**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Операционные системы»**

**Тема: Файловые системы Unix-подобных ОС**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1304 |  | Чернякова В.А. |
| Преподаватель |  | Душутина Е.В. |

Санкт-Петербург

2023

**Цель работы.**

Проанализировать функциональное назначение структурных элементов дерева ФС. Определить размещение корневого каталога (корневой ФС).

**Выполнение работы.**

1. Ознакомимся с типами файлов Linux.

Можно выделить следующие:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Типы файлов** |  | **Назначение** | | --- | --- | --- | | Обычные файлы | — | Хранение символьных и двоичных  данных | | Каталоги | d | Организация доступа к файлам | | Символьные ссылки | l | Предоставление доступа к файлам,  расположенных на любых носителях | | Блочные устройства | b | Предоставление интерфейса для  взаимодействия с аппаратным  обеспечением компьютера | | Символьные устройства | c | | Каналы | p | Организация взаимодействия  процессов в операционной системе | | Сокеты | s | |

Применяя утилиту ls отфильтруем по одному примеру каждого типа файла используемой ФС.

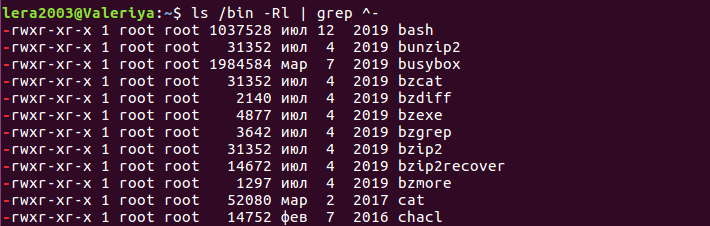
Команда для такой фильтрации выглядит следующим образом:

ls -Rl | grep ^(-, d, l, b, c, p, s)

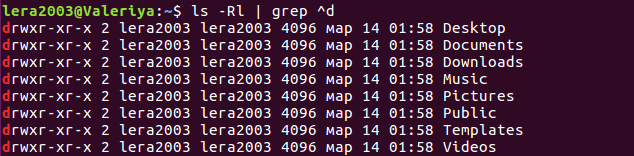
Ключ –l позволяет выводить подробный список, -R отвечает за рекурсивное отображение поддиректорий.

Команда grep позволяет осуществить поиск нужного типа файла, для этого после ^ необходимо указать соответствующий символ.

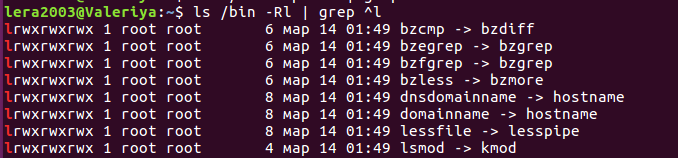
* Обычные файлы



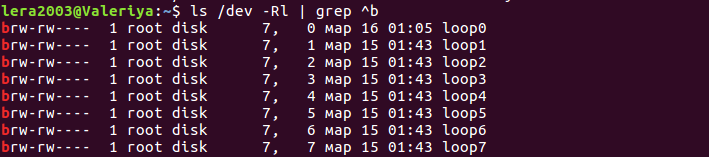
* Каталоги



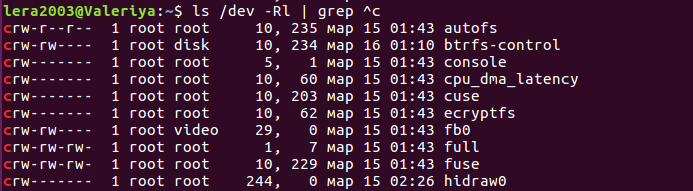
* Символьные ссылки



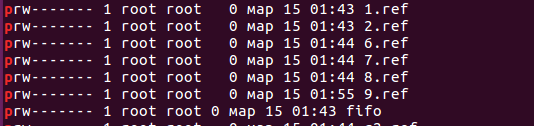
* Блочные устройства



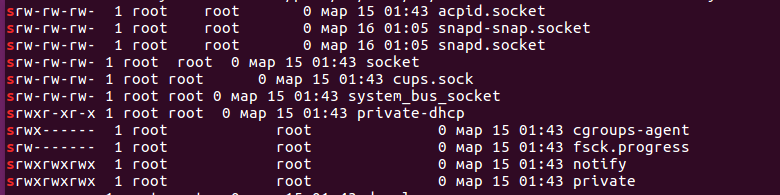
* Символьные устройства



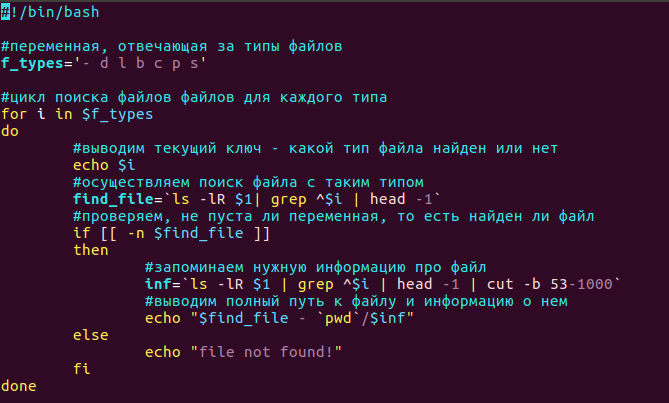
* Каналы



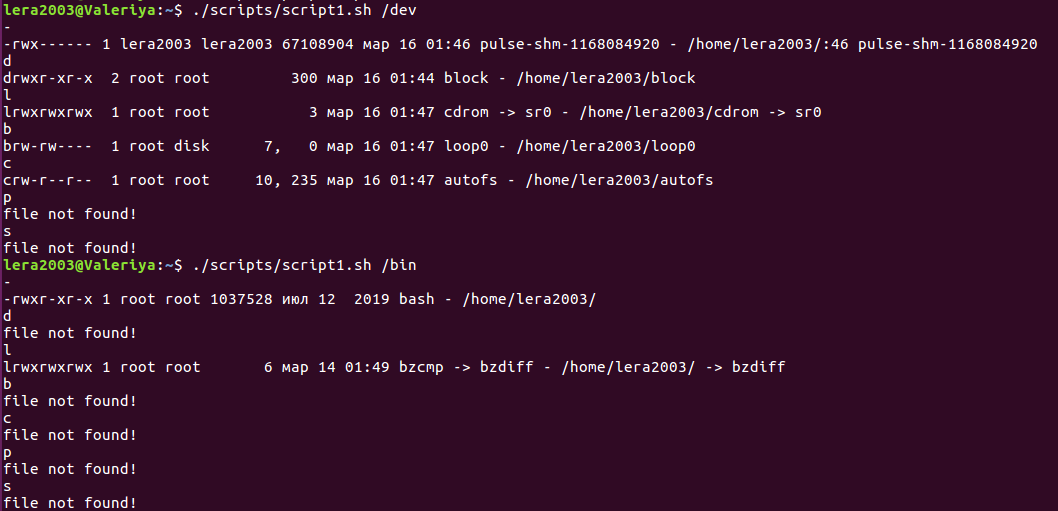
* Сокеты



Теперь создадим скрипт, который будет выводить все типы файлов в директории, переданной в качестве аргумента.

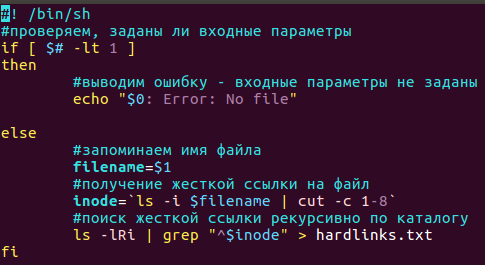


Результат работы скрипта.

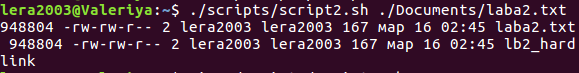


1. Напишем скрипт, который находит все жесткий ссылки на заданный

файл.



Результат работы скрипта.



1. Для файла laba2.txt сформируем символьные ссылки различными способами.

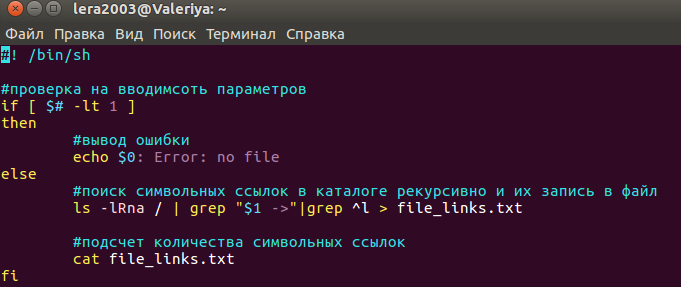
* ln –s



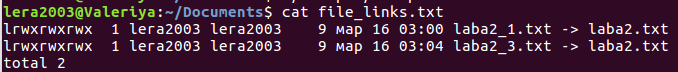
* cp –s



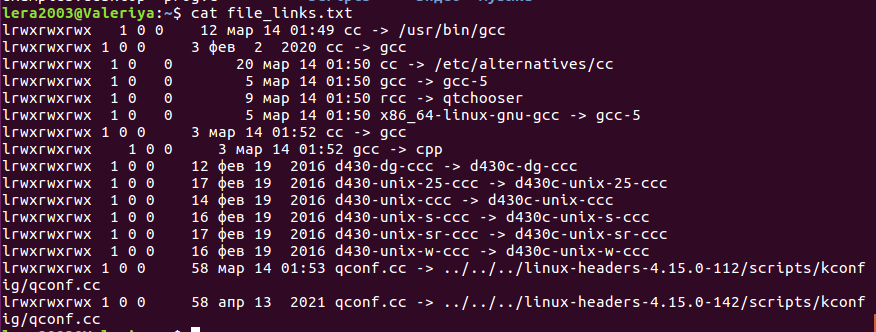
Создадим скрипт, перечисляющий все полно именные символьные ссылки на файл, размещаемые в разных местах файлового дерева.



Результат работы скрипта.



Найдем все символьные ссылки на сс.



1. Найдем все символьные ссылки на файл.



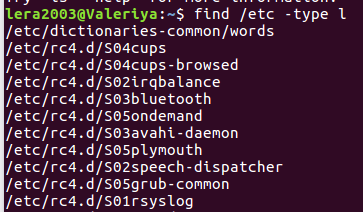
1. Изучим утилиту find.

Команда find - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

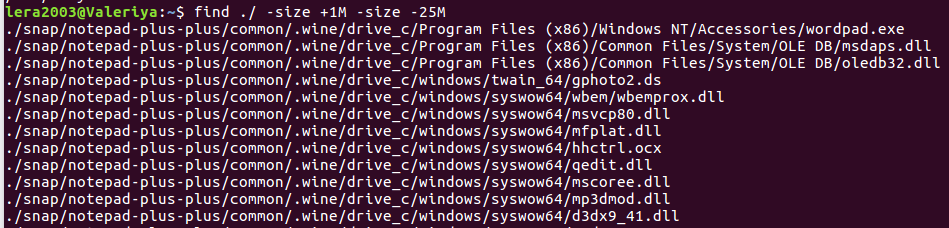
* Поиск файла по имени.



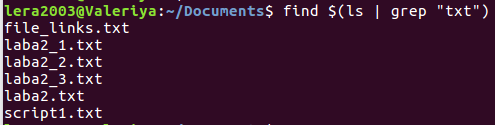
* Поиск по типу файла.



* Поиск файлов по размеру.

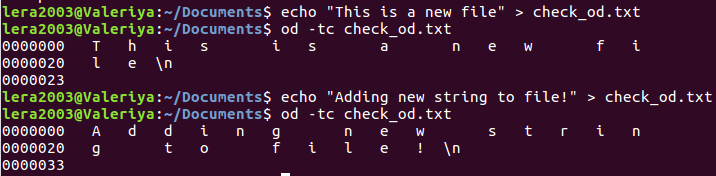


Организуем вложенную команду.



1. Проанализируем содержимое заголовка файла с помощью утилиты od.

Octal Dump (od) — это команда в Linux, используемая для преобразования файловых данных в различные форматы с восьмеричным форматом по умолчанию.

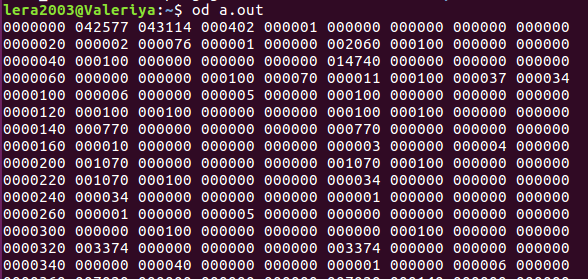


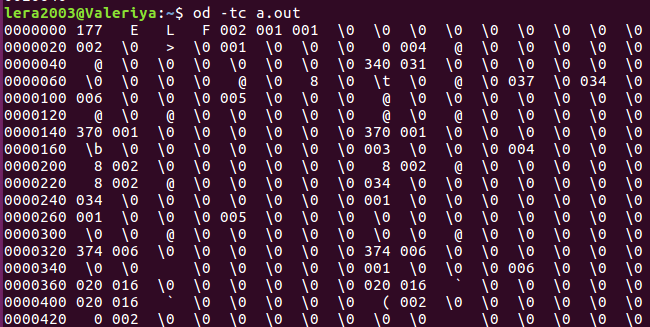
Команда od с ключами –tc выводит дамп памяти, ассоциированный с указанным файлом, побайтно в восьмеричном коде, заменяя код на символы там, где это возможно.

Проанализируем содержимое файла-каталога с помощью утилиты df.

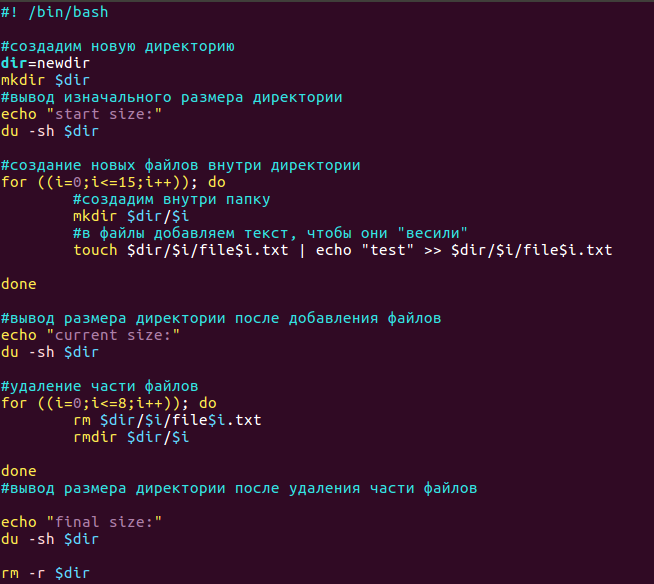


Проанализируем с помощью утилиты od исполняемый файл.

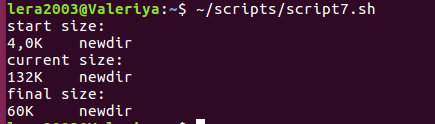




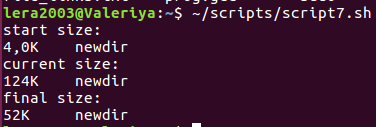
7. Изменим размер каталога, варьируя количество записей. Напишем скрипт, добавляющий и удаляющий часть файлов и выводящий размер директории в течении работы программы.



Результат работы скрипта.

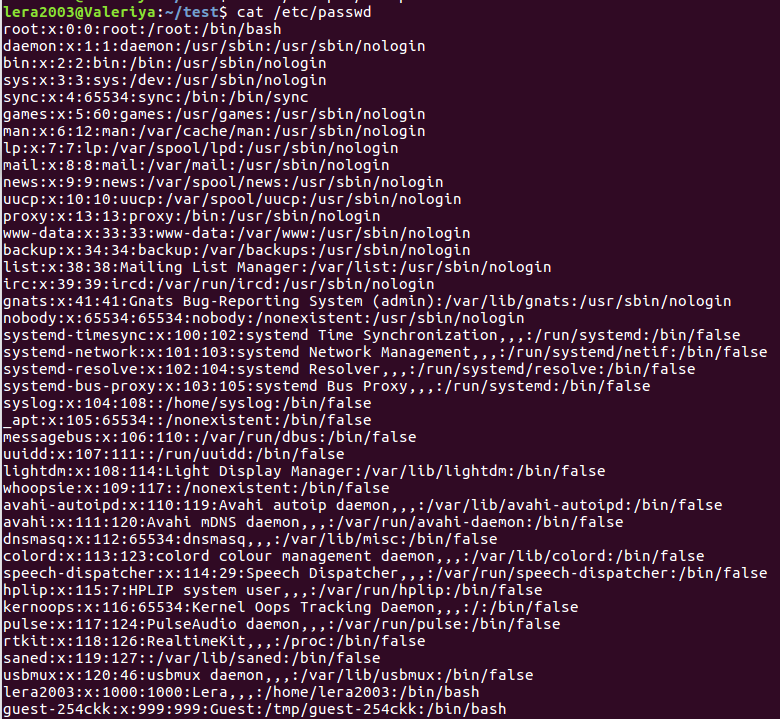


Изменим значение работы цикла for, при создании каталогов до 14. Результат работы скрипта в данном случае.



Таким образом, при добавлении одного каталога и одного текстового файла с его содержимым размер директории увеличивается на 8К.

1. Содержимое ***/etc/passwd.***



Файл /etc/passwd представляет собой текстовый файл с одной записью в строке, представляющей учетную запись пользователя.

Обычно первая строка описывает пользователя root, за которым следуют системные и обычные учетные записи пользователей. Новые записи добавляются в конец файла.

Каждая строка файла /etc/passwd содержит семь полей:

Username. Строка, которую вы вводите при входе в систему. Каждое имя пользователя должно быть уникальной строкой на компьютере. Максимальная длина имени пользователя ограничена 32 символами.

Password. В старых системах Linux зашифрованный пароль пользователя хранился в файле /etc/passwd. В большинстве современных систем это поле имеет значение x, и пароль пользователя сохраняется в файле /etc/shadow.

UID. Идентификатор пользователя — это номер, назначенный каждому пользователю. Он используется операционной системой для обращения к пользователю.

GID. Номер идентификатора группы пользователя, относящийся к основной группе пользователя. Когда пользователь создает файл, группа файла устанавливается на эту группу. Как правило, имя группы совпадает с именем пользователя. Пользователя вторичные группы перечислены в файле /etc/groups.

GECOS или полное имя пользователя. Это поле содержит список значений через запятую со следующей информацией:

Полное имя пользователя или название приложения.

Номер комнаты.

Рабочий номер телефона.

Домашний телефон.

Другая контактная информация.

Home directory. Абсолютный путь к домашнему каталогу пользователя. Он содержит файлы пользователя и конфигурации. По умолчанию домашние каталоги пользователей именуются по имени пользователя и создаются в каталоге /home.

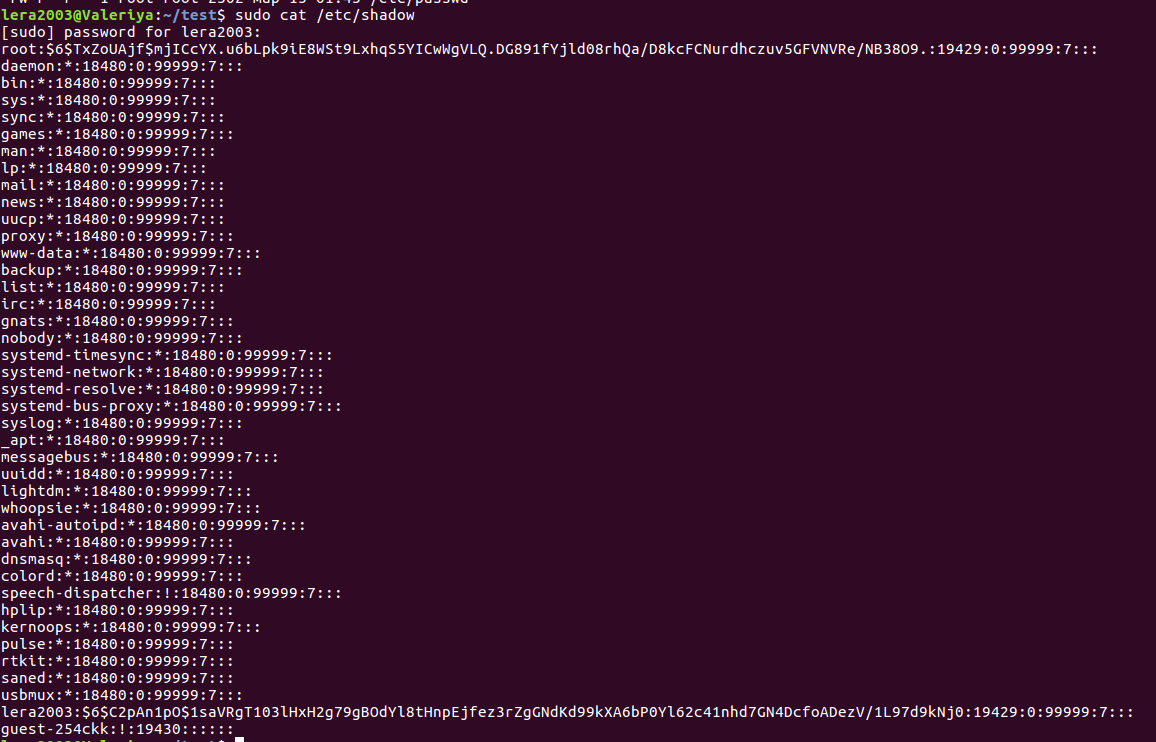
Login shell. Абсолютный путь к оболочке входа пользователя. Это оболочка, которая запускается, когда пользователь входит в систему. В большинстве дистрибутивов Linux оболочкой входа по умолчанию является Bash.

Проанализируем права доступа.



Файл принадлежит root, доступен для чтения всем пользователям, а для записи только root.

Содержимое ***/etc/shadow.***



Так как файл доступен для чтения только root, был совершен вход в качестве супер пользователя с помощью *sudo*.

/etc/shadow — это текстовый файл, содержащий информацию о паролях пользователей системы.

Файл содержит следующую информацию:

Имя пользователя. Строка, которую вы вводите при входе в систему. Учетная запись пользователя, которая существует в системе.

Зашифрованный пароль. Пароль использует формату $type$salt$hashed. $type является методом криптографического алгоритма хеширования и может иметь следующие значения:

$1$ — MD5

$2a$ — Blowfish

$2y$ — Eksblowfish

$5$ — SHA-256

$6$ — SHA-512

Если поле пароля содержит звездочку ( \*) или восклицательный знак ( !), пользователь не сможет войти в систему с использованием аутентификации по паролю. Другие методы входа, такие как аутентификация на основе ключей или переключение на пользователя, по-прежнему разрешены.

В старых системах Linux зашифрованный пароль пользователя хранился в файле /etc/passwd.

Последнее изменения пароля. Это дата последнего изменения пароля. Количество дней исчисляется с 1 января 1970 года (дата эпохи).

Минимальный срок действия пароля. Количество дней, которое должно пройти, прежде чем пароль пользователя может быть изменен. Как правило, он установлен на ноль, что означает отсутствие минимального срока действия пароля.

Максимальный срок действия пароля. Количество дней после смены пароля пользователя. По умолчанию этот номер установлен на 99999.

Период предупреждения. Количество дней до истечения срока действия пароля, в течение которого пользователь получает предупреждение о необходимости изменения пароля.

Период бездействия. Количество дней после истечения срока действия пароля пользователя до отключения учетной записи пользователя. Обычно это поле пустое.

Срок хранения. Дата, когда учетная запись была отключена. Это представляется как дата эпохи.

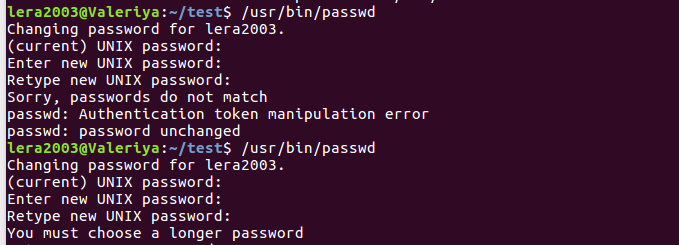
Неиспользованный. Это поле игнорируется. Оно зарезервировано для будущего использования.

Проанализируем права доступа.



Владельцем является root и он имеет право на чтение и запись без выполнения. Для группы право только для чтения.

Изучим утилиту /usr/bin/passwd – она позволяет менять пароль пользователя.



Проанализируем права доступа.



Для того, чтобы обычный пользователь мог изменить свой пароль, существует утилита /usr/bin/passwd, которая доступна администратору и группе root на чтение, запись и исполнение, а всем остальным – на чтение и исполнение. Эта программа может выполнять действия от имени администратора, независимо от того, кто её запустил.

1. Исследуем права владения и доступа, а также их сочетаемость.

В Unix каждому файлу соответствует набор прав доступа, представленный в виде 9-ти битов режима. Он определяет, какие пользователи имеют право читать файл, записывать в него данные или выполнять его.

Для назначения прав используются три группы флагов, первая определяет права для владельца, вторая - права для основной группы пользователя, третья - для всех остальных пользователей в системе.

Для файлов: r - право на чтение из файла; w - разрешает запись в файл (в частности перезапись или изменение); x - позволяет исполнить файл.

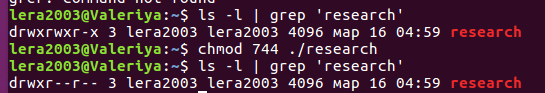
Для каталогов, флаги r w x имеют несколько отличный смысл: r - позволяет читать только имена файлов в каталоге; x - позволяет иметь доступ к самим файлам и их атрибутам (но не именам); w имеет смысл только в сочетании с x, и позволяет (в дополнение к x) манипулировать с файлами в каталоге (создавать, удалять и переименовывать). w без x - не имеет никакого эффекта.

* --- - нет прав, совсем;
* --x - разрешено только выполнение файла, как программы, но не изменение и не чтение;
* -w- - разрешена только запись и изменение файла;
* -wx - разрешено изменение и выполнение, но в случае с каталогом, вы
* не можете посмотреть его содержимое;
* r-- - права только на чтение;
* r-x - только чтение и выполнение, без права на запись;
* rw- - права на чтение и запись, но без выполнения;
* rwx - все права;

9.1. Создадим каталог research и применим утилиты chmod, chown.

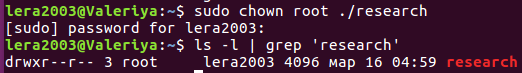
chmod (от англ. change mode) — команда для изменения прав доступа к файлам и каталогам, используемая в Unix-подобных операционных системах.

744 - разрешить все для владельца, а остальным только чтение.



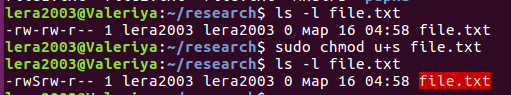
Команда chown позволяет изменить владельца пользователя и / или группы для данного файла, каталога или символической ссылки.

Изменим владельца папки research на root:



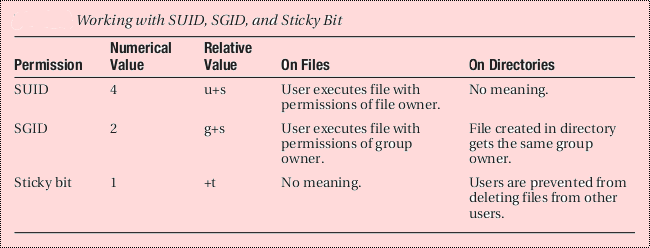
* 1. Расширим права исполнения экспериментального файла с помощью флага SUID.

4555 (-r-sr-xr-x) SUID Каждый пользователь имеет право читать и запускать на выполнение с правами владельца файла.



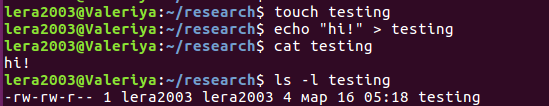
Буква S сигнализирует о флаге SUID в правах пользователя.

Другие виды расширенных прав доступа:

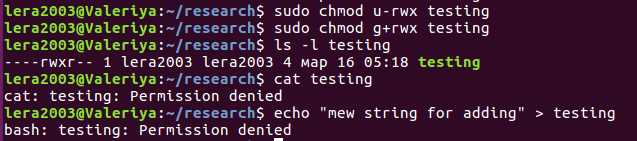


9.3. Экспериментально установить, как формируются итоговые права на использование файла, если права пользователя и группы, в которую он входит, различны.

Создадим файл testing и рассмотрим пользователя и группу lera2003.



Все действия выполнены успешно. Теперь исключим все права у пользователя и дадим их все группе.

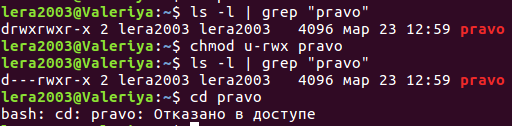


После выполнения операций владелец не может что-либо сделать, не смотря на принадлежность в группе, так как ему чтение, запись и исполнение запрещены.

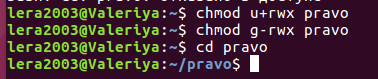
Подведём итог. Права доступа работают следующим образом:

* Оболочка проверяет, являетесь ли вы владельцем файла, к которому вы хотите получить доступ. Если вы являетесь этим владельцем, вы получаете разрешения и оболочка прекращает проверку.
* Если вы не являетесь владельцем файла, оболочка проверит, являетесь ли вы участником группы, у которой есть разрешения на этот файл. Если вы являетесь участником этой группы, вы получаете доступ к файлу с разрешениями, которые для группы установлены, и оболочка прекратит проверку.
* Если вы не являетесь ни пользователем, ни владельцем группы, вы получаете права других пользователей (Other)
  1. Сопоставим возможности исполнения наиболее часто используемых операций, варьируя правами доступа к файлу и каталогу.

Если убрать права у пользователя, то базовые операции недоступны, как для файлов, так и для каталогов.

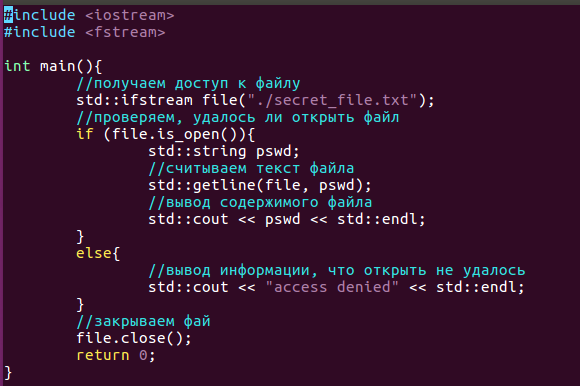
Уберем права у владельца, оставив у группы: 

Если же у владельца права есть – операции доступны.

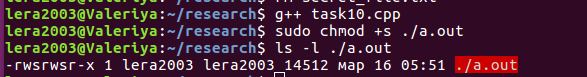


Для команды cat необходимы права для файла - r, для каталога - x. Для записи в файл через поток (например cat>>) права для файла - w, для каталога - x. Для выполнения скрипта права для файла - rx, для каталога - x. Для команды cd права для файла - -, для каталога - x. Для команды ls права для файла - -, для каталога - r.

10. Разработаем «программу-шлюз» для доступа к файлу другого пользователя при отсутствии прав на чтение информации из этого файла.



Командой chmod +s ./a.out дадим исполняемому файлу программы-шлюз разрешение SUID. Это позволит прочитать файл, если мы войдем в систему как другой пользователь.



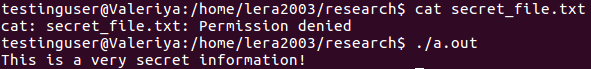
У файла secret\_file.txt уберем права чтения у пользователей и сменим текущего пользователя.







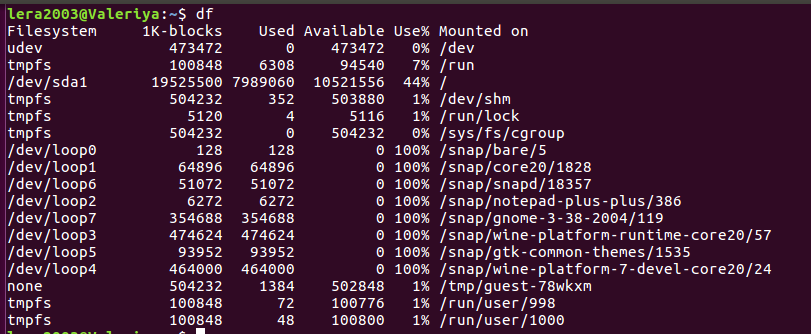
Попробуем прочитать данный файл без помощи программы и с ней.



Таким образом, за счёт того, что исполняемый файл имеет SUID, мы можем прочитать файл.

1. Применяя утилиту df и аналогичные ей по функциональности утилиты, а также информационные файлы типа fstab, получим информацию о файловых системах, возможных для монтирования, а также установленных на компьютере реально.

11.1. Информация об исследованных утилитах и информационных файлах с анализом их содержимого и форматов.

Утилита df предназначена для получения информации о свободном дисковом пространстве. Синтаксис*: df [-ghknP] [device|directory|file].* По умолчанию дисковое пространство измеряется в количестве 512-байтных блоков. Опция –k используется для измерения пространства количеством 1024-байтных блоков. Опция –h используется для измерения пространства в единицах, удобных для чтения человеком (байты, килобайты и т.д.). Опция –P используется для отображения заголовков столбцов таблицы. Опция –g используется для отображения более подробной информации. 

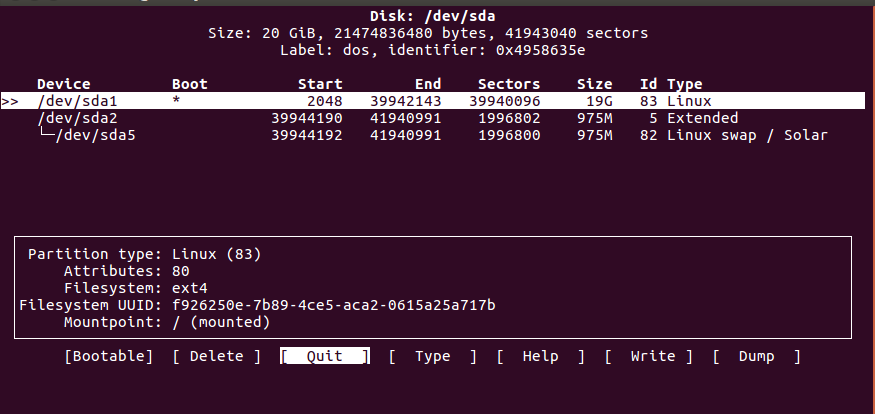
Каждая строка включает следующие столбцы:

* «Файловая система» — имя файловой системы.
* «1K-блоки» — размер файловой системы в 1K-блоках.
* «Использовано» — используемое пространство в блоках по 1К.
* «Доступно» — доступное пространство в блоках по 1К.
* «Use%» — процент используемого пространства.
* «Смонтировано» — каталог, в котором смонтирована фс.

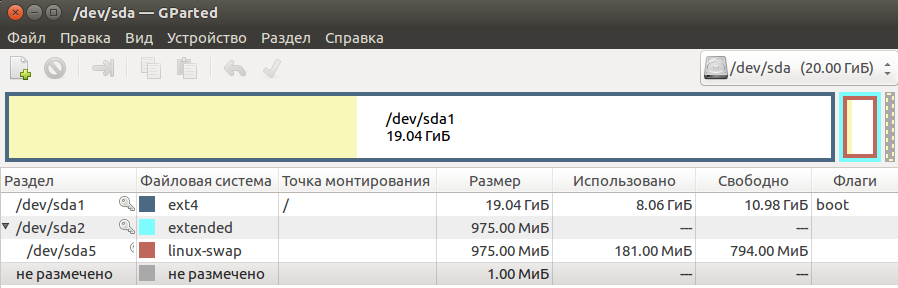
Были найдены следующий файловые системы:

* udev - управление устройствами для новых версий ядра Linux, являющийся преемником devfs, hotplug и HAL. Его основная задача — обслуживание файлов устройств (англ. device nodes) в каталоге /dev и обработка всех действий, выполняемых в пространстве пользователя при добавлении/отключении внешних устройств, включая загрузку firmware.
* tmpfs - временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо физического диска. Подобная конструкция является подобной RAM-диску.
* /dev/sda – термин sd означает диск SCSI, то есть диск с интерфейсом малой компьютерной системы. Итак, sda означает первый жесткий диск SCSI. Аналогично, / hda, отдельный раздел на диске принимает имена sda1, sda2 и т. Д. Активный раздел обозначается звездочкой \* в среднем столбце.
* /dev/loop - это виртуальное / loop устройство, которое в основном представляет собой файл в системе Linux, который действует как блочное устройство. Он используется для монтирования образов дисков, таких как Snap. /dev/loop читается только для чтения, поэтому размер фиксируется после создания и не может быть изменен. Для каждого программного обеспечения, устанавливаемого с помощью snap, создается устройство цикла. После удаления приложения автоматически удаляется связанный цикл /dev/loop.

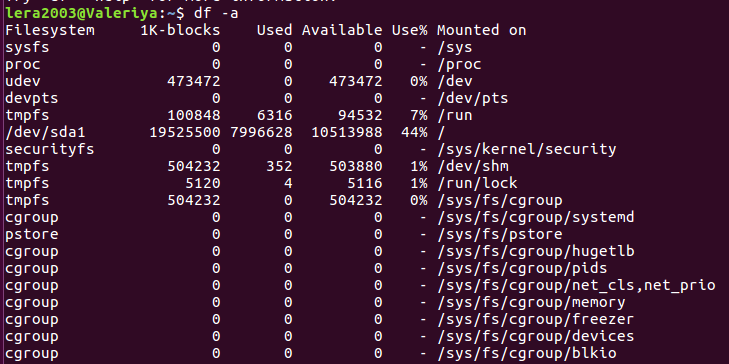
Утилита cfdisk подобна df. Это консольная программа для разметки жесткого диска и создания на нём разделов.



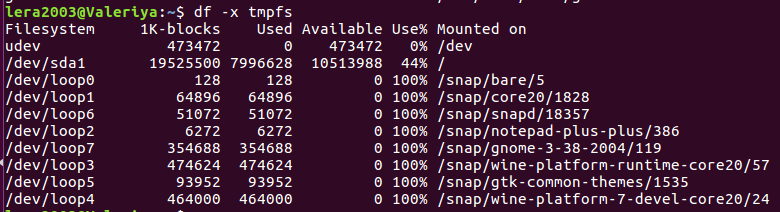
GParted (GNOME Partition Editor) — редактор дисковых разделов, являющийся графической оболочкой к GNU Parted.



Используя df –a, получаем информацию о всех ФС, известных ядру.



Для получения информации о реальных файловых системах на жестком диске: df –x tmpfs.



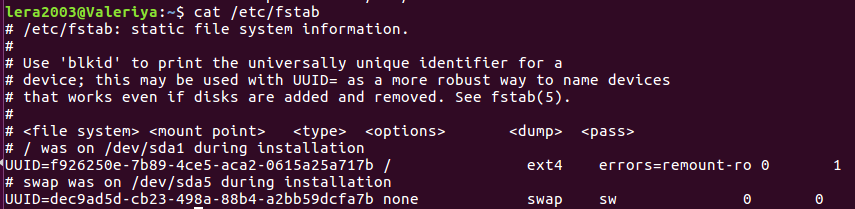
fstab.

Тип файла - обычный файл, разрешения для владельца - чтение и запись,

разрешение для группы и для всех остальных - только чтение. fstab - file systems table - один из конфигурационных файлов в UNIX - подобных системах, содержит информацию о ФС и устройствах хранения информации компьютера.



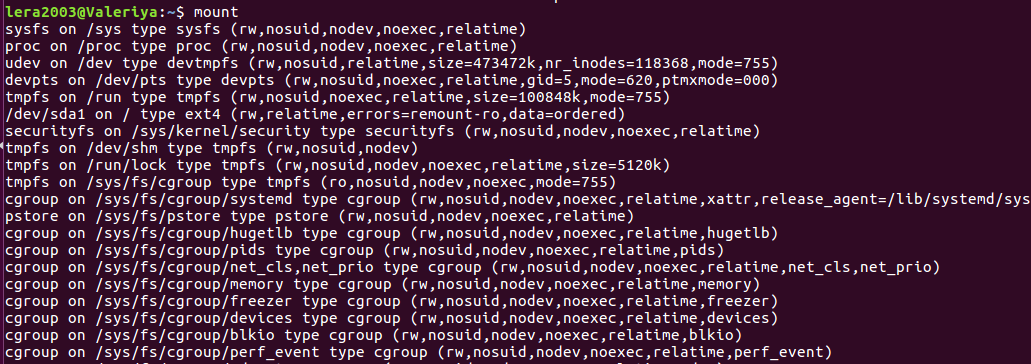
file\_system — сообщает mount, что монтировать.



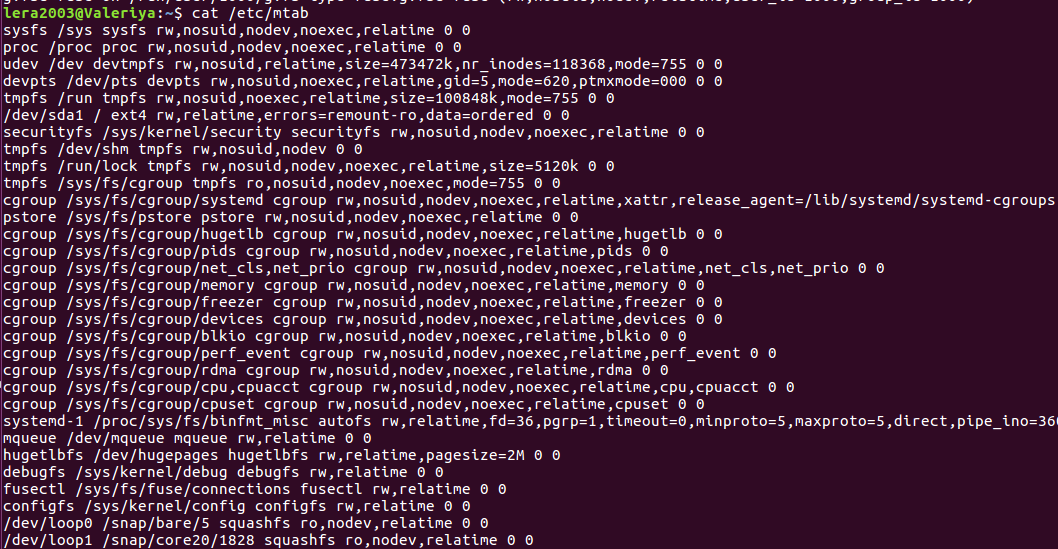
11.2. Привести образ диска с точки зрения состава и размещения всех ФС на испытуемом компьютере, а также образ полного дерева ФС, включая присоединенные ФС съемных и несъемных носителей. Проанализировать и указать формат таблицы монтирования.

mount — утилита командной строки в UNIX-подобных операционных

системах. Применяется для монтирования файловых систем.



Файл /etc/mtab – это файл, содержащий список смонтированных файловых систем. Любая смонтированная файловая система будет отображаться здесь. Если у вас есть диск или том, который не смонтирован, он не будет отображаться в этом файле.





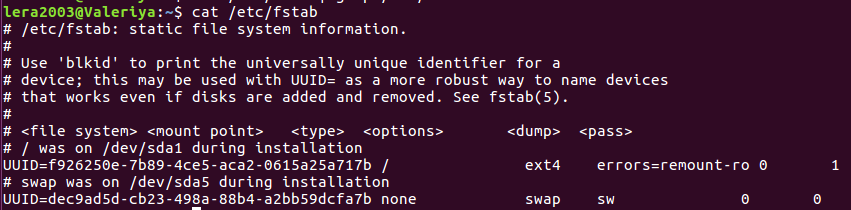
dev / sda - первый SCSI-диск с адресным идентификатором SCSI.

SCSI (англ. Small Computer System Interface) представляет собой набор стандартов для физического подключения и передачи данных между компьютерами и периферийными устройствами.



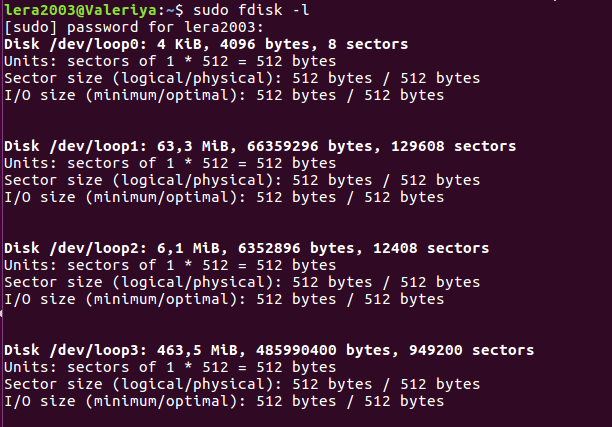
dev / sdb - второй SCSI-диск по адресу.

Файл fstab - это текстовый файл, который содержит информацию о различных файловых системах и устройствах хранения информации в вашем компьютере. Это всего лишь один файл, определяющий, как диск и/или раздел будут использоваться и как будут встроены в остальную систему.

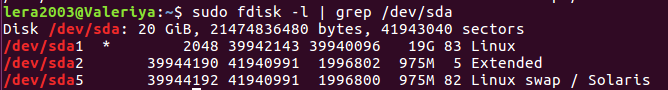


Строки файла содержат следующие поля:

* Что монтируем — некоторое **блочное устройство**, которое должно быть примонтировано.
* Куда монтируем — **точка монтирования** - путь в корневой файловой системе к каталогу, в который будет смонтировано устройство
* Тип **файловой системы** монтируемого раздела
* Опции монтирования файловой системы
* Индикатор необходимости делать резервную копию (как правило не используется и равно 0)
* Порядок проверки раздела (0- не проверять, 1 - устанавливается для корня, 2 - для остальных разделов).



fdisk высвечивает информацию о разделах диска. Кроме того, fdisk создает и удаляет разделы диска, меняет активный раздел. fdisk представляет расширенный набор функций по сравнению с одноименной командой MS-DOS. fdisk обычно используется с меню в интерактивном режиме.

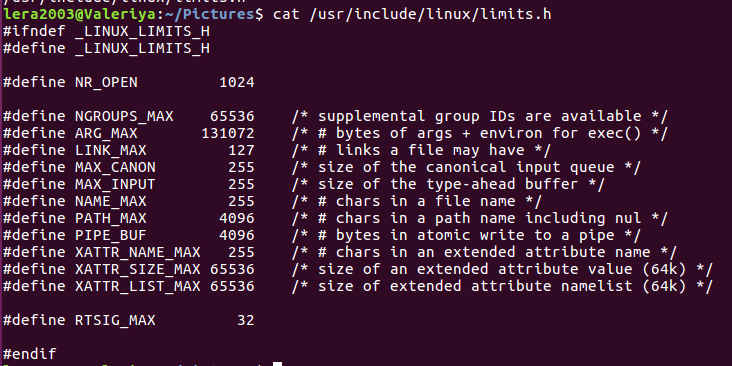




Утилита fdisk позволяет посмотреть информацию о дисках, их разделах, промежутки занимаемых разделом секторов, размер блока каждого диска. Информация о флэш накопителе также отображается.

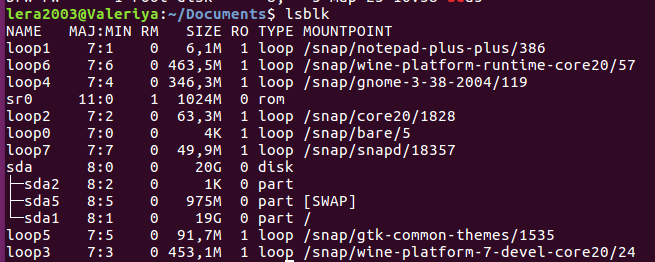
11.3. Привести «максимально возможное» дерево ФС, проанализировать, где это указывается

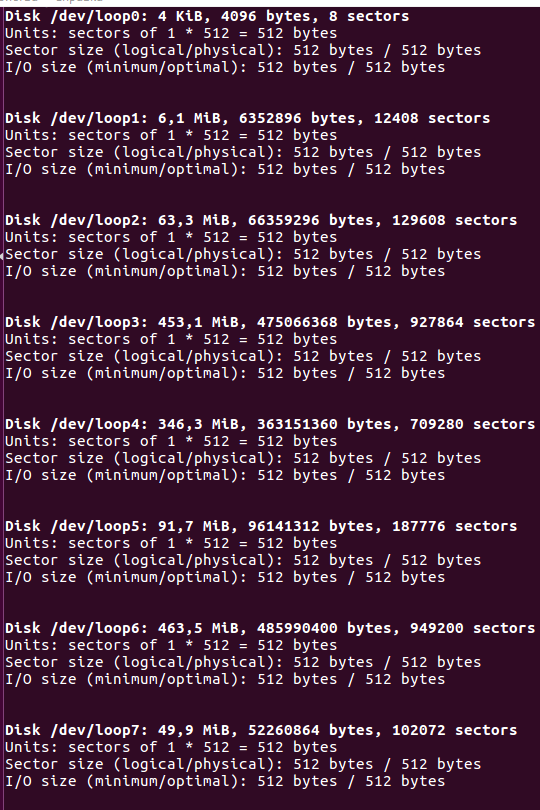
В файле /usr/include/linux/limits.h определена максимальная длина пути.

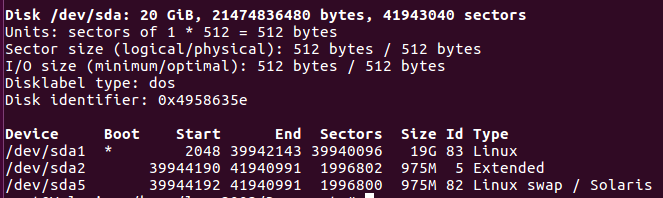


Максимально допустимая длина имени файла – 255 байт.

4096 байт – это длина полного пути до файла, включая имя.







Вывод.

Отображены название диска, его размер и количество секторов.

На моем устройстве присутствует один жесткий диск – dev/sda, который содержит 3 раздела: dev/sda1, dev/sda2, dev/sda5.

Файловые системы частей следующие: sda1 – ext4, sda2 - отсуствует, sda5 – swap. Типы разделов (под что они отведены): sda1 - Linux, sda2 - Extended, sda5 – Linux swap / Solaris.

Присутствует 8 устройств типа loop – виртуальные устройства, позволяющие делать компьютер доступным в качестве блочного устройства. Они не имеют никакого отношения к занятию оперативной памяти. Обычно они используются для монтирования образов дисков.

12. Проанализировать и пояснить принцип работы утилиты file.

Утилита определяет тип файла. Для этого она выполняет разные тесты, которые можно разделить на 3 группы:

Filesystem tests – основаны на анализе кода возврата системного вызова stat(). Программа проверяет не пустой ли файл, и не принадлежит ли он к одному из специальных типов файлов. Все известные типы файлов распознаются, если они определены в системном файле /usr/include/sys/stat.h.

Magic number tests – используются для проверки файлов, данные в которых записаны в определённом формате. В определённом месте в начале таких файлов записано магическое число, которое позволяет ОС определить тип файла. Все известные ОС магические числа по умолчанию хранятся в файле /usr/share/misc/magic.

Language tests – используются для анализа языка, на котором написан файл, если это файл в формате ASCII. Выполняется поиск стандартных строк, которые могут соответствовать определённому языку.

Первый тест, который завершится успешно, выводит тип файла. Типы файлов можно разделить на 3 основные группы:

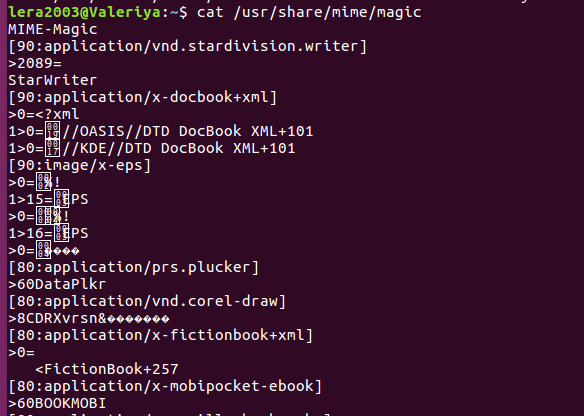
Текстовые – файл содержит только ASCII символы и может быть безопасно прочитан на терминале.

Исполняемые – файл содержит результаты компилирования программы в форме понятной ядру ОС.

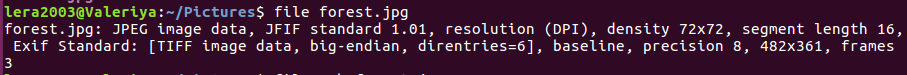
Данные – всё, что не подходит в первые 2 группы (обычно это бинарные или непечатаемые файлы). Исключение составляют well-known форматы, используемые для хранения бинарных данных.

12.1. Привести алгоритм её функционирования на основе информационной базы, размещение и полное имя которой указывается в описании утилиты в технической документации ОС (как правило, /usr/share/file/magic.\*), а также содержимого заголовка файла, к которому применяется утилита. Определить, где находятся магические числа и иные характеристики, идентифицирующие тип файла, применительно к исполняемым файлам, а также файлам других типов.

Информация о тестах на магические числа содержится в файле /usr/share/mime/magic.



12.2. Утилиту file выполнить с разными ключами.



-b, --brief — запрет на демонстрацию имен и адресов файлов в выводе команды;



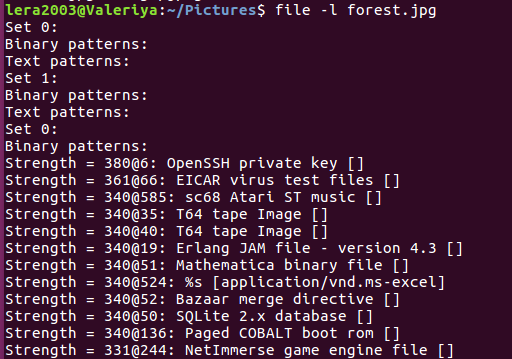
-i, --mime — определение MIME-типа документа по его заголовку;



--mime-type, --mime-encoding — определение конкретного элемента MIME;

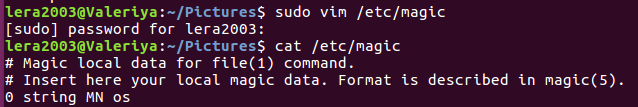


-l, --list — список паттернов и их длина;

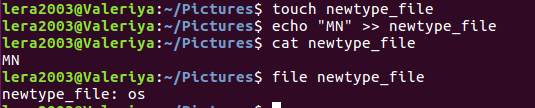


12.3. Привести экспериментальную попытку с добавлением в базу собственного типа файла и его дальнейшей идентификацией.

С помощью утилиты vim откроем файл /etc/magic и добавим новый тип файла os, который будет определено магическим число MA в своем содержании.



Создадим новый файл с магическим числом и определим его тип.



Файл определяется с типом os, следовательно, новый тип был успешно создан.

## Вывод.

При выполнении данной лабораторной работы были изучены принципы

организации файловой системы ОС Linux и получены навыки работы с ней.

Существует 6 типов файлов: обычный файл, каталог, специальный файл

устройства, именованный канал (FIFO), символическая ссылка и сокет.

Каждый файл имеет уникальный индексный дескриптор, указывающий

на метаданные этого файла, в которых содержится служебная информация о

файле (расположение данных файла на диске, владелец файла и права доступа к файлу).

Утилита od позволяет просмотреть дамп памяти файла в разных форматах.

Были рассмотрены все возможные способы формирования ссылок.

Утилита find позволяет искать файлы по заданным критериям и производить заданные действия с найденными файлами.

Атрибут SUID, установленный на исполняемом файле, позволяет любому пользователю, который запустил этот файл, совершать действия от имени владельца файла. Это может быть использовано, например, когда обычный пользователь желает изменить свой пароль.

Утилита df позволяет получить информацию о файловой системе, а утилита file – определить тип файла.

**Список источников.**

* 1. <https://selectel.ru/blog/directory-structure-linux/>
  2. [Losst - Linux Open Source Software Technologies](https://losst.pro/)
  3. [Linux - начинающим. Часть 4. Работаем с файловой системой. Теория - Записки IT специалиста (interface31.ru)](https://interface31.ru/tech_it/2016/01/linux-nachinayuschim-chast-4-rabotaem-s-faylovoy-sistemoy-teoriya.html)