МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

ОТЧЁТ

по лабораторной работе №3

«Работа пользователя в LINUX»

по дисциплине  
«Эксплуатация современных операционных систем»

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кочешков А.А.

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Игнаков К. М.

19-В-2

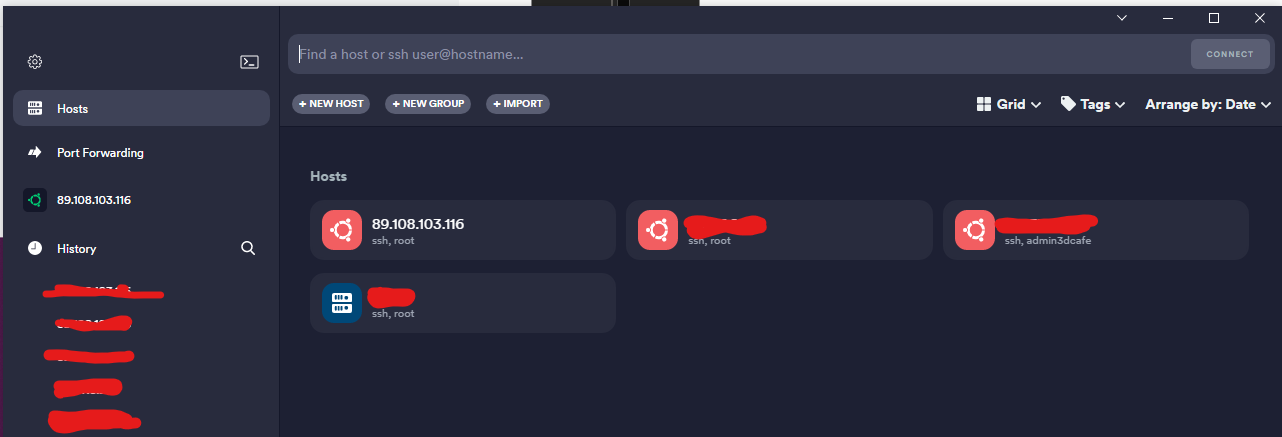
Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

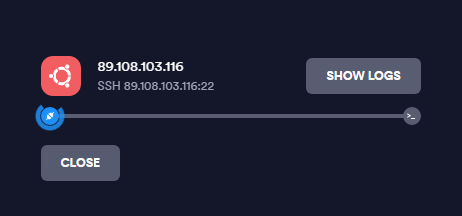
С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022г.

Ход работы

В ходе выполнения лабораторной работы, я арендовал мощности Reg.ru с предустановленным дистрибутивом Ubuntu 20.04 LTS, 15 Гб, 1 ядро. Подключение к серверу происходит по ssh для удобства используется программа Termius.



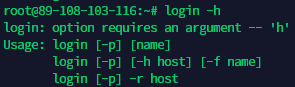


В Linux есть ровно один пользователь, полномочия которого в системе принципиально отличаются от полномочий остальных пользователей — это пользователь с идентификатором "0". Обычно учётная запись пользователя с UID=0 называется root. Root — это «администратор» системы Linux, учётная запись для root обязательно присутствует в любой системе Linux, даже если в ней нет никаких других учётных записей. Пользователю с таким UID разрешено выполнять любые действия в системе, а значит, любая ошибка или неправильное действие может повредить систему, уничтожить данные и привести к другим печальным последствиям.

*Цель*: изучить правила входа в систему и завершения работы.

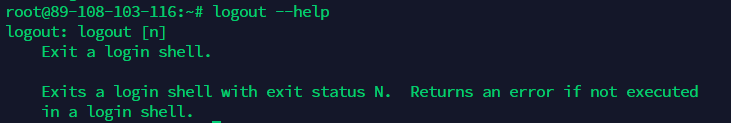
Команда *login*

login — это механизм идентификации и проверки идентичности, вызываемый монитором порта (port monitor), обычно, ttymon, в начале каждого терминального сеанса. Он обеспечивает средства идентификации пользователей в системе и проверки идентичности пользователя. В качестве последнего шага процедуры регистрации вызывается определенная программа, обычно командный интерпретатор sh.



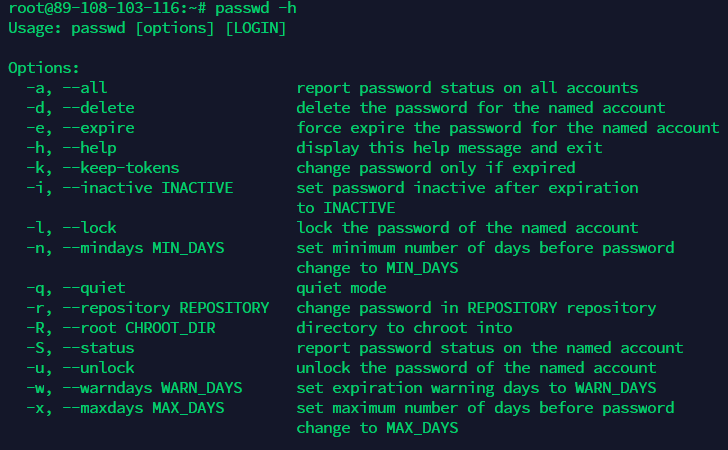
Команда *logout*

logout или lo - завершить работу с командной оболочкой. является функцией, связанной с закрытием соединения RSH. В ответ на эту команду возобновляется приглашение к регистрации в системе.

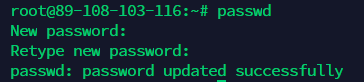


Команда *passwd*

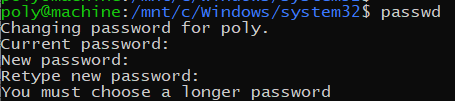
passwd - изменить пароль пользователя



При выполнении этой команды запрашивается новый пароль и подтверждение нового пароля в случае, если подтверждения пароля произошли ошибки, то пароль не будет изменен.

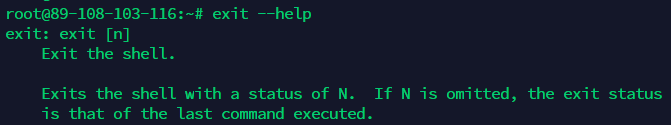


Если пароль окажется слишком коротким, будет выведено сообщение о том, что необходимо ввести более сложный пароль.



Команда *exit*

exit – это стандартная функция C, которая завершает текущий процесс (или Task в vxWorks). Это означает, что вы просто завершили задачу оболочки ядра (а не только RSH-соединение).



Команда *shutdown*

shutdown - выключить компьютер или спланировать выключение. Для данной команды можно использовать множество различных аргументов:

* --help - вывести справку по программе
* -H, --halt - отключить питание, не завершая процессы и не перемонтируя файловые системы
* -P, --poweroff - нормальное выключение
* -r, --reboot - перезагрузка
* -k - не выполнять реальных действий, а только вывести сообщение
* --no-wall - выключить компьютер, но не выводить сообщение
* -c - отменить запланированное выключение linux из командной строки
* -h – спланировать выключение через какое-то время или в указанную дату, например:

Выключить компьютер через 5 минут:

* sudo shutdown -h +5 "Компьютер будет выключен через 5 минут"

Выключить компьютер в 12 ночи:

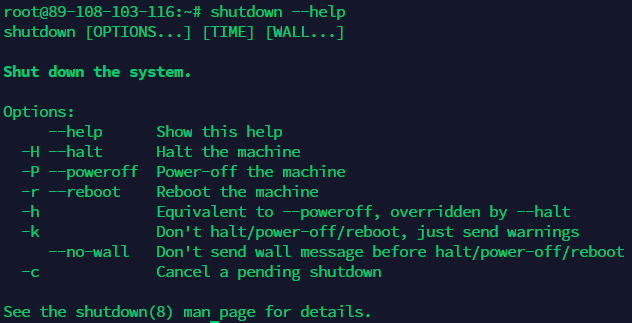
* sudo shutdown -h 21:00

Перезагрузить компьютер прямо сейчас:

* shutdown -r now

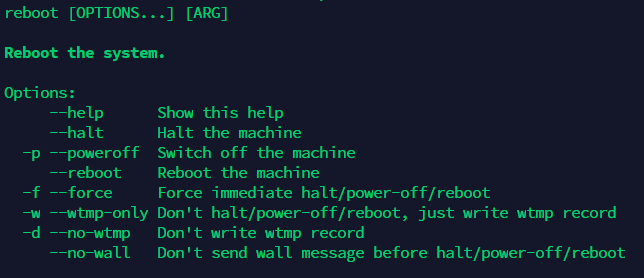
Чтобы отменить отключение, необходимо ввести команду:

* sudo shutdown -c



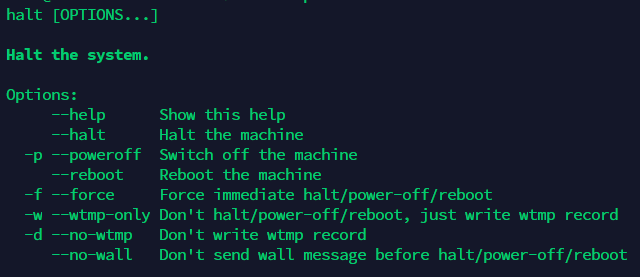
Команда *reboot*

Команда reboot обычно используется для перезагрузки системы, но также можно выключать компьютер. Данная команда эквивалентна команде shutdown -r now.



Команда *halt*

halt – немедленно выключить компьютер. Данная команда используется, когда требуется завершить работу и выключить компьютер.



Команда *Ctrl+Alt+Del* – выход из системы (вызов команды logout)

Команда *Alt+Ctrl* – вызов нового окна терминала

Команда *Alt+Tab* – переключение между окнами

Команда *Alt+Fn* (где n – это номер консоли) – комбинации клавиш для работы в консоли:

* Alt+F1 – выбор активного окна
* Alt+F2 – аналог «Выполнить» в Windows
* Alt+F4 – закрывает окно терминала
* Alt+F7 – захват окна для изменения его расположения
* Alt+F10 – свернуть окно

Вывод: в ходе выполнения, я познакомился с правилами входа в систему и выхода из системы, завершения работы. Изучены некоторые консольные команды, отвечающие за вход и выход из системы, выключение, перезагрузка компьютера.

Основные информационные команды.

*Цель*: научиться пользоваться диалоговым руководством man, apropos, ознакомиться со справочной системой в формате файлов HOW-TO.

Команда *man*

man – команда сокращенная от слова Manual (руководство) – предназначенная для форматирования и вывода справочных страниц.

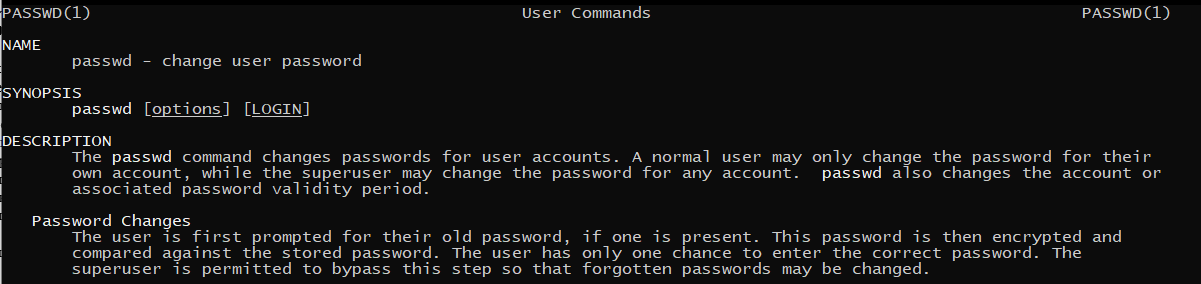
Справочные страницы поделены на 8 стандартных разделов и один дополнительный. Каждый из разделов соответствует той или иной тематике в рамках установленной операционной системы.

1. Прикладные программы и команды оболочки
2. Системные вызовы ядра (функции языка Си)
3. Библиотечные вызовы (функции различных библиотек, установленных в систему)
4. Специальные файлы (находящиеся обычно в каталоге /dev)
5. Форматы файлов и соглашения
6. Игры
7. Различные описания, соглашения и прочее
8. Команды администрирования системы, которые обычно запускаются от имени суперпользователя
9. Ядро операционной системы (нестандартный раздел)

Номер раздела в команде man указывается вторым аргументом, перед названием справочной страницы. Если номер раздела опущен, то поиск справочной страницы ведётся по всем разделам по порядку.

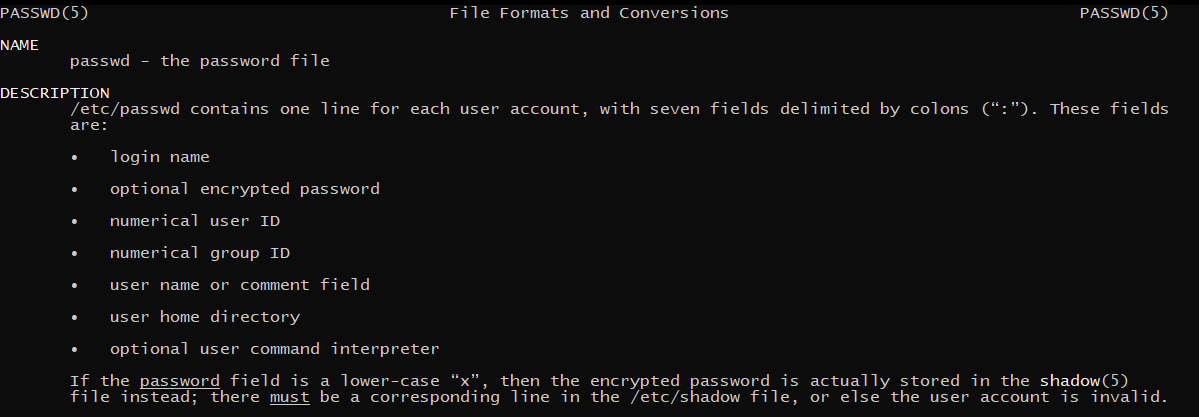
Например:

man passwd (эквивалентен man 1 passwd)

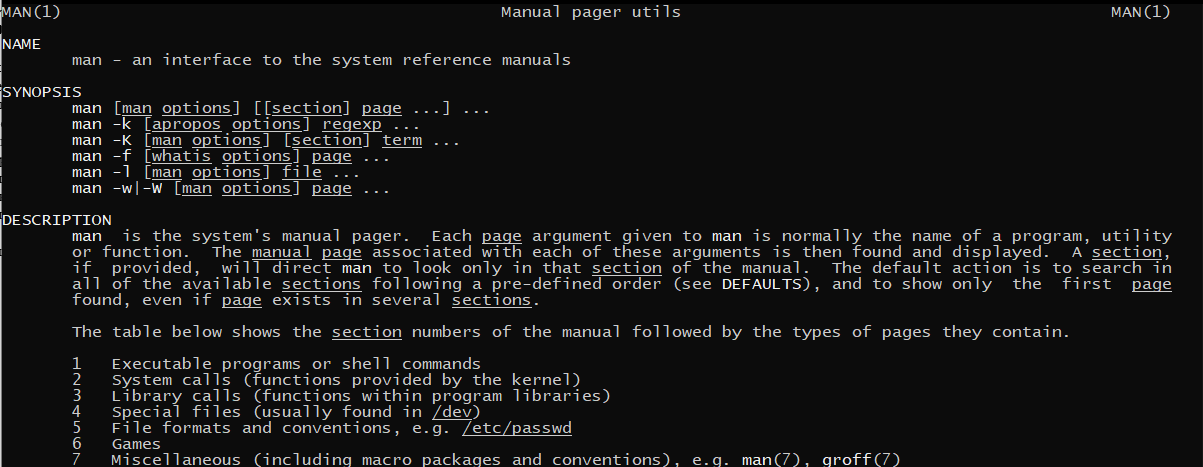


Для того, чтобы выйти из справки, не обходимо нажать «q».

man 5 passwd



Для получения детальной инструкции по использованию команды man используется конструкция: man man



Структура руководства:

В начале страницы можно увидеть название, краткое описание команды и синтаксиса и описание для чего предназначена программа. Есть определённые правила оформления страницы руководства. Есть руководства по командам, программам, функциям и т.д. Не во всех руководствах есть эти заголовки, так как некоторые из них применимы только к конкретным командам.

Список заголовков:

Название (Name)- название команды, по которой просматривается руководство

* Синопсис (Synopsis)- краткое описание команды и синтаксиса
* Конфигурация (Configuration)- детали настройки для устройства
* Описание (Description)- описание основного назначения программы
* Опции (Options)- опции которые принимает команда
* Выходной статус (Exit Status)- возможные значения, возвращаемые командой при завершении работы
* Возвращаемое значение (Return Value)- если руководство запущено по какой-то библиотеке, то это указывает на значение, которое вернет библиотека функции, вызвавшая ее
* Ошибки (Errors)- список всех значение, которые может принимать errno в случае ошибки выполнения команды
* Окружение (Environment)- список переменных окружения, которые относятся к команде или программе
* Файлы (Files)- список файлов, которые использует команда или программа
* Атрибуты (Attributes)- список различных атрибутов команды
* Версии (Versions)- список изменений в ядре Linux или библиотеке, которую использует команда
* Соответствие (Conforming to)- описание любых стандартов, которым может соответствовать команда
* Заметки (Notes)- дополнительные заметки
* Ошибки (Bugs)- известные ошибки
* Примеры (Examples)- один или несколько примеров использования команды
* Авторы (Authors)- авторы, которые разработали и поддерживают команду
* Просмотрите также (See Also)- рекомендуемые материалы по команде

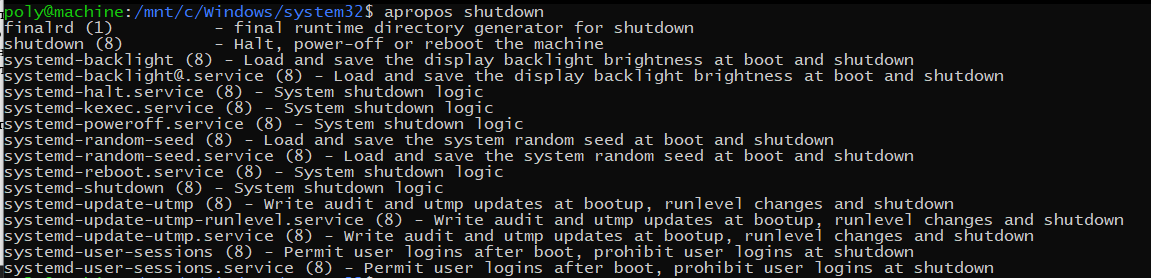
Команда *apropos*

apropos – это команда для поиска файлов man-страниц в Unix и Unix-подобных операционных системах. Она особенно полезна при поиске команд, не зная их точных имен.

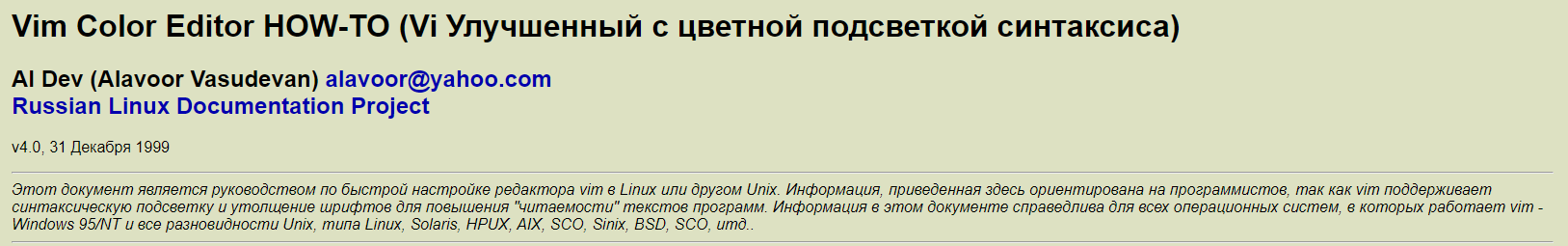
Команда apropos часто используется в качестве оболочки для man -k command для поиска в разделах «name» всех справочных страниц указанной строки/строк (называемых ключевыми словами). Результатом является список всех справочных страниц, содержащих поисковый запрос в названии или описании. Это часто бывает полезно, если вы знаете, какое действие требуется, но не помните точную команду или имя страницы.

Apropos обычно ищет в предварительно скомпилированной базе данных, которая используется совместно с whatis, команду для получения краткого описания конкретной команды, точное имя которой уже известно.

Пример работы:



Файлы справки HOW-TO имеют вид текстовых файлов с описанием алгоритма для выполнения определенной задачи. Могут храниться как централизовано, так и в папке с программой, по которой требуется обучающая информация. Но такой способ справки уже устарел, т.к. многие программы имеют встроенную систему документации, не доступную, ни через man, ни через info. Такие файлы можно найти на сайте " https://www.opennet.ru/docs/5.shtml". Здесь содержатся подробные инструкции по установке и использовании файлов, например: «Vim Color Editor HOW-TO (Vi Улучшенный с цветной подсветкой синтаксиса)».



Файлы HOW-TO рассчитаны на предоставление информации по разрешению возникших проблем или настройке тех или иных компонент. Такие файлы содержат аннотацию, структуру, описание и алгоритм действий.

*Вывод*: в ходе выполнения пункта, были изучена работа с командой вывода справки man, вспомогательной командой apropos и ознакомление со справочной системой HOW-TO.

Управление учетными записями пользователей.

*Цель*: получить информацию о пользователях с помощью команд who, w, id, logname, finger. Создать новую учетную запись командой adduser. Ознакомиться с форматом и использованием файла паролей /etc/passwd, /etc/shadow. Привести примеры назначения учетных записей псевдопользователей. Определить состав и свойства домашнего каталога пользователя, профиля пользователя, профиля по умолчанию /etc/skel.

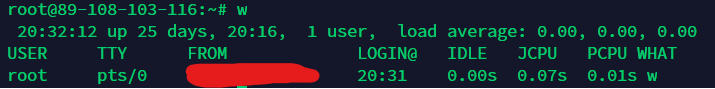
Команда *who*

who – получение информации об активных пользователях, которые подключены к системе, в том числе и о терминальных сессиях, через которые происходит подключение.



Команда *w*

w – получить более детальную информацию о пользователях и используемых ими терминалах. Также выводит на экран текущий процесс, запущенный в терминале, время работы пользователя в терминале.



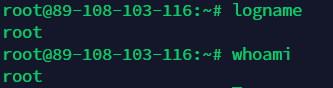
Команда *id*

id – вывести информацию о пользователе: идентификаторы пользователя, его группы, некоторые права и настройки пользователя



Команда *logname*

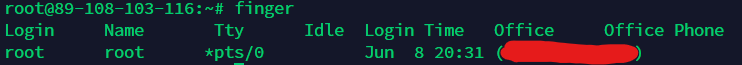
logname – выводит имя пользователя, который зарегистрирован в системе на данный момент. Эквивалентна команде whoami.



Команда *finger*

finger – вывести информацию о пользователе, вошедшим в систему. На некоторых дистрибутивах эта утилита не установлена, её можно установить с помощью команды “apt install finger”, если используется система Debian или Ubuntu.

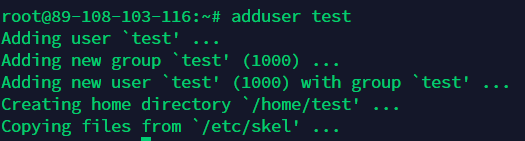




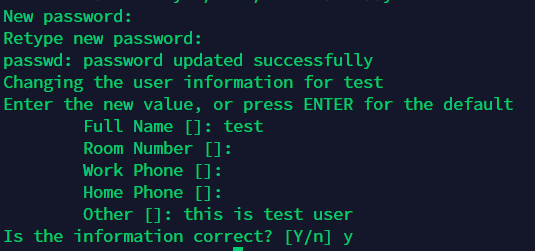
Команда *adduser*

Adduser – добавляют пользователей в систему, исходя из параметров, заданных в командной строке и информации из файла /etc/adduser.conf.

Создадим тестового пользователя test, с помощью команды adduser test.

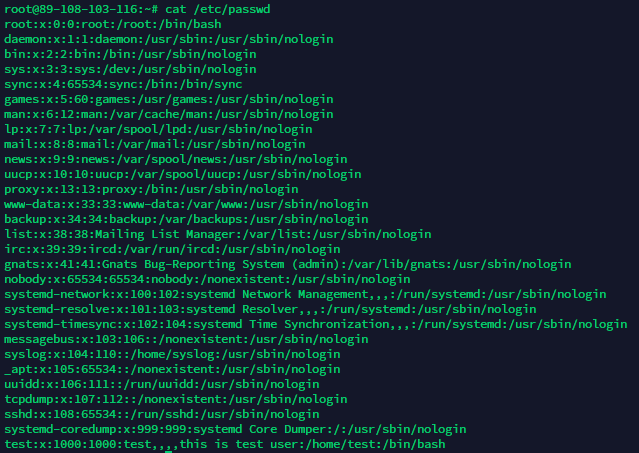


В ходе выполнения команды был создан пользователь test, создана новая группа test, пользователь test был добавлен в группу test, также была создана директория /home/test. Далее запрашивается пароль для нового пользователя, полное имя, номер комнаты, рабочий и домашний телефоны, другая информация о пользователе, данные можно оставить пустыми:



/etc/passwd – файл, содержащий в текстовом формате список всех пользовательских учётных записей, имеющихся в системе. Является первым и основным источником информации о правах пользователя операционной системы.

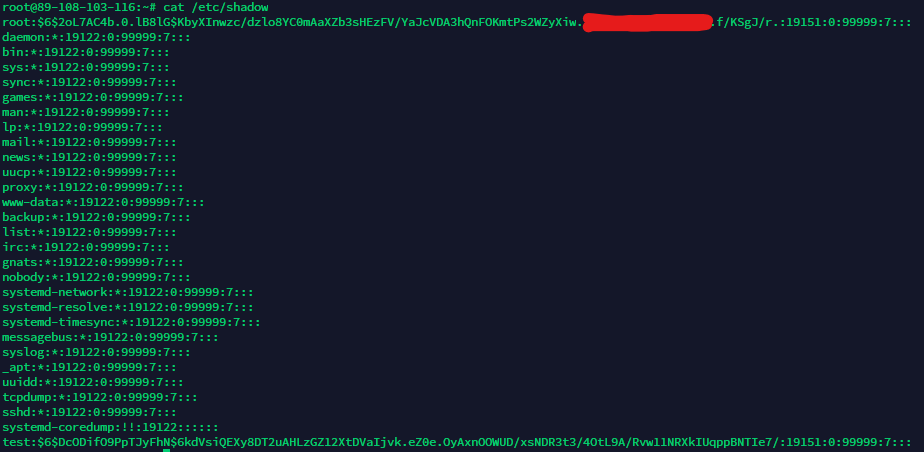
Для того, чтобы посмотреть содержимое файла, можно воспользоваться утилитой cat. Команда “cat /etc/passwd” покажет содержимое файла passwd в каталоге /etc/.



Каждая строка файла описывает одного пользователя и содержит семь полей, разделённых двоеточиями:

1. Имя пользователя (логин)
2. Хеш пароля
3. Идентификатор пользователя (uid)
4. Идентификатор группы (gid)
5. Информационное поле GECOS (полное имя пользователя, номера телефонов и т.д.)
6. Начальный (домашний) каталог
7. Регистрационная оболочка (shell)

/etc/shadow — это текстовый файл, содержащий информацию о паролях пользователей системы. Посмотрим содержание этого файла. Для этого воспользуемся командой "cat /etc/shadow".



Как и в файле passwd, каждое поле в файле shadow отделяется двоеточием:

1. Имя пользователя. Совпадает с именем пользователя в файле /etc/passwd
2. Пароль (зашифрованный). Пустая запись означает, что для входа пароль не нужен, запись "\*'' - вход заблокирован
3. Количество дней (с 1 января 1970), когда пароль был сменен в последний раз.
4. Число дней до смены пароля (0 показывает, что он может быть сменен всегда)
5. Число дней, после которых пароль должен быть сменен (99999 показывает, что пользователь может не менять пароль фактически никогда)
6. Число дней, в течение которых пользователь получает предупреждения о необходимости пароль сменить (7 для полной недели)
7. Число дней после окончания действия пароля, когда еще можно работать. Если пароль не сменить, после данного срока аккаунт будет заблокирован
8. Число дней, начиная с 1 января 1970, после которых пароль будет заблокирован.
9. Зарезервировано для возможного будущего использования

Псевдопользователи

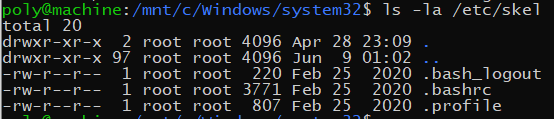
В изображении выше показано куда больше пользователей, чем ожидается, но это все пользователи системы. Для того, чтобы отделить пользователей, которых мы сами создали, от пользователей системы, необходимо посмотреть на идентификатор пользователя UID:

* От 1 до 500 – специальный пользователь
* ≥ 1000 – обычный пользователь
* От 501 до 999 – реальный пользователь

Виды псевдопользователей:

* Nobody – предназначен для запуска программ, служб и т.п., которым не требуются какие-либо специальные разрешения
* Daemon – используется серверными процессами системы
* Sys – владеет системными файлами
* Adm – владеет файлами бюджета
* Bin – владеет исполняемыми файлами пользователя
* Uucp – используется UUCP
* Lp – используется подсистемами 1р или lpd
* Usenet, audit, cron, mail, new – обслуживают связанные с ними процессы и файлы

/etс/skel - файл, который содержит в себе файлы и настройки поведения командной оболочки Bash. Посмотреть содержимое домашнего каталога с скрытыми файлами можно с помощью команды “ls -la /etс/skel”



Включает в себя 3 фала конфигурации:

.bash-logout

.bashrc — настройка поведения консоли

.profile — размещение пользовательских исполняемых файлов по умолчанию

Все файлы этой директории копируются в домашнюю папку нового пользователя (/home/<username>) при создании командой adduser или useradd. Точка перед именем файла – признак скрытого файла. Домашний каталог служит в качестве хранилища для личных файлов, каталогов и программ пользователя, включая личные файлы конфигурации. /etc/skel позволяет создать домашний каталог по умолчанию для всех новых пользователей на компьютере или в сети и, таким образом, убедиться, что все пользователи начинают с одних и тех же настроек или среды.

*Вывод*: в ходе выполнения пункта были изучены команды для создания нового пользователя. Была создана учетная запись test, были изучены спецификации UID, ознакомился с особенностями файлов /etc/passwd и /etc/shadow.

Работа с устройствами и файловой системой.

*Цель*: изучить принцип именования устройств, назначение и применение специальных файлов устройств в классическом формате Unix и в современном формате udev. Найти в каталоге /dev специальные файлы основных устройств компьютера и составить отчет об аппаратной конфигурации. Сравнить распределение аппаратных ресурсов (I/O Ports, IRQ, Mem) в linux с аналогичными данными в Windows (из 2 работы). Ознакомиться с основными командами работы с дисками: mount, umount, df, mkfs, fsck, fdisk. Изучить свойства и параметры операции монтирования файловых систем. Выполнить монтирование файловой системы на сменном носителе. Рассмотреть описание автоматически монтируемых файловых систем в файле /etc/fstab. Объяснить использованный в fstab формат ссылки на устройство корневой файловой системы.

Различают несколько видов файлов: обычные файлы, каталоги и специальные файлы. К обычным файлам относят файлы с важными данными.

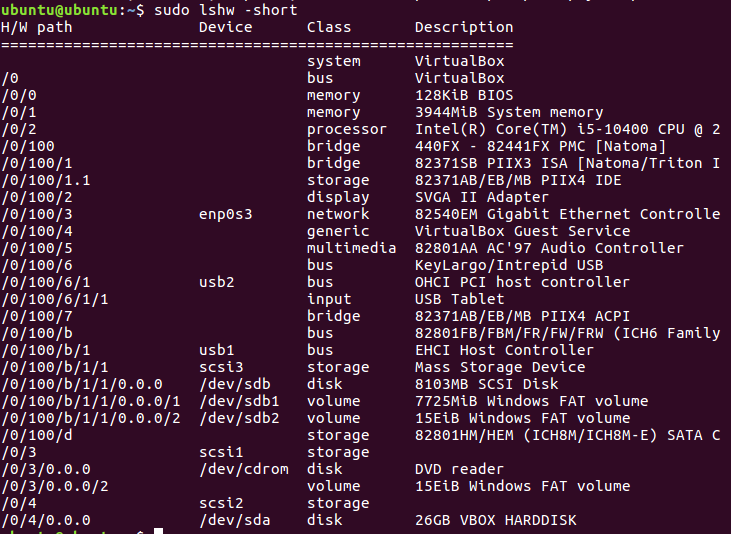
Каталоги – это файлы, в качестве данных которых выступают списки других файлов и каталогов. В данных каталога осуществляется связь имени файла с его индексным дескриптором. Поэтому один и тот же файл может существовать под разными именами и/или в разных каталогах: все имена будут связаны с одним и тем же индексным дескриптором (жесткие ссылки). Файлы всегда содержатся в каталогах.

А специальные файлы – это механизм использования ввода/вывода. Большинство специальных файлов находятся в /dev. Они используются для обозначения физических устройств компьютера. Размера у специального файла не существует. Так как этот файл обозначает физическое устройство. Другими словами, это ссылка на соответствующий драйвер.

Специальные файлы делятся на два типа:

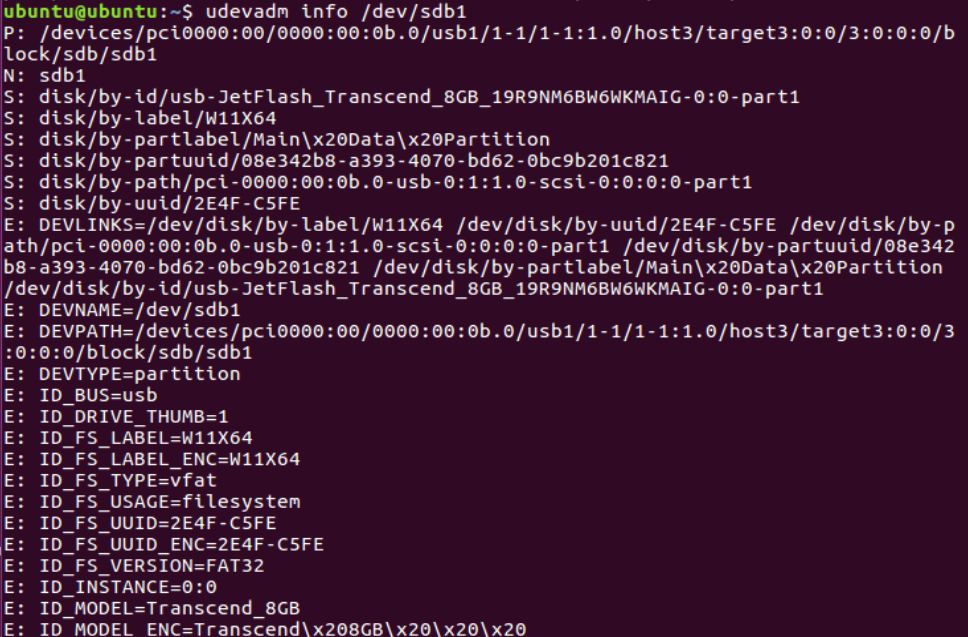
* Блочные - запись и чтение допускаются только блоками
* Символьные - можно читать и писать отдельными байтами.

С помощью команды "sudo lshw -short" получим список классов устройств используемого компьютера.



udev – это механизм, используемый для создания и именования узлов устройств в каталоге /dev, соответствующих устройствам, присутствующим в системе.

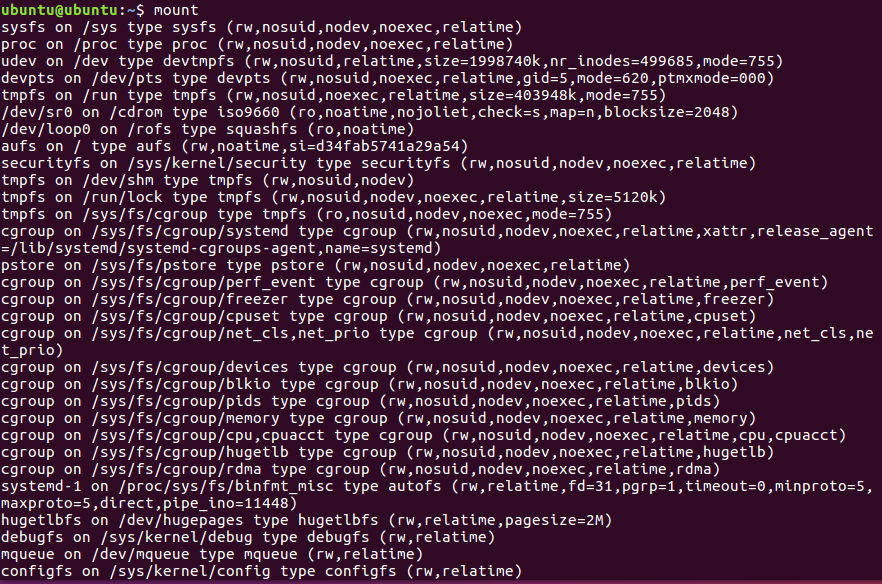
Для того, чтобы узнать атрибуты устройства, необходи ввести команду “udevadm info /dev/sd<устройство>”, в моем случае это sdb1.



Команда *mount*

mount – монтирует файловую систему. Все файлы, доступные в Unix системах, составляют иерархическую файловую структуру, которая подобна растущему дереву имеет ветки (каталоги) и листья (файлы в каталогах). Корень этого большого дерева обозначается как “/”. Физически файлы могут располагаться на различных устройствах. Команда mount служит для подключения файловых систем разных устройств к этому большому дереву.

Если в консоль ввести команду mount, то будет выведен список всех смонтированных файловых систем в текущий момент и параметры монтирования.



Некоторые из них:

* sysfs – с помощью этой файловой системы вы можете задавать различные настройки ядра во время выполнения
* tmpfs – позволяет размещать любые пользовательские файлы в оперативной памяти компьютера. Достаточно создать блочное устройство нужного размера, затем подключить его к папке, и вы можете писать файлы в оперативную память
* procfs – по умолчанию смонтирована в папку proc и содержит всю информацию о запущенных в системе процессах, а также самом ядре
* securityfs – эта файловая система предназначена для использования модулями безопасности
* ext2, ext4 – это стандартные файловые системы Linux

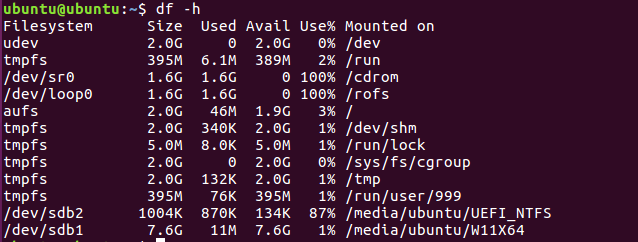
Команда *unmount*

unmount – применяется для размонтирования файловых систем, предварительно смонтированных посредством вызова утилиты mount.

Команда *df*

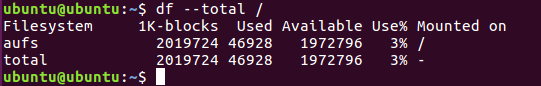
df – выводит имена файлов устройств разделов, полный объем их пространства, объем занятого и свободного пространства, процентный показатель использования разделов, а также точки их монтирования.

Команда “df -h” выведет справку в удобном читаемом виде.



Здесь записи представлены в виде таблицы, каждый столбец имеет свое название: тип файловой системы, ее размер, сколько использовано и доступно, также сколько использовано в процентном соотношении и где смонтировано.

Для того, чтобы узнать сколько в целом использует ресурсов система:

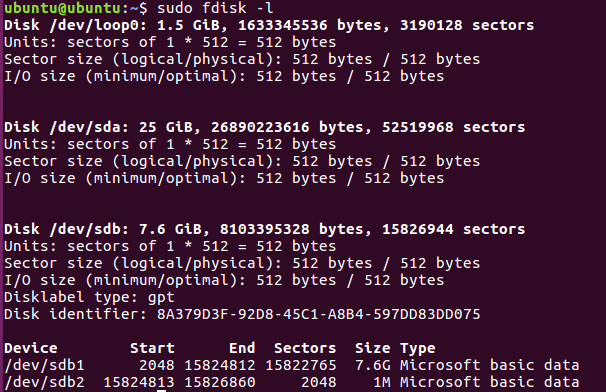


Команда mkfs – позволяет создавать и форматировать файловые системы.

Команда *fsck*

fsck – используется для проверки и исправления ошибок одной или нескольких файловых систем. Также fsck параллельно проверяет данные на разных физических дисках, чтобы сократить общее время, необходимое для полной проверки всех дисков. Если файловых ввести команду fsck без параметров, то она по порядку проверит файловые системы, указанные в /etc/fstab.

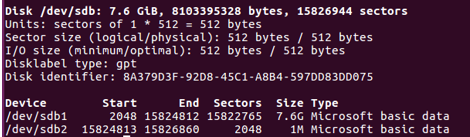
Команда fdisk – используется для управления разделами жёсткого диска.



Здесь выводится подробная информация об устройствах: имя, список разделов, формат таблицы разделов, список разделов, размер раздела и его файловую систему.

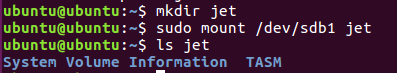
Выполнить монтирование на съемном носителе.

Для начала необходимо вставить флэш-накопитель в компьютер.

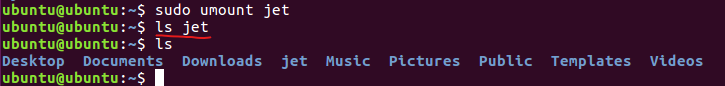


Создадим папку "jet" в домашнем каталоге, смонтируем в нее файловую систему флешки с помощью команды "mount" и посмотрим содержимое этой папки командной "ls".

Создаем новый каталог jet, с помощью команды mkdir, после этого монтируем в нее наш флэш-накопитель.



Теперь для размонтирования необходимо воспользоваться утилитой umount. Как видим, папка jet “пустая”, размонтирование прошло успешно.



*fstab* — это файл, содержащий список устройств хранения информации и параметры монтирования. Различные накопители, которые необходимо автоматически присоединить во время загрузки операционной системы, по порядку перечисляются в файле fstab.



В каждой строчке содержится следующая информация:

* Устройство — это раздел диска, который нужно монтировать
* Файловая система – указывает в какой файловой системе нужно монтировать это устройство
* Точка монтирования – куда нужно монтировать это устройство
* Опции - параметры монтирования файловой системы
* Проверка – указывает в какой очереди нужно проверять устройство на ошибки: 1 - в первую очередь, 2 - во вторую, 0 - не проверять
* Резерв – указывает нужно ли делать резервную копию раздела, может принимать значения только 0-нет и 1-да

*Вывод*: в ходе работы в этом пункте, были изучены основы работы с монтированием устройств, получение информации о файловой системе с помощью разных команд, рассмотрены автоматически монтируемых файловых систем, которые хранятся в файле /etc/fstab.

Управление файлами

*Цель*: изучить и опробовать:

- получение информации о файлах и каталогах: форматы команды ls;

- команды файловых операций: cd, pwd, cp, mv, rm, mkdir, rmdir;

- операции над группами файлов;

- создание прямой и символической ссылки ln;

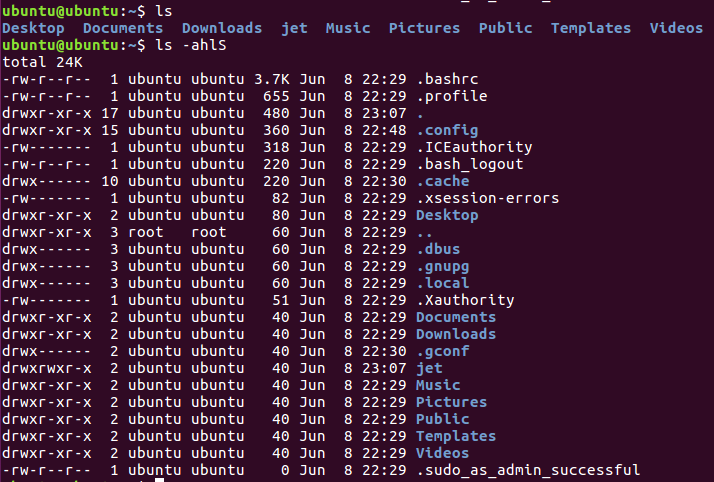
- команды вывода cat,tee,more,file, поиска find, фильтра grep;

- перенаправление потоков ввода-вывода (<, >, >>) и каналы (|);

Команда *ls*

ls – сокращенно от list, показывает файлы и папки в текущей директории. Для того чтобы узнать какие папки и файлы есть в другой директории, необходимо передать путь к ней. Изначально команда ls показывает только содержимое директории, для более подробной информации существует множество аргументов:

* -a – отображать все файлы, включая скрытые, это те, перед именем которых стоит точка;
* -A – не отображать ссылку на текущую папку и корневую папку . и ..;
* --author – выводить создателя файла в режиме подробного списка;
* -b – выводить Escape последовательности вместо непечатаемых символов;
* --block-size – выводить размер каталога или файла в определенной единице измерения, например, мегабайтах, гигабайтах или килобайтах;
* -B – не выводить резервные копии, их имена начинаются с ~;
* -c – сортировать файлы по времени модификации или создания, сначала будут выведены новые файлы;
* -C – выводить колонками;
* --color – включить цветной режим вывода, автоматически активирована во многих дистрибутивах;
* -d – выводить только директории, без их содержимого, полезно при рекурсивном выводе;
* -D – использовать режим вывода, совместимый с Emacs;
* -f – не сортировать;
* -F – показывать тип объекта, к каждому объекту будет добавлен один из специализированных символов \*/=>@|;
* --full-time – показывать подробную информацию, плюс вся информация о времени в формате ISO;
* -g – показывать подробную информацию, но кроме владельца файла;
* --group-directories-first – сначала отображать директории, а уже потом файлы;
* -G – не выводить имена групп;
* -h – выводить размеры папок в удобном для чтения формате;
* -H – открывать символические ссылки при рекурсивном использовании;
* --hide – не отображать файлы, которые начинаются с указанного символа;
* -i – отображать номер индекса inode, в которой хранится этот файл;
* -l – выводить подробный список, в котором будет отображаться владелец, группа, дата создания, размер и другие параметры;
* -L – для символических ссылок отображать информацию о файле, на который они ссылаются;
* -m – разделять элементы списка запятой;
* -n – выводить UID и GID вместо имени и группы пользователя;
* -N – выводить имена как есть, не обрабатывать контролирующие последовательности;
* -Q – брать имена папок и файлов в кавычки;
* -r – обратный порядок сортировки;
* -R – рекурсивно отображать содержимое поддиректорий;
* -s – выводить размер файла в блоках;
* -S – сортировать по размеру, сначала большие;
* -t – сортировать по времени последней модификации;
* -u – сортировать по времени последнего доступа;
* -U – не сортировать;
* -X – сортировать по алфавиту;
* -Z – отображать информацию о расширениях SELinux;
* -1 – отображать один файл на одну строку.

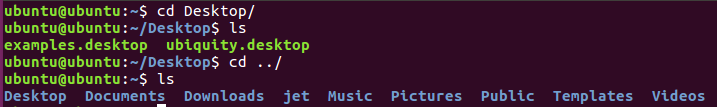


На примере первый вызов показал только папки, во второй команде использовались флаги -a -h -l -S, теперь отображаются скрытые файлы, с размером файлов в читаемом для человека виде, владелец, группа, дата создания и отсортированы по размеру от большего к меньшему.

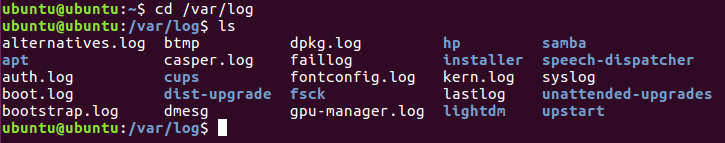
Команда *cd*

cd – от change directory – сменить директорию

Для примера попробуем перейти в папку “Desktop”, а затем вернуться обратно:



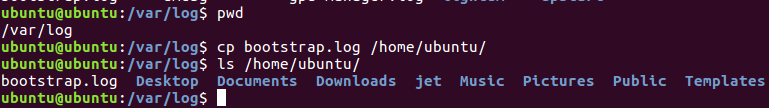
Команда “cd ../” вернет пользователя на родительскую директорию. Если пользователю необходимо перейти в конкретную директорию, и он знает её путь, он может его передать от текущего расположения или от корня “/”.



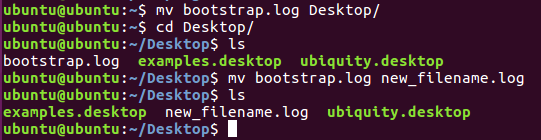
Команда *pwd* – покажет текущую папку от корня.



Команда *cp* – копировать файл.



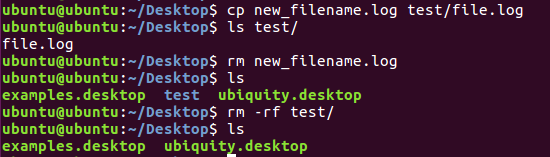
Команда *mv* – переместить файл, так же эту команду часто использую для того, чтобы переименовать файл.



Команда *mkdir* – создать каталог.

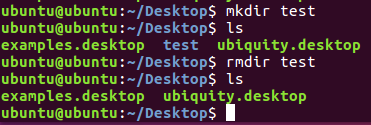


Команда *rm* – удаление файлов и каталогов. Для примера скопируем файл new\_filename в test, после этого удалим его и удалим каталог test со всем содержимым.



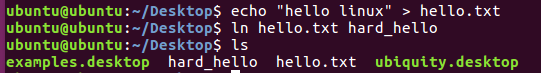
Для удаления каталога с содержимым использовались 2 аргумента -r и -f, -r – производить удаление рекурсивно по всем подкаталогам, -f – не возвращать код ошибочного завершения, если ошибки были вызваны несуществующими файлами; не запрашивать подтверждения операций.

Команда rmdir – удаление каталога.



Создание жестких и символических ссылок:

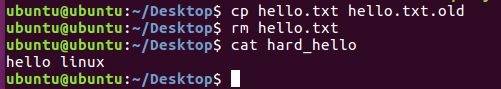
Команда ln – устанавливает связь между файлом и именем файла. Чтобы создать жесткую ссылку:



Первой командой был создан файл hello.txt и в него был записан текст “Hello linux” после этого с помощью команды ln создали жесткую ссылку на первоначальный файл. С помощью команды cat посмотрим содержимое hard\_hello:

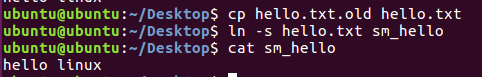


Попробуем сохранить исходный файл и удалить существующий, чтобы посмотреть, что будет находится внутри жесткой ссылки после удаления:



Как видим после удаления файла, содержимое сохранилось, как когда мы видели в Windows 7.

Создадим символическую ссылку:



Попробуем удалить файл и посмотреть содержимое:



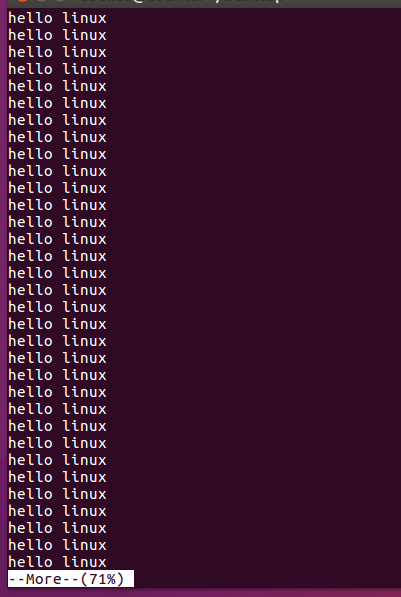
Получили ошибку, что указанный файл не найден.

Команды вывода:

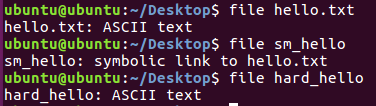
Команда cat – считывает данные из файлов и выводит их содержимое, пример её работы был приведен выше.

Команда tee – считывает стандартный ввод, затем записывает его содержимое в стандартный вывод.

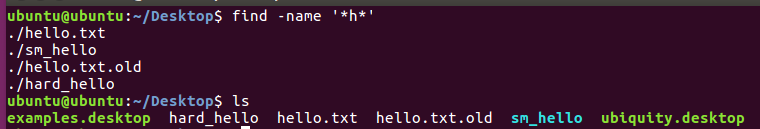
Команда more – постраничный просмотра файлов в терминале. Создадим файл "test.txt", запишем в него “Hello linux” много раз и выполним постраничный просмотр файла, нажимая клавишу "Пробел":



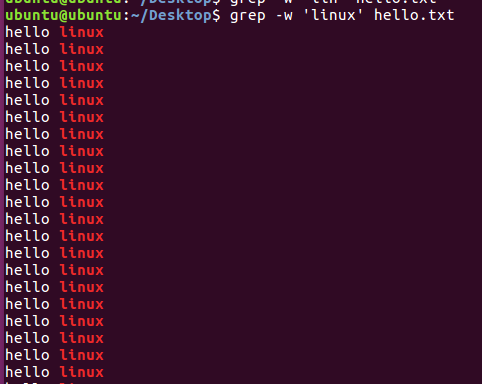
Команда file – отображает тип файла.



Команда поиска find – поиск файлов по имени и другим свойствам. Может производить поиск в одном или нескольких каталогах с использованием критериев, заданных пользователем. Попробуем найти файлы, в которых есть буква “h”:

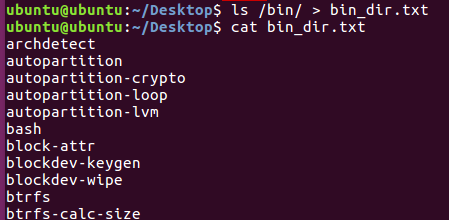


Команда фильтра grep – выполняет поиск образца в текстовых файлах и выдает все строки, содержащие этот образец. Выполним поиск предложения из файла, среди огромного количества текста с помощью команды “grep -w 'linux' hello.txt”



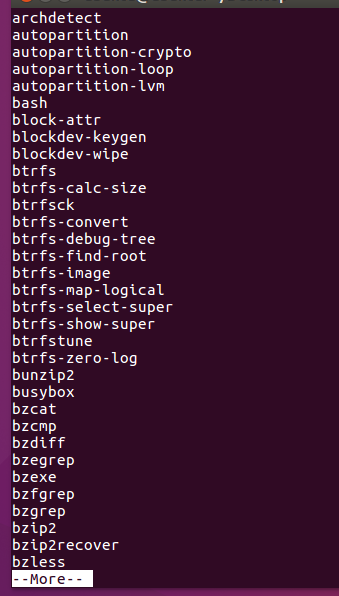
Перенаправление потоков ввода и вывода:

* “> file” – пример этого перенаправления был показан выше, когда записывали данные в файл.



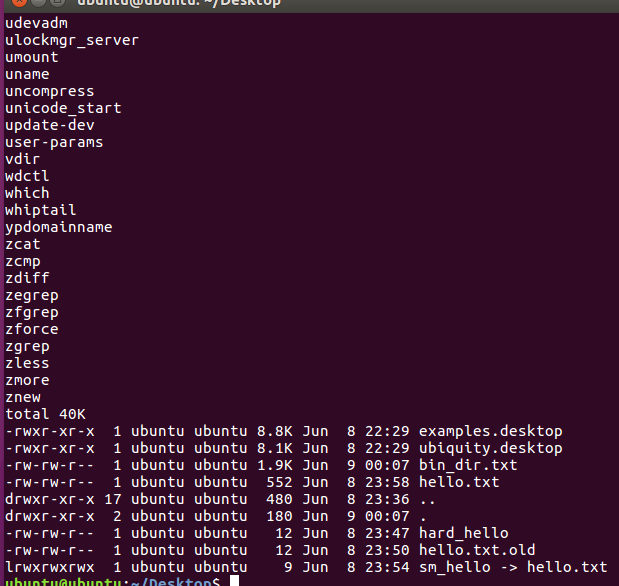
* “< file” – использовать файл как источник данных:



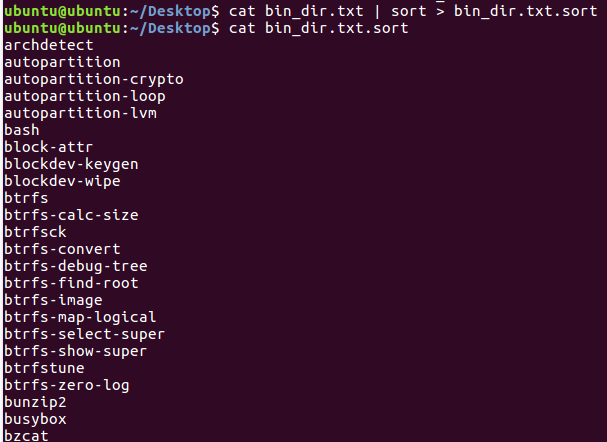


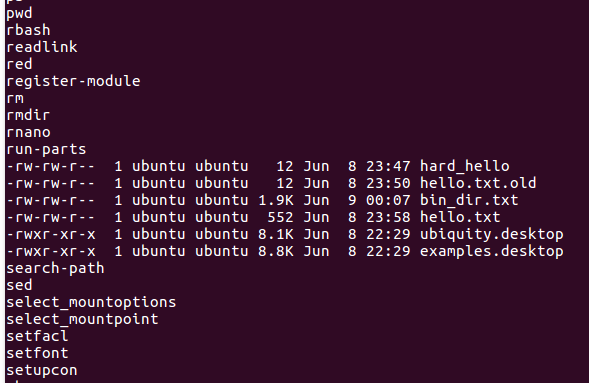
* “>> file” – добавляет ввод в конец файла, в отличие от первого варианта, данное перенаправление не перезаписывает содержимое.





* “ | ” – перенаправляет вывод одной команды в качестве ввода другой в определенное место, например в файл.





*Вывод:* в ходе выполнения данного пункта был получен опыт работы с каталогами и фалами, создание жестких и символических ссылок. Практически было применено перенаправление потоков ввода и вывода.

Управление доступом к файловой системе

*Цель:* Состав индексного дескриптора файла. Определение владельца-пользователя и владельца группы, команды chown, chgrp. Смысл режимов доступа r w x для файла и каталога. Дополнительные флаги SUID, SGID, StikyBit для файла и каталога.

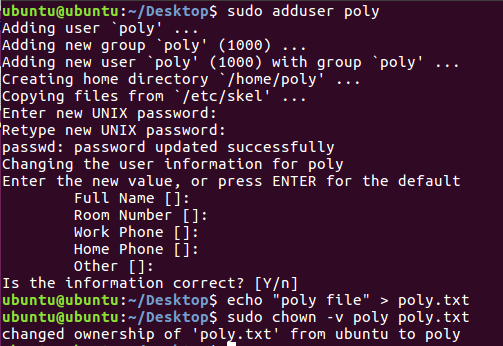
Выполнить управление правами доступа: chmod. Опробовать различные варианты и составить таблицу, сопоставляющую различные права доступа на каталог и файлы в нем и доступные или недоступные действия пользователя над файлами (в том числе варианты -w-, --x, показать действие права x на каталог). Найти, как назначаются права на symlink, сравнить с NTFS.

Состав индексного дескриптора файла:

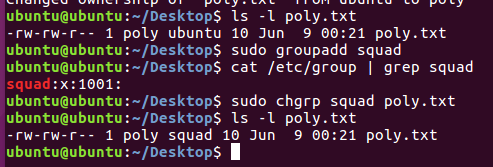
* di\_gid- идентификатор группы владельцев (GID)
* di\_mtime- время последней модификации файла
* di\_ctime- время создания файла
* di\_atime- время последнего обращения к файлу
* di\_size- размер файла в байтах
* di\_uid- идентификатор владельца файла (UID)
* di\_mode- тип и права доступа к данному файлу

Как и в любой ОС некоторые пользователи или группы пользователей имеют свои разрешения на файлы и каталоги. Например, обычным пользователям закрыт доступ к некоторым файлам и каталогам по решению владельца, который их создал или относящиеся к нему. С помощью команд “chown” и “chgrp” можно изменять владельца - пользователя и владельца -групп.

Команда chown – позволяет изменить владельца-пользователя. Создадим пользователя "poly", текстовый файл "poly.txt" у изначального пользователя "poly" и изменим владельца- пользователя для этого файла.



Команда chgrp – позволяет изменить владельца-группу для файла. Создадим группу "squad" с помощью команды "sudo groupadd squad" и изменим владельца-группу для того же файла:



Смысл режимов доступа r, w, x для файла и каталога заключается в безопасности. Как говорилось выше, для каждого пользователя, группы и остальным владелец файла или каталога хочет ограничить доступ к ним, выставляя соответствующие параметры. Различают 3 параметра доступа к файлам и каталогам:

* r (read) – доступ на чтение
* w (write) – доступ на запись
* x (execute) – доступ на исполнение

Дополнительные флаги SUID, SGID, StikyBit для файла и каталога существуют с целью упрощения выполнения различных программ обычным пользователям от имени суперпользователя:

* SUID – если этот флаг установлен, то при выполнении программы, id пользователя, от которого она запущена, заменяется на id владельца файла. Это позволяет обычным пользователям запускать программы от имени суперпользователя
* SGID – этот флаг работает аналогично, но различие заключается в том, что пользователь считается членом группы, с которой связан файл, а не групп, к которым он действительно принадлежит. Если SGID флаг установлен на каталог, все файлы, созданные в нем, будут связаны с группой каталога, а не пользователя
* StikyBit – если этот флаг установлен, то пользователи могут только создавать, читать и выполнять файлы, но не могут удалять файлы, принадлежащие другим пользователям

Команда chmod - предназначена для изменения прав доступа к файлам и каталогам.

Различают следующие виды прав:

* r – чтение
* w – запись
* x – выполнение
* s - выполнение от имени суперпользователя

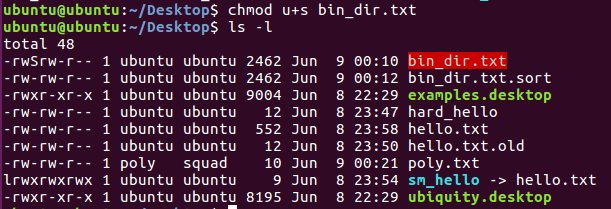
Также для следующих категорий пользователей можно установить эти права:

* u - владелец файла
* g - группа файла
* o - все остальные пользователи

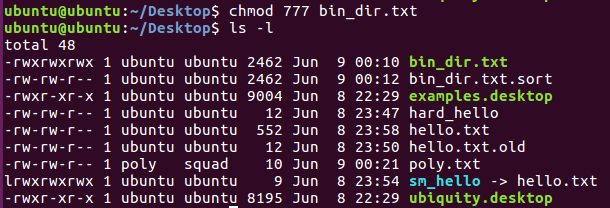
Права можно добавлять использую символ "+" и убавлять "-". Однако задать права можно не только символом, но и кодом, где каждая цифра дает разные права для пользователей:

* 0 - никаких прав
* 1 - только выполнение
* 2 - только запись
* 3 - выполнение и запись
* 4 - только чтение
* 5 - чтение и выполнение
* 6 - чтение и запись
* 7 - чтение запись и выполнение

Для примера поставить права суперпользователя на файл bin\_dir.txt:



А теперь выдадим все права на этот файл всем пользователям:



Полезная таблица для работы с утилитой chmod.



Права доступа на symlink никак не используются и не могут быть изменены. При попытке изменить права на symlink, права изменяются у самого объекта, на который она ссылается.

*Вывод*: в ходе выполнения были изучены команды для изменения владельца файла, владельца групп, рассмотрены флаги у файлов на права доступа, опробованы различные варианты прав на каталоги и файлы.

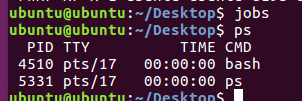
Управление заданиями

*Цель*: научиться запускать и управлять заданиями в оперативном и фоновом режимах. Команды jobs, ps, top, fg, bg, kill, stop, at, Ctrl+Z, nice. Интерпретировать информацию о состоянии и свойствах процессов.

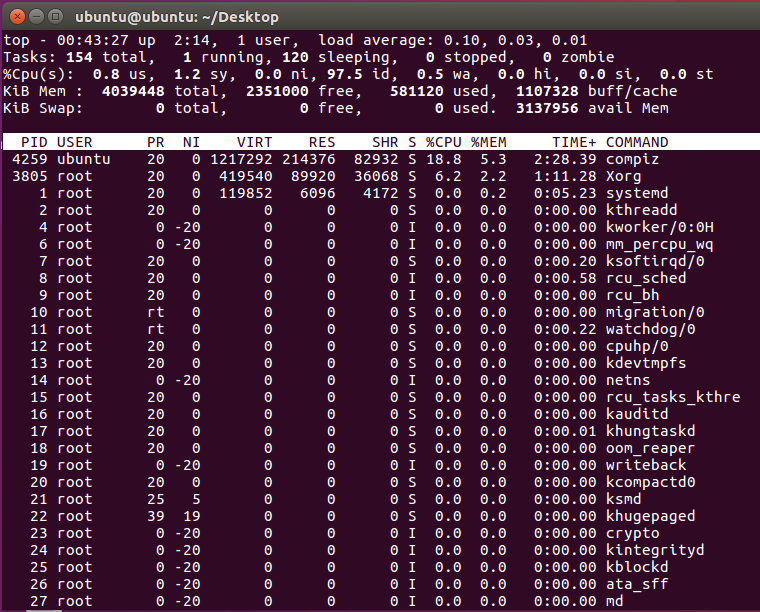
Задания могут выполняться прямо сейчас или в фоном режиме. В фоновом режиме задания выполняются и занимают какой-то объем памяти, но при необходимости мы можем их задействовать на передний план без потерь данных и ожиданий. На переднем плане может быть только одно задание. Поэтому задание на переднем плане считается заданием, с которым пользователь взаимодействует прямо сейчас.

Команда jobs – выводит список фоновых процессов, запущенных в данном терминале

Команда ps – выводит список всех запущенных процессов



Команда top – выводит информацию о всех процессах, запущенных в системе: идентификатор процесса, пользователь, запустивший процесс, использование процессом системных ресурсов, время работы процесса, команда, запустившая процесс.



Команды fg, bg – позволяют осуществлять переключение между фоновым и оперативным режимами выполнения программы прямо во время выполнения

Команда kill – позволяет принудительно завершить процесс

Команда at – позволяет запустить определенную задачу в указанное время

Сочетание клавиш Ctrl+Z – позволяет остановить выполнение процесса. Данное сочетание клавиш часто необходимо в случае запуска длительных по времени выполнения программ в оперативном режиме, так как в отличие от команды stop, это сочетание клавиш может прервать выполнение программы, не дожидаясь освобождения терминала

Команда nice - позволяет запустить процесс с заданным приоритетом. Приоритет задается в диапазоне от -20 (высокий) до 19 (низкий).

*Вывод*: был получен опыт работы с процессами в фоновом и оперативном режиме с помощью консольных команд, а также остановка этих команд с помощью команды kill.

Базовая файловая структура Linux

*Цель*: Структура и назначение основных каталогов файловой системы Linux. Получение информации о системе с помощью псевдофайловой системы /proc.

В дистрибутивах Linux каталоги могут отличаться, в ходе л.р. был использован дистрибутив Ubuntu 16.04.7.

/bin – cодержит файлы исполняемого типа. Программы, с которыми можно работать в однопользовательском режиме и прочие утилиты, которые применяются, пока не подключён каталог /usr/.

/sbin — системные исполняемые файлы. Во-первых, как и в случае с /bin, содержатся исполняемые файлы, доступные на ранних этапах загрузки. Во-вторых, тут располагаются программы, выполняемые с правами суперпользователя, те же утилиты обслуживания и пр.

/etc - в этой папке Linux расположены конфигурационные файлы всех программ, которые установлены в системе. Также есть скрипты для запуска демонов, автозагрузки программ и т. п. Главное, что следует запомнить, — речь идёт о файлах по настройке и конфигурации.

/dev — файлы устройств. В операционной системе Linux «всё есть файл». И различные внешние устройства тоже. В результате, любые подключённые микрофоны, клавиатуры, флешки и т. п. — это лишь файлы, которые находятся в каталоге /dev.

/proc — информация о процессах. Подсистема, которая динамически создаётся ядром. Содержит всю информацию о процессах, которые запущены в реальном времени. Тут можно найти данные об использовании системных ресурсов и получить другую информацию о системе.

/var — переменные файлы. Речь идёт о файлах, которые часто меняются — это базы данных, кеши, системные журналы и пр. Следует подробнее рассмотреть каталог Linux из папки /var/:

* /var/log — файлы логов;
* /var/lib — базы данных;
* /var/lock — файлы блокировок;
* /var/mail — почта;
* /var/spool — принтер, печать;
* /var/run — pid процессов.

/tmp – в данном каталоге располагаются временные файлы, которые созданы пользователями (users), программами или системой. Любой пользователь имеет права записи в данную директорию. При перезагрузке компьютера они удаляются.

/usr — программы пользователя. Этот каталог является самым большим и имеет много функций. Вкратце рассмотрим его структуру:

* /usr/bin/ — исполняемые файлы (графические редакторы, браузеры, плееры — не используются при загрузке системы);
* /usr/sbin/ — бинарные файлы программ, предназначенных для системного администрирования (выполняются с правами суперпользователя);
* /usr/lib/ — библиотеки для программ из /usr/sbin либо /usr/bin;
* /usr/local — файлы пользователя (программы, библиотеки и настройки, созданные пользователем).

/home — домашняя папка. Linux директория пользователей, где содержатся домашние каталоги всех пользователей. В данных каталогах пользователи хранят личные файлы, программные настройки и пр. Аналогия с Windows — папка на диске C. Но тут есть разница: home располагается на отдельном разделе, в результате чего при переустановке системы все данные и настройки сохраняются.

/boot — файлы загрузчика. Linux директория, где содержатся файлы, связанных с загрузчиком системы (ядро vmlinuz, образ initrd и пр.).

/lib — системные библиотеки. Тут располагаются файлы системных библиотек.

/opt — дополнительные программы. Эта папка Linux включает в себя установленные проприетарные программы, драйвера либо игры, то есть софт, созданный производителем в качестве отдельных исполняемых файлов.

/mnt — монтирование. Сюда сисадмины монтируют внешние либо дополнительные файловые системы Linux.

/media — съемные носители. Монтируются подключаемые внешние накопители — оптические диски, USB-устройства.

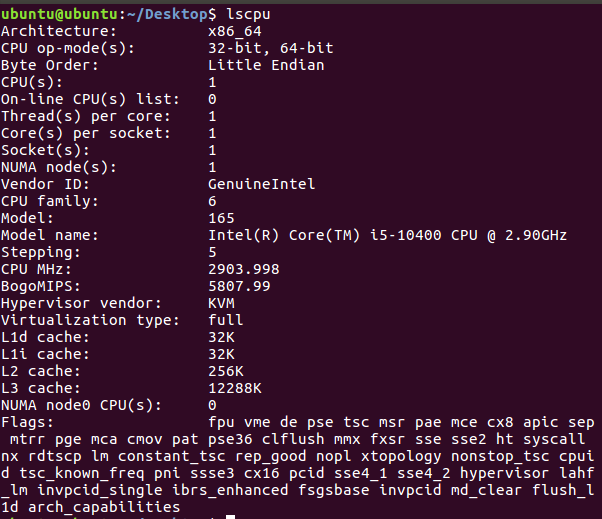
/srv — сервер. Файлы сервисов и серверов. Как пример — web-сервер Apache.

/run — процессы. Каталог включает в себя PID-файлы процессов, однако в отличие от /var/run, он располагается в TMPFS, в результате чего всё удаляется после перезагрузки.

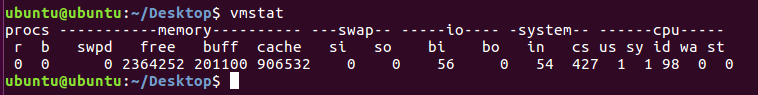
/sys — информация о системе. Каталоги Linux в этой папке предназначены для получения информации от ядра о системе. По сути, речь идёт о файловой системе, организованной ядром и позволяющей просматривать и менять множество параметров системной работы, ту же работу swap и много чего ещё.

Получение информации о системе:

Команда lscpu – получение информации о процессоре



Команда vmstat – информация о памяти, включая общее доступное количество и количество свободной памяти

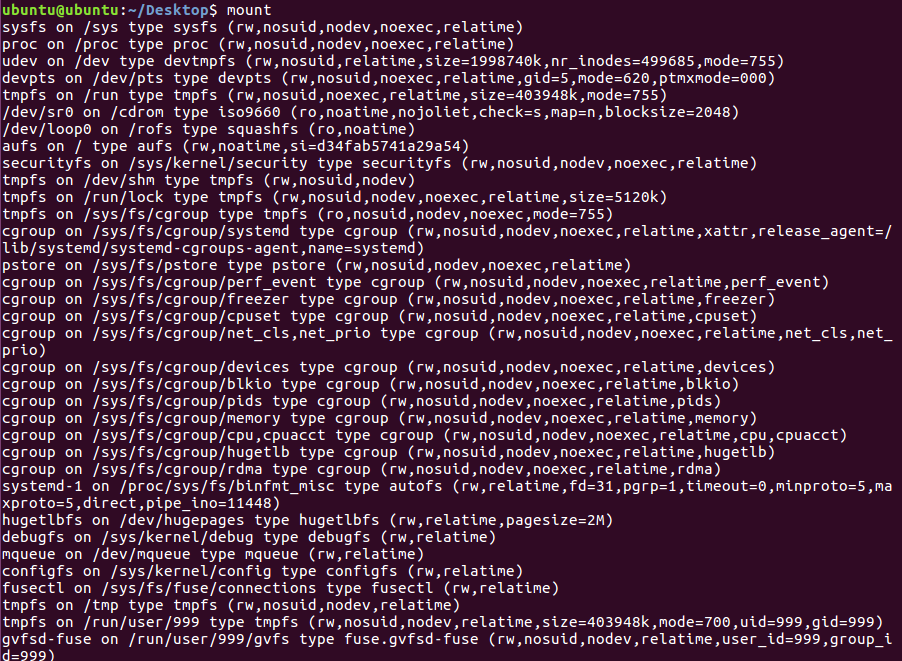


Здесь содержится информация о доступном объеме памяти, какой объем не используется, занятая память под разные режимы, общий объем области подкачки, зарезервированная память и многая другая информация.

Команда devices – выводит список устройств. Выводится большой список все устройств. Все эти устройства разделены на 2 блока: символьные устройства и блочные устройства.

Команда filesystens – список поддерживаемых файловых систем.

Команда mount – список смонтированных файловых систем



Команда partitions – информация о доступных разделах

Команда version – версия ядра

Команда swaps – показать файлы подкачки

Команда net/dev – показать сетевые интерфейсы

Команда interrupts – показать прерывания

Команда buddyinfo – содержится информация, которую можно использовать при диагностике проблем с фрагментацией памяти

Команда diskstats – статистика дискового ввода-вывода по каждому устройству

*Вывод*: в ходе выполнения были изучены структура и назначения корневых каталогов, получена основная информация о системе с помощью псевдофайловой системы /proc.

Вывод

В ходе выполнения лабораторной работы была произведена большая работа по изучению дистрибутива Linux. На практике был получен опыт по монтированию устройств, созданию пользователей, групп, настройке прав доступа к файлам и каталогам, работа с потоками ввода и вывода, работа с фоновыми процессами. Изучена структура и назначения каталогов.