МИНОБРНАУКИ РОССИИ

Изображение выглядит как текст

Автоматически созданное описаниеФедеральное 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»

ПРОВЕРИЛ:

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

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Перевалов А.Д.

19-В-1

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

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

Нижний Новгород 2022

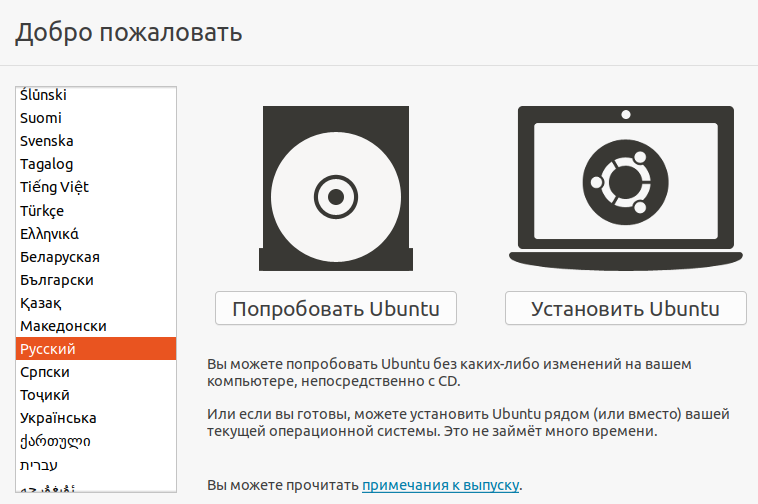
**1. Регистрация в системе**

Для выполнения лабораторной работы создадам виртуальную машину с ОС Linux (64-бит, Ubuntu 20.04.4 Desktop). Данная система уже с графической оболочкой, но также возможно работать через терминал, например, удаленно, используя лишь один терминал.

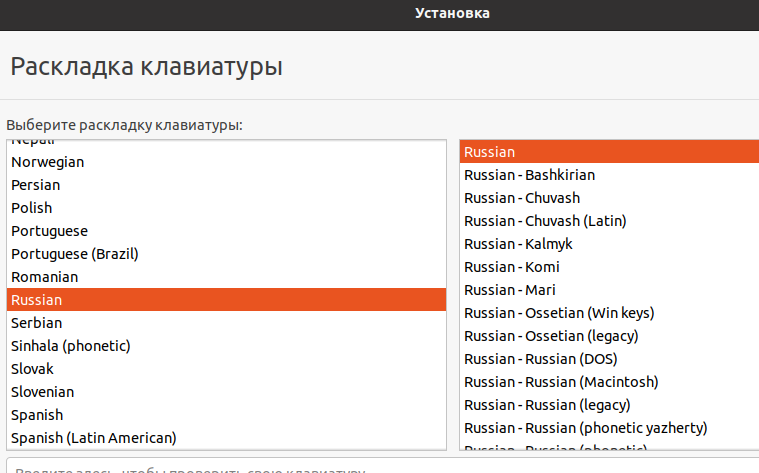
Проверяем диск:



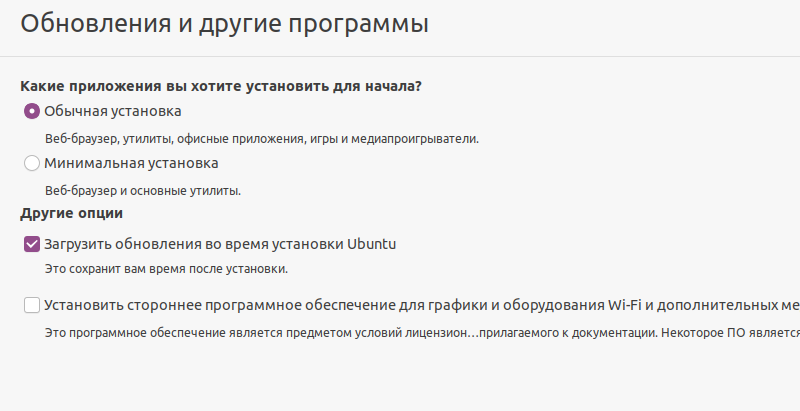
Выбираем язык и устанавливаем Ubuntu:



Выберем раскладку:



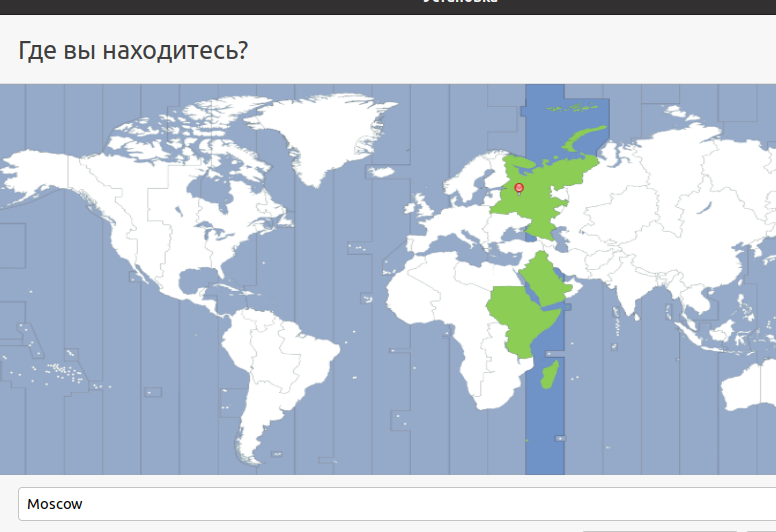
Выбираем вариант установки:



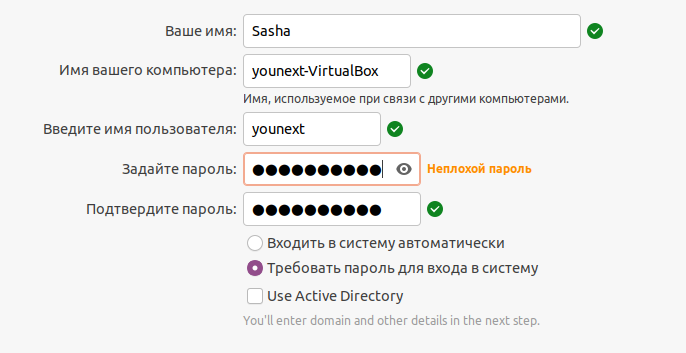
Выбираем тип установки:



Указываем наше положение для определения часового пояса:



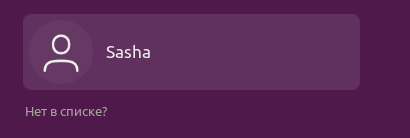
Далее указываем имя и задаем пароль:



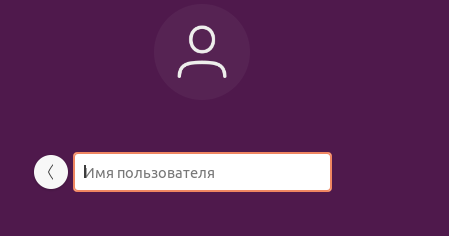
Пошла установка:



Далее перезагружаем и при запуске системы выбираем пользователя и вводим пароль:



Также в интерфейсе предоставляется возможность зарегистрироваться через кнопку «Нет в списке?»:

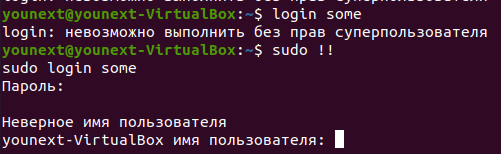


* **Login** – это команда приглашения входа в систему. В ответ на это приглашение нужно ввести имя пользователя (например, root). Зайдем в терминал. Как мы можем заметить, требуются права суперпользователя:

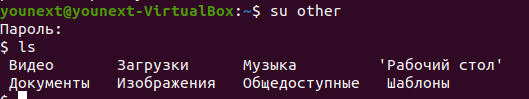


Команда sudo – необходима для работы от лица root. !! – показывает, что необходимо выполнить предыдущую команду от лица root.

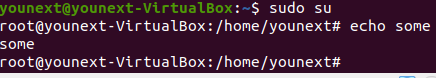
Если попытаться зайти на несуществующего пользователя, то система выдаст ошибку:



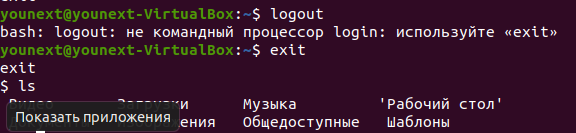
Также для того, чтобы зайти за другого пользователя есть команда su:



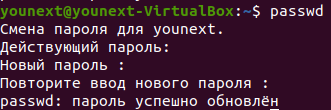
Команда sudo su – явно заходит под root, и все дальнейшие действия будут работать от его лица:



* **Exit** – позволяет выйти из учетной записи, под которой сидит основной пользователь, как и представлено выше. Также данная команда используется чтобы отключить подключение по ssh.



* **Logout** – команда выхода пользователя системы. Аналогична «Сменить пользователя» в Windows.
* **Passwd** - команда приглашения для ввода пароля пользователем. Стоит отметить ее некоторые особенности при работе в терминале – она появляется каждый раз, когда пользователь пытается внести изменения в систему (учетная запись администратора).

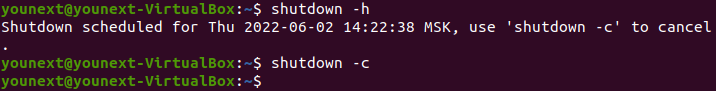


Запросим информацию о статусе, используя ключ –S:



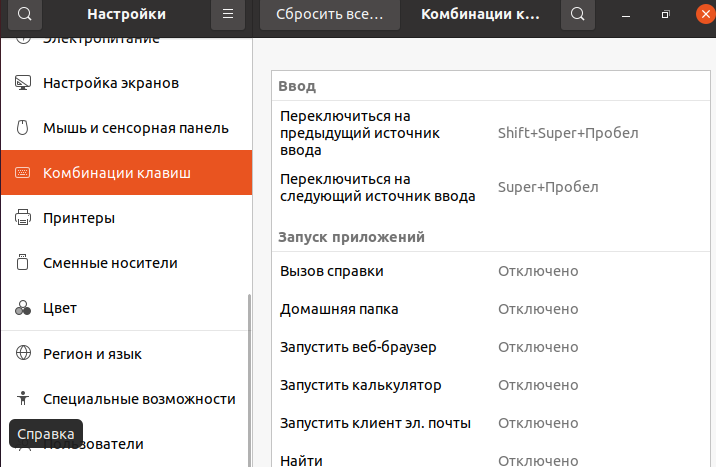
В выводе команды первое поле показывает имя пользователя, затем статус пароля (P = Password Set (пароль задан), LK = Password locked (пароль заблокирован), NP = No Password (нет пароля)). Третье поле отображает время последнего изменения пароля, а четвертое и последнее поле - минимальный и максимальный срок действия пароля, срок вывода предупреждения и срок дезактивации пароля.

* **Shutdown** – команда, вызывающая завершение работы через минуту после выполнения запроса. При использовании ключей может выполнить отложенную перезагрузку/завершение работы. Используем ключ -h — полная остановка системы (компьютер будет выключен). Для отмены используем ключ -с.



* **Reboot** - команда, позволяющая производить перезагрузку системы. С функциональной точки зрения является аналогом одного из вариантов команды shutdown.
* **halt** - команда, позволяющая произвести быструю и экстренную остановку системы. Остановка системы с выключением питания компьютера:
* Ctrl+Alt+Del – вызывает команду logout.
* Alt+Ctrl – вызов нового окна терминала
* Alt+Tab – переключение между окнами
* 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.

Стоит отметить, что в данной ОС очень гибкая система настройки горячих клавиш и комбинаций, а также языков и вообще настройки системы. Для этого достаточно вызвать «Параметры» - «Устройства» - «Комбинации клавиш».



Здесь пользователь может задать любые удобные для него комбинации клавиш. Вообще, стоит заметить, что Ubuntu, как и другие Ос семейства Linux, очень гибкие в настройке под конкретного пользователя и платформу.

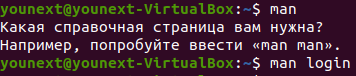
**Вывод:** при загрузке Linux Ubuntu система требует указать имя пользователя и назначить пароль. Существует множество команд для входа в систему и завершения работы.

**2. Основные информационные команды**

Рассмотрим основные информационные команды системы:

* **man**— команда Unix, предназначенная для форматирования и вывода справочных страниц.

Чтобы запустить руководство по какой-либо команде достаточно в командной строке набрать команду man, а затем через пробел название команды, по которой нужно руководство. После этого система запустить руководство по команде – конечно, если найдет.









Для удобства навигации все man-страницы соответствуют единому стандартному формату. Вот список некоторых разделов, которые часто используются на страницах:

1. NAME — имя команды и краткое однострочное описание ее назначения.

2. SYNOPSIS — список опций и аргументов командной строки, которые принимает команда, либо параметры функции и ее заголовочный файл.

3. DESCRIPTION — более подробное описание назначения и принципов работы команды или функции.

4. EXAMPLES — типовые примеры использования, обычно от самых простых к более сложным.

5. OPTIONS — описания для каждой из опций, которые принимает команда.

6. EXIT STATUS — коды возврата и их значения.

7. FILES — связанные с командой или функцией файлы.

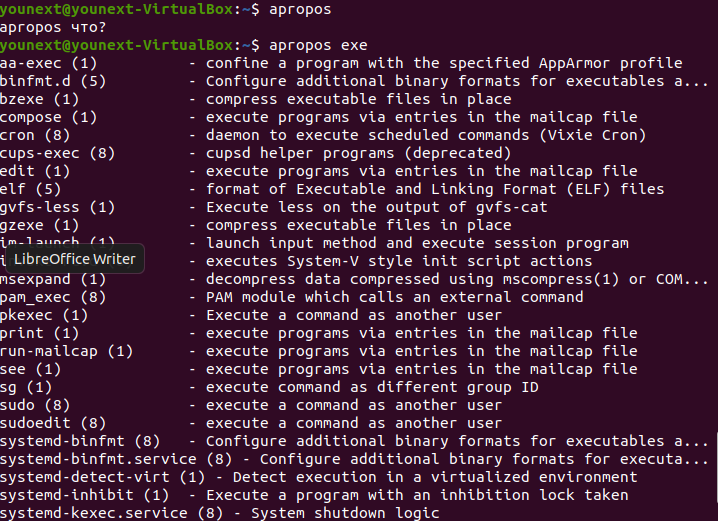
8. BUGS — вероятные проблемы, связанные с работой команды или функции и ожидающие решения. Также известны как KNOWN BUGS.

9. SEE ALSO — список связанных команд и функций.

10. AUTHOR, HISTORY, COPYRIGHT, LICENSE, WARRANTY — информация о программе: ее история, условия использования, создатели программы.

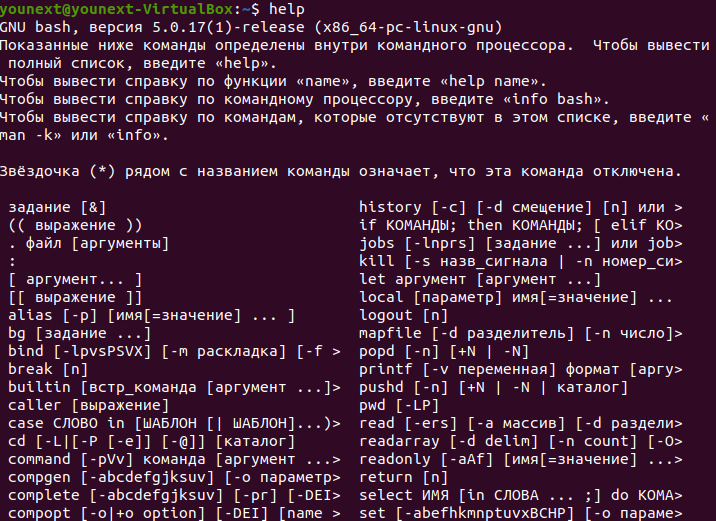
* **Apropos** – позволяет найти ключевое слово или регулярное выражение в именах и кратких описаниях справочных страниц. Используемая apropos поисковая база данных обновляется программой mandb. В зависимости от установки она может запускаться как периодически выполняемое задание cron или вручную, после установки новых справочных страниц.

Введем команду «apropos exe»: Мы получили на выход список файлов и их назначение.

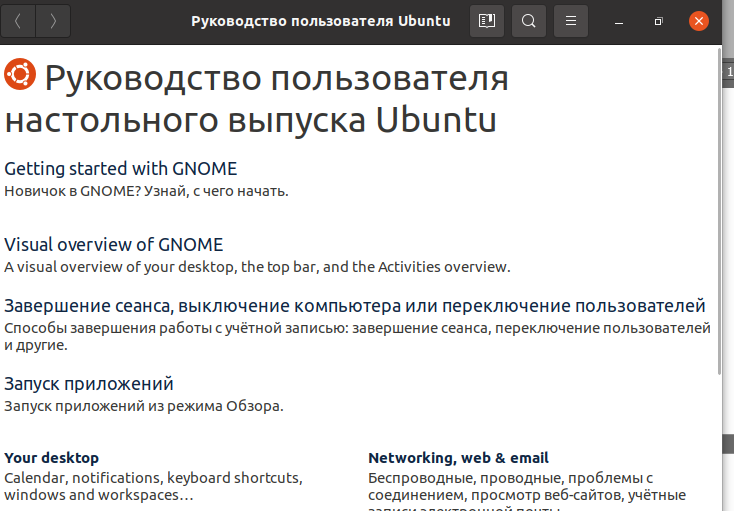


* **help** выводит информацию о встроенных командах. Отображает краткое описание встроенных команд. Если шаблон указан, дает подробную справку по всем командам, совпадающим с шаблоном в противном случае, выводит список разделов справки, печати.

Встроенный ключ --h или –help дает подробную справку о команде в терминале Bash Linux-систем.

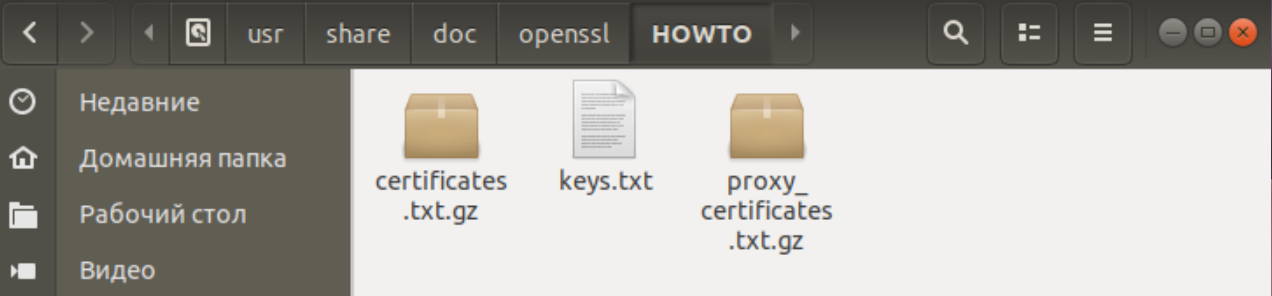


Также к справочной системе можно обратиться и через графический интерфейс. Поиск работает по такому же принципу, как и в Windows:



Значительное количество информации находится в уже установленной системе (при инсталляции по умолчанию) в виде так называемых страниц man, info и howto.

Страницы howto — это небольшие текстовые файлы, в которых описана последовательность действий для выполнения конкретной задачи (настройка чего-либо). В разных дистрибутивах howto располагаются в разных местах, в моем случае /usr/share/doc/openssl.у



Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Вывод: man — это пейджер справочных страниц системы. Каждый параметр страницы, переданный man, обычно является названием программы, утилиты или функции.

Команда apropos осуществляет поиск переданной пользователем строки в заголовках страниц руководств. Она является эквивалентом команды man –k.

Чтобы узнать какая последовательность действий выполняется для конкретной задачи, можно обратиться к файлу HOWTO.

**3. Управление учетными записями пользователей**

Попробуем получить информацию обо всех пользователях данной ОС с помощью командной строки, а также список всех зарегистрированных учетных записей можно получить в панели управления:

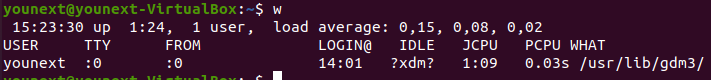
* Users



* Who – команда, позволяющая получить список терминалов и псевдотерминалов, используемых в данный момент в системе, и список пользователей, занимающих данные терминалы:



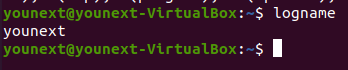
* W - команда, позволяющая получить более детальную информацию о пользователях и используемых ими терминалах. Помимо имен пользователей и названий терминалов выводит на экран текущий процесс, запущенный в терминале, время работы пользователя в терминале и т.д.



* id - команда, позволяющая вывести на экран информацию о текущем пользователе: идентификаторы пользователя и группы, а также некоторые права и настройки пользователя.



* logname, - команда, которая выводит имя пользователя, зарегистрированного в данный момент в системе.

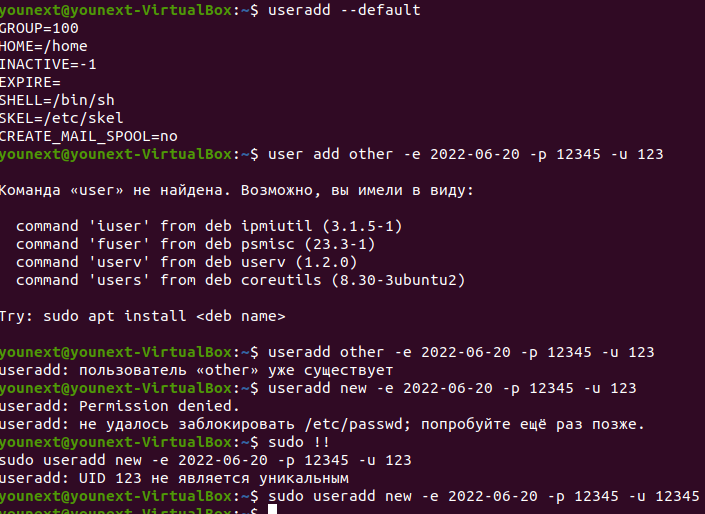


* finger - сетевой протокол, предназначенный для предоставления информации о пользователях удалённого компьютера.



* useradd– позволяет нам создать пользователя с различными настройками. Также есть еще одна команда adduser, которая также создает новых пользователей, но фактически adduser – это высокоуровневая настройка над useradd, и не во всех дистрибутивах Linux доступна, поэтому даже в man рекомендуют использовать useradd.

Создание нового пользователя:



Новый пользователь был создан с конкретным UID, на некоторый срок до 2 июня 2022 года, с паролем 12345.

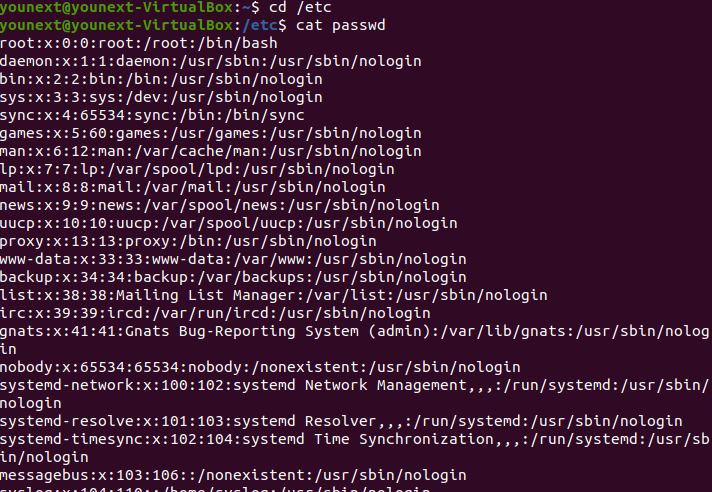
После создания можно удалить пользователя с помощью команды sudo userdel –remove:



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

* Имя пользователя (логин).
* Хеш пароля.
* Идентификатор пользователя (uid).
* Идентификатор группы (gid).
* Информационное поле GECOS (полное имя пользователя, номера телефонов и т.д.).
* Начальный (домашний) каталог.
* Регистрационная оболочка (shell).

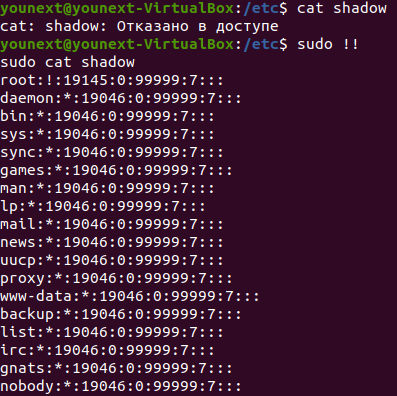
Как уже было указано выше, в ходе просмотра данного файла в нем были обнаружены записи о существующих пользователях, в том числе, о вновь созданном пользователе.



**/etc/shadow** — файл, который хранит хеши паролей всех пользователей в системе. Процессы суперпользователя могут читать его напрямую, для остальных создана специальная библиотека PAM. Она позволяет непривилегированным приложениям спрашивать у неё, правильный ли пароль ввёл пользователь, и получать ответ. Библиотека PAM как правило действует с привилегиями вызвавшего процесса.

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

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



Для каждой версии UNIX существуют записи файла паролей для нескольких псевдопользователей. Эти записи не подлежат редактированию. Такие пользователи без права входа в систему располагают соответствующими процессами для каждого аспекта, связанного с принадлежностью системы. Ниже приводится список наиболее распространенных псевдопользователей:

**Daemon** - используется серверными процессами системы

**bin** - владеет исполняемыми файлами пользователя

**sys** - владеет системными файлами

**adm** - владеет файлами бюджета

**uucp** - используется UUCP

**lp** - используется подсистемами 1р или **lpd**

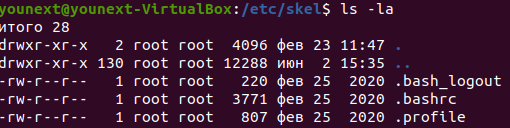
**nobody** - используется NFS

Существуют и другие стандартные псевдопользователи, такие как **audit, cron, mail, new и Usenet**. Все они обслуживают связанные с ними процессы и файлы.

/**etс/skel**/ - файл, который содержит в себе файлы настройки поведения командной оболочки **Bash**. По умолчанию включает в себя 3 файла:

* .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.



В этом разделе мы узнали, что с помощью команд who, w, id, logname, finger можно получить информацию о пользователях. Командой adduser можно создать нового пользователя. Файлы /etc/passwd и /etc/shadow являются первым и основным источником информации о правах пользователя операционной системы. С помощью /etc/skel можно определить состав и свойства домашнего каталога пользователя, профиля пользователя, профиля по умолчанию.

**4. Работа с устройствами и файловой системой**

**Udev** – это механизм, используемый для создания и именования узлов устройств /dev, соответствующих устройствам, присутствующим в системе. Udev использует информацию о соответствии, предоставленную sysfs, с правилами, предоставленными пользователем, для динамического добавления необходимых узлов устройства.

Существуют следующие виды файлов:

1. Обычные файлы (сюда относятся все файлы с данными, играющими роль ценной информации сами по себе);
2. Каталоги — это файлы, в качестве данных которых выступают списки других файлов и каталогов. Именно в данных каталогах осуществляется связь имени файла (словесного обозначения для людей) с его индексным дескриптором (истинным именем-числом). Отсюда следует, что 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 (механизм жестких ссылок). Файлы всегда содержаться в каталогах, иначе просто недоступны.
3. Специальные файлы:

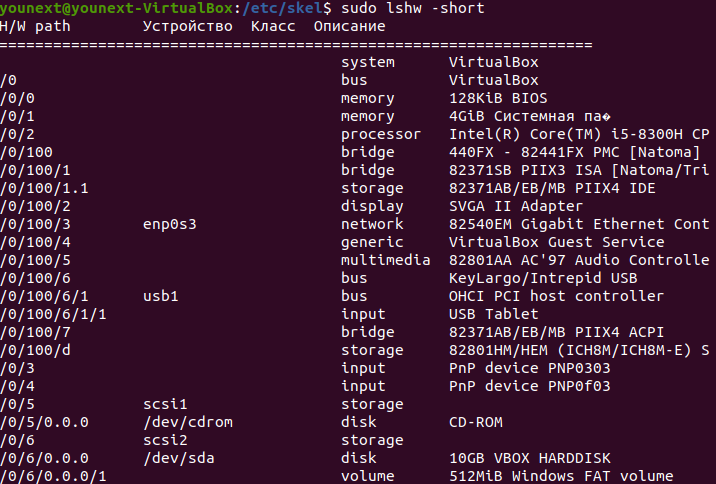
Механизм использования ввода-вывода. Большинство специальных файлов находятся в /dev. Они используются для обозначения физических устройств компьютера. Например, файл /dev/console - соответствующий консольному терминалу.  
Все, что выводится в файл /dev/console, будет просто появляться на экране терминала. При попытке чтения из файла /dev/console, вы будете получать то, что вводится с клавиатуры.

Специальные файлы делятся на два типа:

* 1. Блочные - запись и чтение допускаются только блоками. Например, разделы дисков с файловой системой являются таковыми.
  2. Символьные - можно читать и писать отдельными байтами. К ним относятся терминалы, неформатированные диски и разделы дисков без файловой системы.

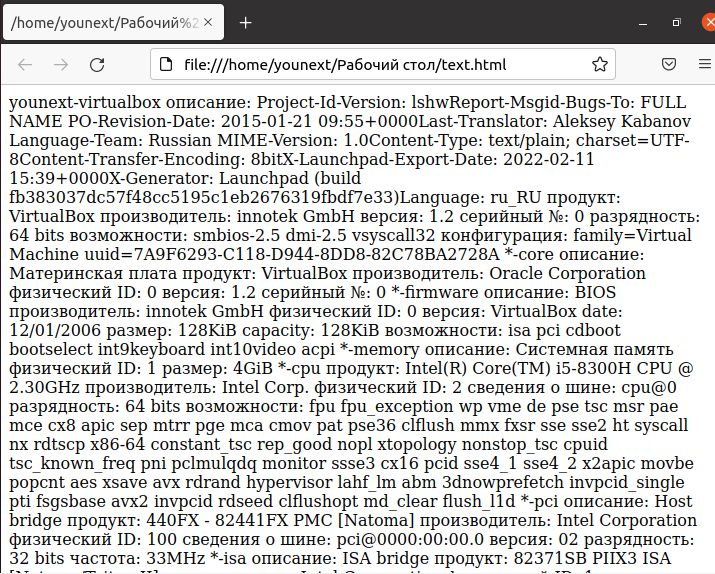
Понятия "размер" у специального файла не существует. Ведь этот файл обозначает физическое устройство. Фактически, это ссылка на соответствующий драйвер. Вместо длины, команда ls показывает для таких файлов два числа: "мажорный" и "минорный" номера устройств. Будем считать, что "мажор" — это порядковый номер драйвера устройства, а "минор" - внутренний номер устройства в таблице обслуживающего его драйвера.

Выведем краткую информацию обо всех устройствах ПК:



Для сохранения журнала работы утилиты в файле достаточно использовать параметр форматирования вывода и перенаправить вывод в файл:

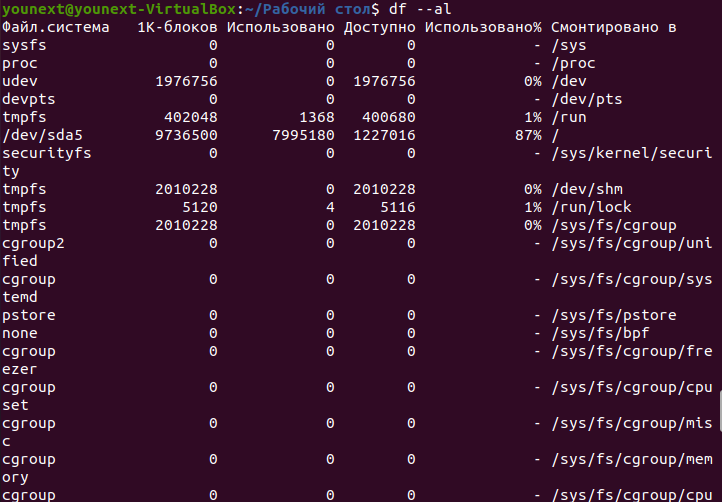




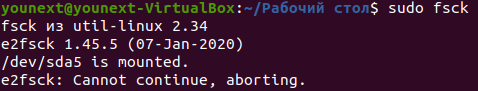
Теперь рассмотрим некоторые команды работы с дисками:

* Mount - утилита командной строки в UNIX-подобных операционных системах. Применяется для монтирования файловых систем.
* Umount - утилита командной строки в UNIX-подобных системах. Применяется для размонтирования файловых систем, предварительно смонтированных посредством вызова утилиты mount. Для её использования требуются привилегии суперпользователя.
* Df - утилита в UNIX и UNIX-подобных системах, показывает список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования.

Список файловых систем:



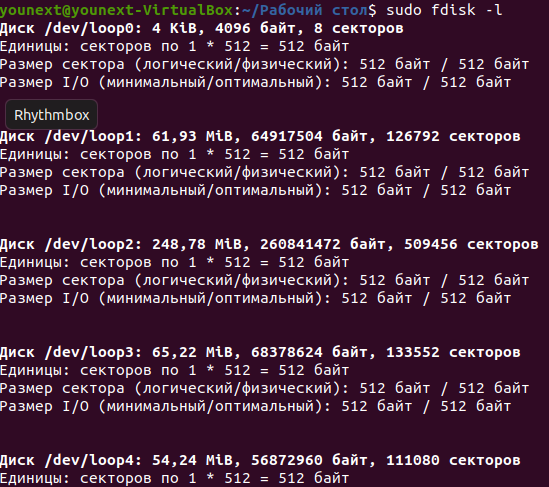
* Mkfs - это команда, используемая для форматирования блочного устройства хранения с определенной файловой системой. Эта команда является частью UNIX и UNIX-подобных операционных систем. В UNIX блочное запоминающее устройство должно быть отформатировано файловой системой до того, как оно будет смонтировано и доступно через иерархию файловой системы операционной системы.
* Fsck - используется для проверки и, в случае необходимости, исправления ошибок одной или нескольких файловых систем. filesys может быть именем блочного устройства (например, /dev/sda7), точкой монтирования (/, /mnt/sda7 и т. д.) этого устройства, меткой раздела или UUID-индексом. Обычно fsck параллельно проверяет данные на разных физических дисках, чтобы сократить общее время, необходимое для полной проверки всех дисков. Если файловых систем не указано и не указана опция -A, то fsck по порядку проверит файловые системы, указанные в /etc/fstab. Это эквивалентно опции -As.

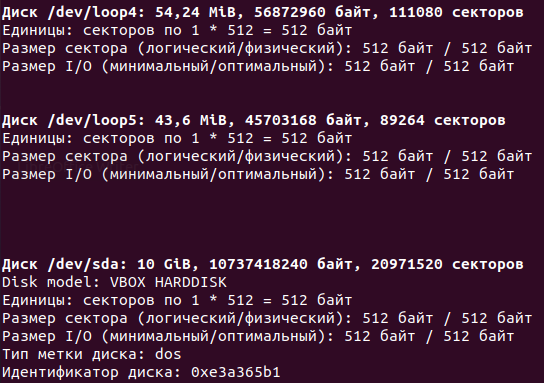


* Fdisk - общее название системных утилит для управления разделами жёсткого диска. Широко распространены и имеются практически в любой операционной системе, но работают по-разному. Используют текстовый интерфейс пользователя. В системе Linux можно запускать fdisk в интерактивном или командном режиме. При запуске нужно обязательно указывать устройство диска, например: /dev/hdb, /dev/sda и т. п. В отдельных случаях (при изменении раздела в командном режиме), потребуется указать раздел диска (например, /dev/hda1).

Файлы флешек и других устройств хранения данных находятся в каталоге /dev. Устройства нумеруются по алфавиту sda, sdb, sdc, sdd и т д. Так как флешка подключается последней у нее будет старшая буква. Например, sdb или sdc. Имя sda принадлежит жесткому диску. Можно, посмотреть содержимое каталога /dev/ командой ls, но такой способ медленный и не наглядный.

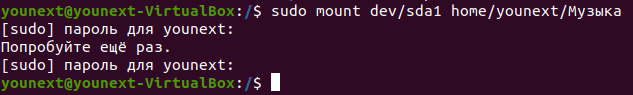
Воспользуемся утилитой fdisk:





Как видно, здесь можно посмотреть детальную информацию об устройстве: имя файла, список разделов, формат таблицы разделов, список разделов. А главное - размер раздела и его файловую систему.

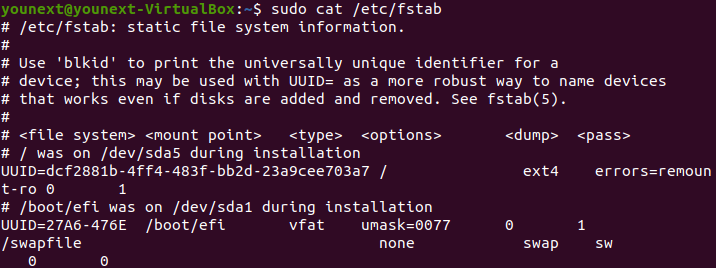
Создадим папку в домашнем каталоге. И используя команду mount, смонтируем туда файловую систему:



Теперь демонтируем:



Рассмотрим конкретный fstab:



В каждой строке имеются следующие поля:

1) ЧТО монтируем – блочные устройства, которое примонтировано. Это может быть файл-образ, раздел диска или диск целиком. С файлами-образами всё вполне очевидно — нужно указать путь к файлу. Диски и их разделы числятся в системе как, например, /dev/sda (диск целиком) или /dev/hdc1 (раздел диска), но монтировать разделы встроенного диска стоит по UUID. Это позволит правильно монтировать разделы даже если они поменяют свой порядок, например, при добавлении новых разделов или подключении новых дисков.

2) КУДА монтируем – точка монтирования – путь в корневой ФС к каталогу, в который будет смонтировано устройство. Крайне желательно использовать пустой каталог.

3) Тип файловой системы. В моем примере ФС с типом ext4.

4) Опции монтированной ФС. Выбраны по дефолты, но с доп.опцией error.

По умолчанию:

А) rw (чтение-запись);

Б) suid (соблюдать биты setuid и setgid);

В) dev (интерпретировать символы и блокировать устройства в файловой системе);

Г) exec (разрешить выполнение двоичных файлов и скриптов);

Д) auto (монтировать файловую систему, когда используется опция -a команды mount);

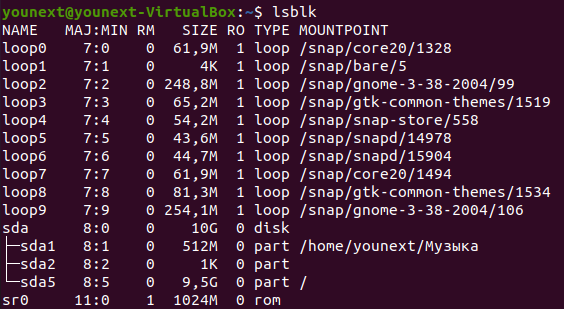
Е) nouser (сделать файловую систему не монтируемой обычным пользователем);

Ж) async (выполнять операции ввода-вывода в файловой системе асинхронно).

5) Индикатор необходимости делать резервную копию. У меня выставлен 0.

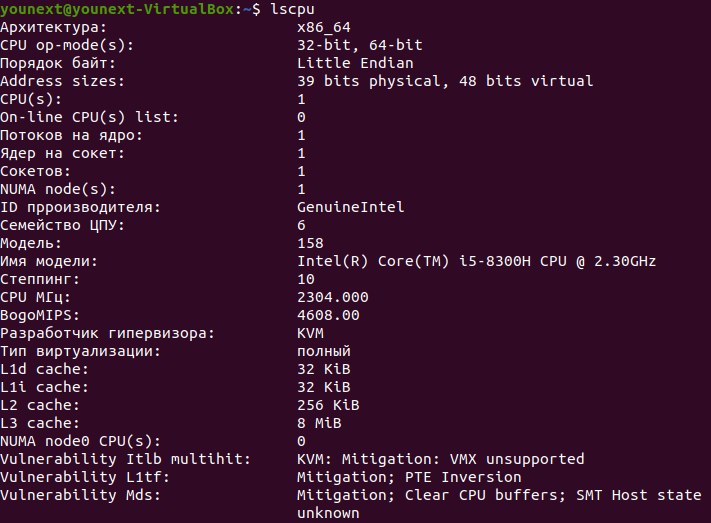
6) Порядок проверки раздела. У меня выстелен 1 – так как это основная файловая система. Только у нее может быть 1.

**Команда lsblk** – выводит на экран информацию обо всех блочных устройствах, таких как жесткие диски, приводы DVD и прочее:



У нас на виртуальной машине подключен 1 диск: «физический диск» с размером в 10 Гб. Находится файл устройства /dev/sda1.

**Команда lscpu** – выводит информацию о CPU:



**Команда lsusb** – выводит информацию обо всех устройствах, присоединённых через USB-интерфейс:



**Команда lspci** – тоже, что и выше, но через интерфейс pci:

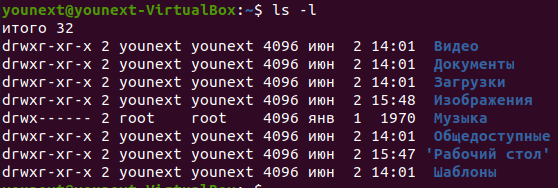


После изучения данного раздела можем сказать, что совершить монтирование диска в Linux можно с помощью команды «$ mount файл\_устройства папка\_назначения», демонтировать - «$ umount файл\_устройства папка\_назначения». Также можно с помощью команды fdisk совершить разметку диска в терминале. В «/dev» находится большинство специальных файлов. Они используются для обозначения физических устройств компьютера.

**5. Управление файлами**

* Ls - утилита Unix, которая печатает в стандартный вывод содержимое каталогов. Команда ls сначала выводит список всех файлов (не каталогов), перечисленных в командной строке, а затем выводит список всех файлов, находящихся в каталогах, перечисленных в командной строке. Если не указано ни одного файла, то по умолчанию аргументом назначается текущий каталог. Опция -d заставляет ls не считать аргументы-каталоги каталогами. Будут отображаться только файлы, которые не начинаются с . или все файлы, если задана опция -a.

Для задания одноколоночного или многоколоночного режима вывода могут использоваться опции -1 и -C, соответственно.





* Cd - команда командной строки для изменения текущего рабочего каталога в Unix, DOS и других операционных системах. Она также доступна для использования в скриптах командного интерпретатора или в пакетных файлах.



* Pwd — консольная утилита в UNIX-подобных системах, которая выводит полный путь от корневого каталога к текущему рабочему каталогу: в контексте которого (по умолчанию) будут исполняться вводимые команды.



* Cp - команда Unix в составе GNU Coreutils, предназначенная для копирования файлов из одного в другие каталоги (возможно, с другой файловой системой). Исходный файл остаётся неизменным, имя созданного файла может быть таким же, как у исходного, или изменится.



* Команда «&&» - позволяет создать конвейер команд, то есть сначала выполняется первая команда, потом, если команда была выполнена успешно, то выполняется вторая команда. Если же первая команда была выполнена не успешно (тут под понятием «команда» может быть и целая программа), то выполнение прерывается



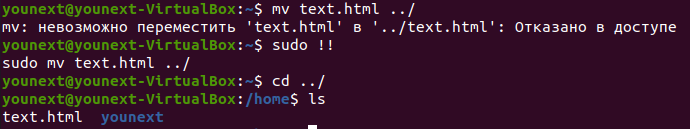
* Команда echo – печатает на экране, то, что будет написано после команды



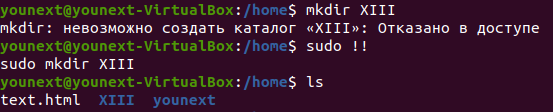
* Mv - утилита в UNIX и UNIX-подобных системах, используется для перемещения или переименования файлов.

Если в качестве аргументов заданы имена двух файлов, то имя первого файла будет изменено на имя второго.

Если последний аргумент является именем существующего каталога, то mv перемещает все заданные файлы в этот каталог (mv file ./dir/ )



* Rm - утилита в UNIX и UNIX-подобных системах, используемая для удаления файлов из файловой системы
* Mkdir - команда для создания новых каталогов.



* Rmdir - rm удаляет каждый заданный файл. По умолчанию каталоги не удаляются, но если заданы опции -r или -R, то будет удаляться все дерево каталогов ниже заданного каталога, включая и его.



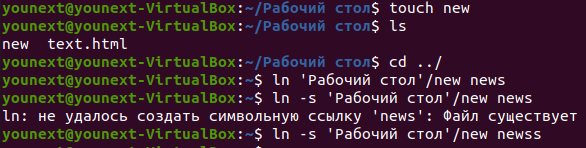
* Ln - команда UNIX, устанавливающая связь между файлом и именем файла.

ln файл имя\_ссылки

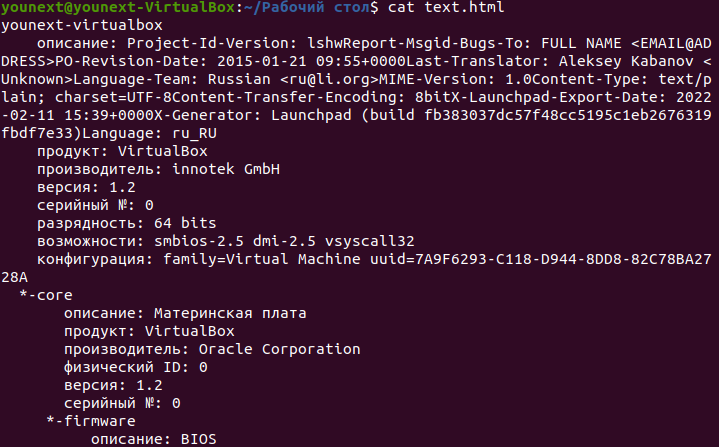
# создаётся «жёсткая» ссылка (hard link)

ln -s файл имя\_ссылки

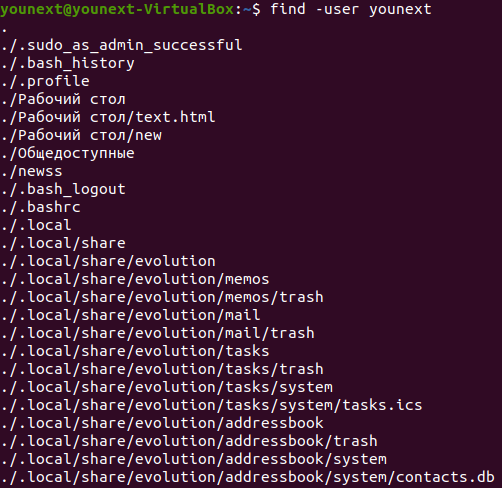
# создаётся «символьная» ссылка (symbolic link)



* **cat** — одна из наиболее часто используемых команд в Linux. Она считывает данные из файлов и выводит их содержимое.



* **Tee** обычно используется для разделения вывода программы, так что он может быть отображен и сохранен в файле. Команда может использоваться для захвата промежуточного вывода перед изменением данных другой командой или программой. Команда Tee считывает стандартный ввод, затем записывает его содержимое в стандартный вывод.
* Утилита **more** предназначена для постраничного просмотра файлов в терминале Linux. Своим названием она обязана надписи more (в русскоязычном варианте — дальше), появляющейся внизу каждой страницы.
* **Find** — утилита поиска файлов по имени и другим свойствам, используемая в UNIX‐подобных операционных системах. Может производить поиск в одном или нескольких каталогах с использованием критериев, заданных пользователем.



* **Grep** - утилита командной строки, которая находит на вводе строки, отвечающие заданному регулярному выражению, и выводит их, если вывод не отменён специальным ключом



* Перенаправление потоков - возможность командной оболочки (например, в UNIX) перенаправлять стандартные потоки в определенное пользователем место, например, в файл.

Команда ls позволяет получить информацию о каталогах и файлах; cd, pwd, cp, mv, rm, mkdir, rmdir; cat, tee, more, file - команды вывода; команда поиска find, команда фильтра grep.

# **6. Управление доступом к файловой системе.**

Каждому файлу на диске соответствует один и только один индексный дескриптор файла, который идентифицируется своим порядковым номером - индексом файла. Это означает, что число файлов, которые могут быть созданы в файловой системе, ограничено числом индексных дескрипторов, которое либо явно задается при создании файловой системы, либо вычисляется исходя из физического объема дискового раздела. Индексный дескриптор файла имеет следующее строение:

|  |  |
| --- | --- |
| Название поля | Описание |
| di\_mode | Тип и права доступа к данному файлу |
| di\_uid | Идентификатор владельца файла (Owner UID) |
| di\_size | Размер файла в байтах |
| di\_atime | Время последнего обращения к файлу (Access time) |
| di\_ctime | Время создания файла |
| di\_mtime | Время последней модификации файла |
| di\_gid | Идентификатор группы (GID) |

Как видно из таблицы, индексный дескриптор каждого файла содержит два важных поля: идентификатор владельца файла (di\_uid) и идентификатор группы владельцев файлов (di\_gid). Каждый пользователь или группа пользователей имеют определенные права по отношению к файлу или каталогу, владельцы файла, как правило, имеют более широкие права. Владельцами файла обычно становятся пользователи, которые создают данный файл.

Существуют специальные команды, позволяющие явно определить владельца-пользователя или владельца-группу для файла:

* chown - команда, позволяющая изменить владельца-пользователя для файла:
  + sudo chown -v user file.txt - смена владельца-пользователя для файла file.txt на пользователя user с выводом диагностического сообщения, которое подтвердило смену пользователя и указало имя старого пользователя.
  + При повторном выполнении данной команды было выведено диагностическое сообщение о том, что данный пользователь уже является владельцем файла.
  + 
* chgrp - команда, позволяющая изменить владельца-группу для файла:
  + sudo chgrp -R group ex - смена группы-владельца на группу group для директории ex с рекурсивной сменой группы-владельца для всего содержимого директории ex.

В операционных системах Linux используются 3 базовых права доступа – на чтение (read), запись (write) и исполнение (execute). Соответственно, права назначаются пользователю (user), группе (group) и всем остальным (world).

Для того, чтобы позволить обычным пользователям выполнять программы от имени суперпользователя без знания его пароля была придумана такая вещь, как SUID и SGID биты. Рассмотрим эти полномочия:

* **SUID** - если этот бит установлен, то при выполнении программы, id пользователя, от которого она запущена заменяется на id владельца файла. Фактически, это позволяет обычным пользователям запускать программы от имени суперпользователя;
* **SGID** - этот флаг работает аналогичным образом, только разница 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. Такое поведение используется для организации общих папок;
* **Sticky-bit** - этот бит тоже используется для создания общих папок. Если он установлен, то пользователи могут только создавать, читать и выполнять файлы, но не могут удалять файлы, принадлежащие другим пользователям.

Для управления правами доступа к файлам и папкам используется команда chmod, которая позволяет задать права доступа к тому или иному файлу или папке для пользователя-владельца, группы-владельца и остальных пользователей. В качестве аргумента данная команда может принимать список прав доступа в символьном виде или в виде числовых эквивалентов:

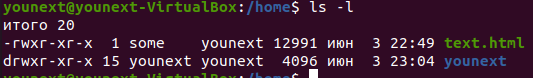
* sudo chmod -v 755 1.txt - данная команда устанавливает для пользователя-владельца права 7 (rwx), для группы-владельца - 5(r-х), для остальных пользователей - 5(r-х).



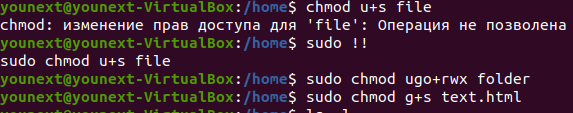
Рассмотрим подробнее, что значат условные значения флагов прав:

* **---** - 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;
* **--s** - установлен SUID или SGID бит, первый отображается в поле для владельца, второй для группы;
* **--t** - установлен sticky-bit, а значит пользователи не могут удалить этот файл.

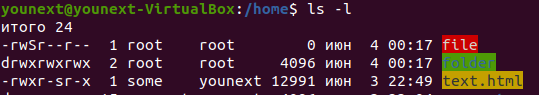
Чтобы узнать права на файл linux выполним такую команду, в папке, где находится этот файл:



Назначим всем пользователям полный доступ к folder, для файла file установим SUID, а для 1.html - SGID:



Теперь видим список каталогов и файлов с видом разрешения:



Изменение прав доступа к самой символической ссылке не имеет никакого смысла.

можно считать, что их просто нет (вариант: они одинаковы для любого процесса).

манипуляции с правами доступа к символической ссылке «прозрачно транслируются» на объект файловой системы, на который она ссылается.

Можем сделать вывод, что дополнительные флаги SUID, SGID, StikyBit для файла и каталога позволяют изменить права. С помощью команды «ls -l» можно увидеть права на файл или каталог, с использованием флагов прав. Также были изучены состав и свойства индексного дескриптора файла, команды для изменения владельцев файлов, варианты доступа к файлам и папкам, была применена команда назначения доступа.

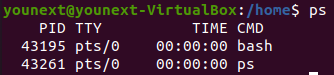
**7. Управление заданиями**

Всякая выполняющаяся в Linux программа называется процессом. Linux как многозадачная система характеризуется тем, что одновременно может выполняться множество процессов, принадлежащих одному или нескольким пользователям.

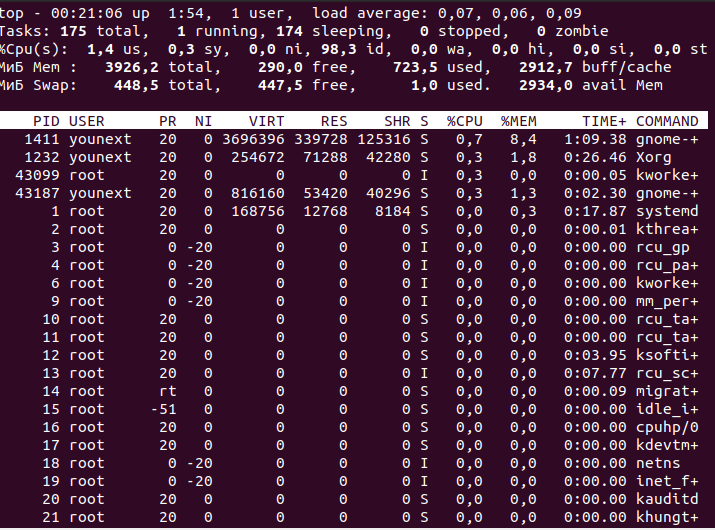
Задания могут быть либо на переднем плане (foreground), либо фоновыми (background). На переднем плане в любой момент времени может быть только одно задание. Задание на переднем плане — это то задание, с которым вы взаимодействуете; оно получает ввод с клавиатуры и посылает вывод на экран (если, разумеется, вы не перенаправили ввод или вывод куда-либо ещё).

Между заданиями в фоновом режиме и приостановленными заданиями есть большая разница. Приостановленное задание не работает — на него не тратятся вычислительные мощности процессора. Это задание не выполняет никаких действий. Приостановленное задание занимает некоторый объем оперативной памяти компьютера, через некоторое время ядро откачает эту часть памяти на жёсткий диск «до востребования». Напротив, задание в фоновом режиме выполняется, использует память и совершает некоторые действия, которые, возможно, вам требуются, но вы в это время можете работать с другими программами.

* jobs - команда, которая выводит список фоновых процессов, запущенных в данном терминале
* ps - команда, позволяющая вывести список всех запущенных процессов:

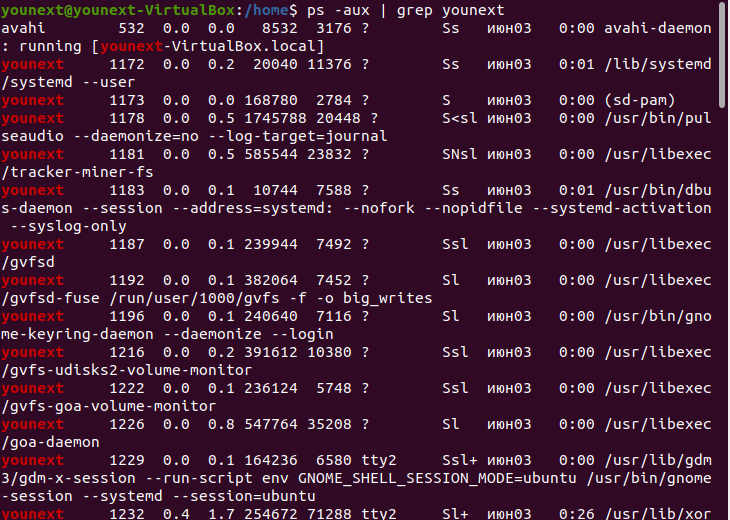


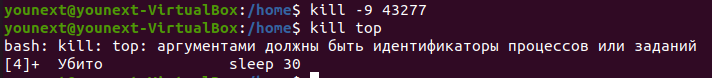
* Ctrl+Z - сочетание клавиш, позволяющее остановить выполнение процесса. Данное сочетание клавиш часто необходимо в случае запуска длительных по времени выполнения программ в оперативном режиме, так как в отличие от команды stop, это сочетание клавиш может прервать выполнение программы, не дожидаясь освобождения терминала.
* nice - команда позволяющая запустить процесс с заданным приоритетом. Приоритет задается целыми числами в диапазоне: -20 (высокий) - 19 (низкий)
* stop, - команда, позволяющая приостановить работу процесса
* at, - команда, позволяющая запустить определенную задачу в указанное время
* top - команда, выводящая на экран информацию о всех процессах, запущенных в системе: идентификатор процесса, пользователь, запустивший процесс, использование процессом системных ресурсов, время работы процесса, команда, запустившая процесс



* fg,bg - команды, позволяющие осуществлять переключение между фоновым и оперативным режимами выполнения программы прямо во время выполнения
* kill - команда, позволяющая принудительно завершить выполнение процесса:

Создадим процесс sleep 30 (запуск длительного процесса). Воспользуемся утилитой kill:





**8. Базовая файловая структура Linux**

В разных дистрибутивах Linux каталоги тоже могут быть разные: в некоторых будут дополнительные файлы/каталоги конфигурации, а некоторые файлы конфигурации, вполне возможно, будут называться как-то по-другому. Приведем названия каталогов, которые входят в дистрибутив Fedora.

Основные каталоги корневой файловой системы:

/ — корневой каталог

/bin — содержит стандартные утилиты Linux, основные исполняемые файлы, доступные всем пользователям, а также содержит символьные ссылки на исполняемые файлы.

/boot — содержит конфигурационные файлы загрузчика в каталоге grub, образы ядра, файлы Initrd.

/dev — содержит файлы устройств. Большинство устройств в Linux представляют из себя файлы в особой файловой системе. И вот эти файлы хранятся в каталоге /dev, куда к ним обращается система для выполнения задач, связанных с вводом/выводом.

/etc — содержит конфигурационные файлы операционной системы и всех сетевых служб. В Linux общесистемные настройки хранятся в разных конфигурационных файлах, которые можно редактировать обычным текстовым редактором.

/home — домашний каталог пользователей. Домашний каталог пользователя обозначается /home/Имя\_Пользователя.

/lib — здесь находятся различные библиотеки и модули ядра. В процессе установки различных программ устанавливаются зависимости для корректной работы программы. Вот эти зависимости — набор собранных особым образом файлов, которые подключаются во время установки к устанавливаемой программе.

/mnt и /media — обычно в этих каталогах содержатся точки монтирования. В современных дистрибутивах Linux этот процесс обычно происходит автоматически. При этом в каталогах /mnt или /media создается подкаталог, имя которого совпадает с именем монтируемого тома.

/opt — здесь обычно размещаются установленные программы, имеющие большой дисковый объем, или вспомогательные пакеты.

/proc — это каталог псевдофайловой системы procfs, которая используется для предоставления информации о процессах. В системе Linux присутствует виртуальный файловый объект, именуемый каталогом /proc. Он существует только во время работы системы в оперативной памяти компьютера. Каталог представляет интерес и с точки зрения безопасности. Многие из утилит, выводящие информацию о системе (например, команда ps), берут свои исходные данные именно из этого каталога.

/root — каталог пользователя root. Для доступа нужно предварительно пройти аутентификацию.

/run — это совершенно новый каталог для хранения данных, которые были запущены приложениями, требующимися в процессе работы.

/sbin — набор утилит для системного администрирования, содержит исполняемые файлы, необходимые для загрузки системы и ее восстановления. Запускать эти утилиты имеет право только root.

/tmp — каталог, в котором хранятся временные файлы. Linux, в отличие от Windows, следит за чистотой и регулярно очищает этот каталог.

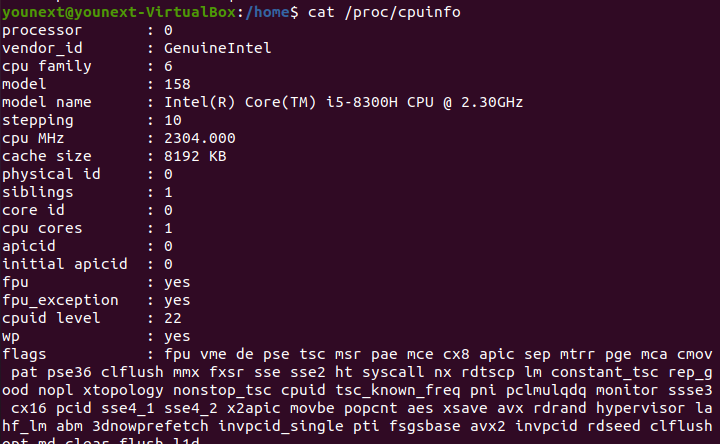
/usr — содержит пользовательские программы, документацию, исходные коды программ и ядра. По размеру это один из самых больших каталогов файловой системы. В этот каталог устанавливаются практически все программы.

/var — содержит файлы, которые подвергаются наиболее частому изменению.

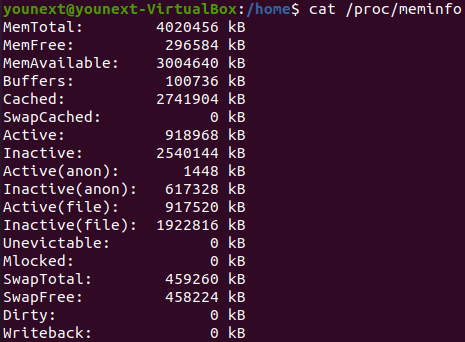
Файловая система /proc является механизмом для ядра и его модулей, позволяющим посылать информацию процессам (отсюда и название /proc). С помощью этой виртуальной файловой системы Мы можем работать с внутренними структурами ядра, получать полезную информацию о процессах и изменять установки (меняя параметры ядра).

* /proc/cpuinfo - информация о процессоре (модель, семейство, размер кэша и т.д.)

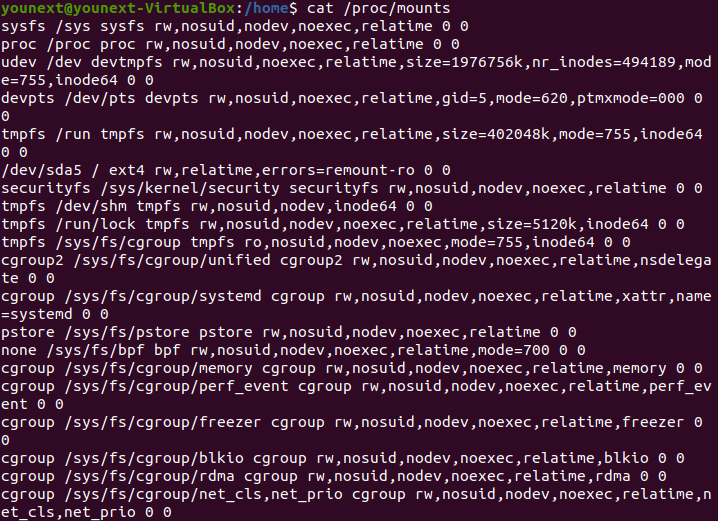
Информация о процессоре:



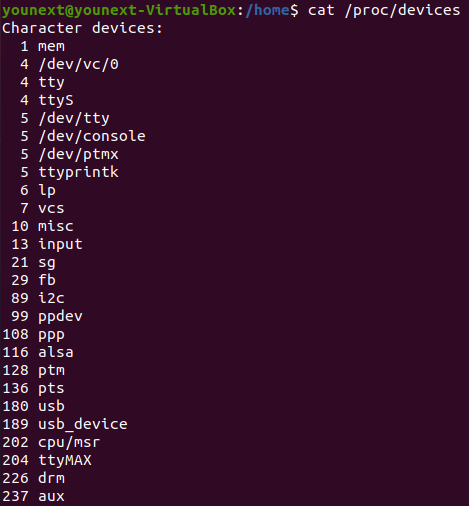
* /proc/meminfo - информация о RAM, размере свопа и т.д.



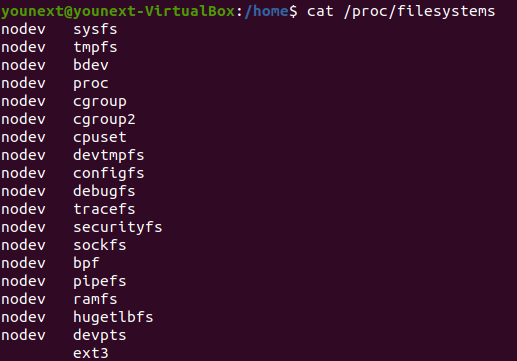
* /proc/mounts - список подмонтированных файловых систем.



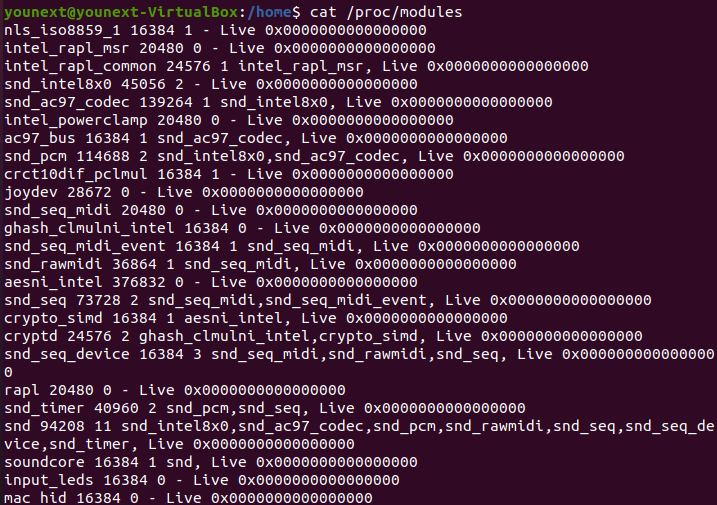
* /proc/devices - список устройств.



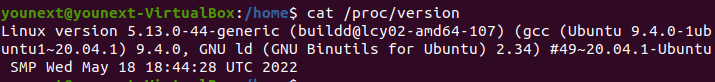
* /proc/filesystems - поддерживаемые файловые системы.



* /proc/modules - список загружаемых модулей.



* /proc/version - версия ядра.



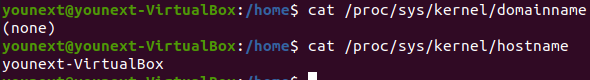
* /proc/cmdline - список параметров, передаваемых ядру при загрузке.



* /proc/self подкаталог - с его помощью приложение найдет информацию о себе. /proc/self является символической ссылкой на каталог процесса обращающегося к /proc.



* В каталоге /proc/sys/kernel находится информация общего плана для ядра. Соответственно в /proc/sys/kernel/{domainname, hostname} находится информация о доменном имени и host имени, которую пользователь может изменить.



# **Вывод**

В данной лабораторной работе мы установили данную ОС, зарегистрировалась в системе. Научились пользоваться диалоговым руководством. Ознакомились со справочной системой в формате HOW-TO. Научились управлять учетными записями пользователя на данной ОС. Совершили работу с устройствами и файловой системой. Управляли некоторыми файлами, совершили операции над группами файлов, совершили работу с командами вывода, поиска и фильтра. Описали и применили функции управления доступом к файловой системе. Научились управлять заданиями. Описали базовую структуру Linux.

7.Управление файлами. Операции над файлами, группами файлов и каталогами.

**1. Команды *chown* и *chgrp***

Эти команды служат для смены владельца файла и группы файла. Выполнять смену владельца может только суперпользователь, смену группы может выполнить сам владелец файла или суперпользователь. Для того, чтобы иметь право сменить группу, владелец должен дополнительно быть членом той группы, которой он хочет дать права на данный файл. Формат этих двух команд аналогичен:

[root]# chown sasha имя-файла

[root]# chgrp usersgrp имя-файла

**2. Команда *mkdir***

Команда mkdir позволяет создать подкаталог в текущем каталоге. В качестве аргумента этой команде надо дать имя создаваемого каталога. Во вновь созданном каталоге автоматически создаются две записи: **.** (ссылка на этот самый каталог) и **..** (ссылка на родительский каталог). Чтобы создать подкаталог, вы должны иметь в текущем каталоге право записи. Можно создать подкаталог не в текущем, а в каком-то другом каталоге, но тогда необходимо указать путь к создаваемому каталогу:

[user]$ mkdir /home/sasha/labrab/sem5/labrab1

Команда mkdir может использоваться со следующими опциями:

•-m mode — задает режим доступа для нового каталога (например, -m 755);

•-p — создавать указанные промежуточные каталоги (если они не существуют).

**3. Команда *cat***

Команда cat часто используется для создания файлов (хотя можно воспользоваться и командой touch). По команде cat на стандартный вывод (т. е. на экран) выводится содержимое указанного файла (или нескольких файлов, если их имена последовательно задать в качестве аргументов команды). Если вывод команды cat перенаправить в файл, то можно получить копию какого-то файла:

[user]$ cat file1 > file2

Собственно, первоначальное предназначение команды cat как раз и предполагало перенаправление вывода, так как эта команда создана для конкатенации, т. е. объединения нескольких файлов в один:

[user]$ cat file1 file2 ... fileN > new-file

Именно возможности перенаправления ввода и вывода этой команды и используются для создания новых файлов. Для этого на вход команды cat направляют данные со стандартного ввода (т. е. с клавиатуры), а вывод команды — в новый файл:

[user]$ cat > newfile

После того, как вы напечатаете все, что хотите, нажмите комбинацию клавиш <Ctrl> + <D> или <Ctrl> + <C>, и все, что вы ввели, будет записано в newfile. Конечно, таким образом создаются, в основном, короткие текстовые файлы.

**4. Команда *cp***

Хотя для копирования файлов иногда пользуются командой cat, но в Linux существует для этого специальная команда cp. Ее можно применять в одной из двух форм:

[user]$ cp [options] source destination

[user]$ cp [options] source\_directory new\_directory

В первом случае файл или каталог source копируется, соответственно, в файл или каталог destination, а во втором случае файлы, содержащиеся в каталоге source\_directory копируются в каталог new\_directory. Для копирования надо иметь права на чтение файлов, которые копируются, и права на запись в каталог, в который производится копирование.

Если в качестве целевого указывается существующий файл, то его содержимое будет затерто, поэтому при копировании надо соблюдать осторожность. Впрочем, можно использовать команду cp с опцией -i, тогда перед перезаписью существующего файла будет запрашиваться подтверждение.

**5. Команда *mv***

Если вам необходимо не скопировать, а переместить файл из одного каталога в другой, вы можете воспользоваться командой mv. Синтаксис этой команды аналогичен синтаксису команды cp. Более того, она сначала копирует файл (или каталог), а только потом удаляет исходный файл (каталог). И опции у нее такие же, как у cp.

Команда mv может использоваться не только для перемещения, но и для переименования файлов и каталогов (т. е. перемещения их внутри одного каталога). Для этого надо просто задать в качестве аргументов старое и новое имя файла:

[user]$ mv oldname newname

Но учтите, что команда mv не позволяет переименовать сразу несколько файлов (используя шаблон имени), так что команда mv \*.xxx \*.yyy не будет работать. При использовании команды mv, также как и при использовании cp, не забывайте применять опцию -i для того, чтобы получить предупреждение, когда файл будет перезаписываться.

**6. Команды *rm* и *rmdir***

Для удаления ненужных файлов и каталогов в Linux служат команды rm (удаляет файлы) и rmdir (удаляет пустой каталог) . Для того, чтобы воспользовался этими командами, вы должны иметь право записи в каталоге, в котором расположены удаляемые файлы или каталоги. При этом полномочия на изменение самих файлов не обязательны. Если хотите перед удалением файла получить дополнительный запрос на подтверждение операции, используйте опцию -i.

Если вы попытаетесь использовать команду rm (без всяких опций) для удаления каталога, то будет выдано сообщение, что это каталог, и удаления не произойдет. Для удаления каталога надо удалить в нем все файлы, после чего удалить сам каталог с помощью команды rmdir. Однако можно удалить и непустой каталог со всеми входящими в него подкаталогами и файлами, если использовать команду rm с опцией -r.

Если вы дадите команду rm \*, то удалите все файлы в текущем каталоге. Подкаталоги при этом не удалятся. Для удаления как файлов, так и подкаталогов текущего каталога надо тоже воспользоваться опцией -r. Однако всегда помните, что в Linux нет команды восстановления файлов после их удаления (даже если вы спохватились сразу же после ошибочного удаления файла или каталога)! Так что дважды подумайте до удаления чего-либо и не пренебрегайте опцией -i.

