду mkfs. Ее краткое описание было дано в ходе выполнения заданий лабораторной работы.
5. 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