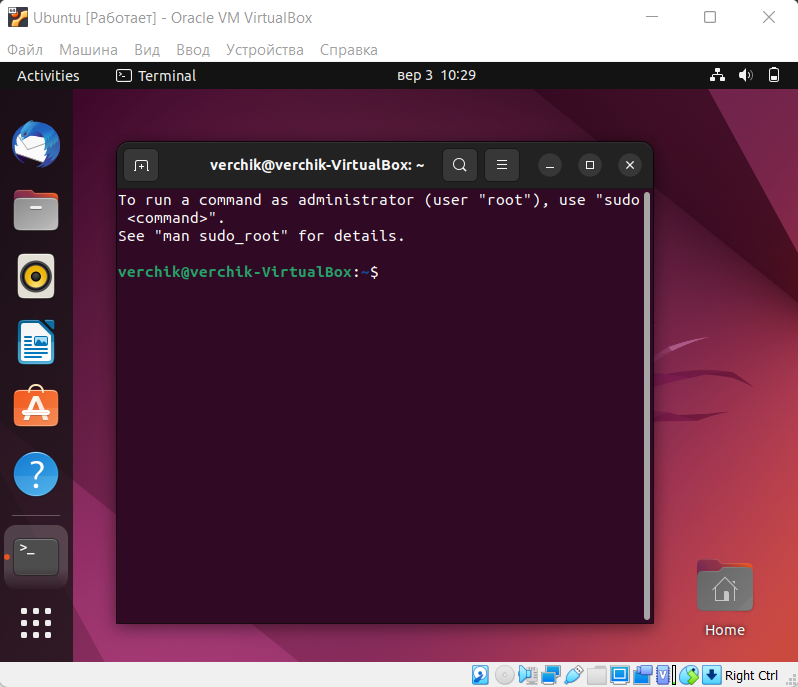
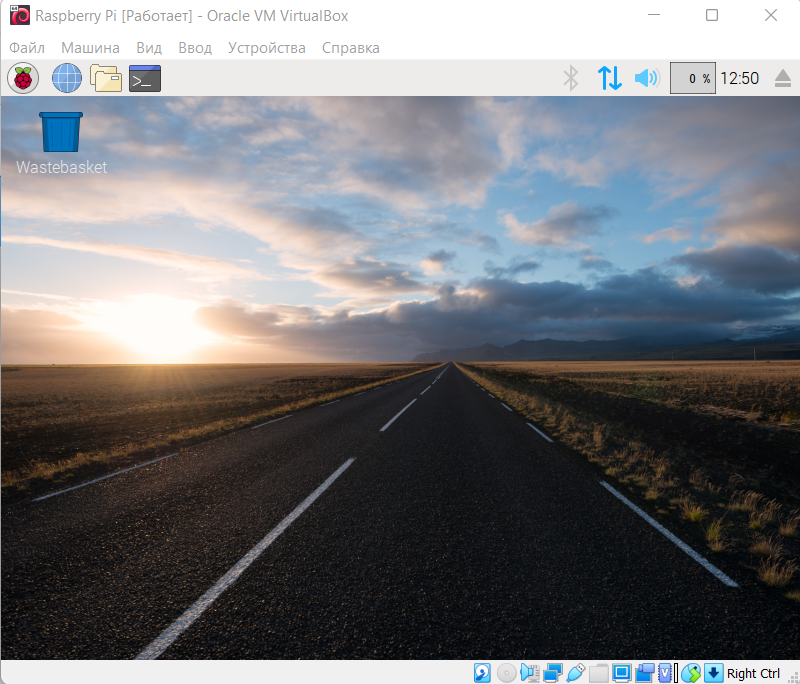
**Задание 1. Развертывание виртуальных машин под управлением ОС Raspberry PI и ОС Ubuntu**

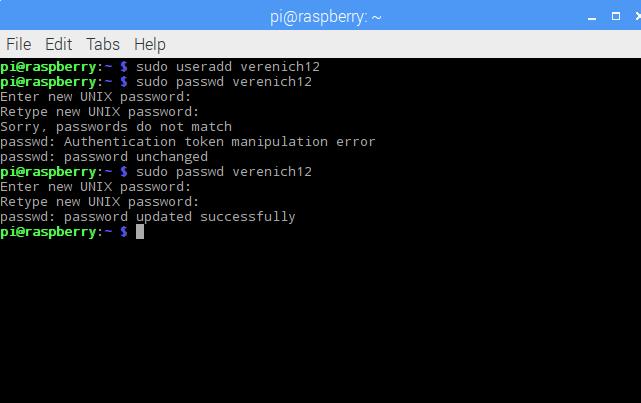
1.1. Изучить материалы учебного курса Raspbian x86 on VirtualBox on a Windows PC и Как установить Ubuntu в VirtualBox.

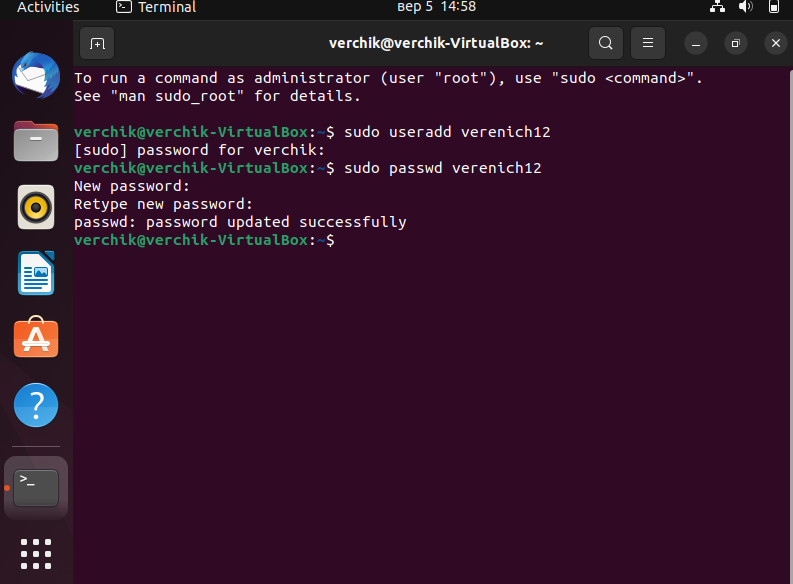
1.2. Установить OC Raspberry PI Desktop. Учесть, что в процессе установки создается учетная запись с именем pi и паролем raspberry. Пароль после установки рекомендуется сменить. Созданный пользователь обладает правами администратора. Продемонстрировать процедуру установки скриншотами в отчете.





1. Установить OC Ubuntu Mate или Kali Linux. В процессе установки создать пользователя с именем по следующему шаблону ФамилияN, где Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13. Созданный пользователь обладает правами администратора. Продемонстрировать процедуру установки скриншотами в отчете.





**Задание 2. Настройка ssh-соединений между хостовой системой и гостевыми машинами**

2.1. Изучить материалы из учебного курса по настройке ssh в ОС Raspberry PI и ОС Ubuntu:

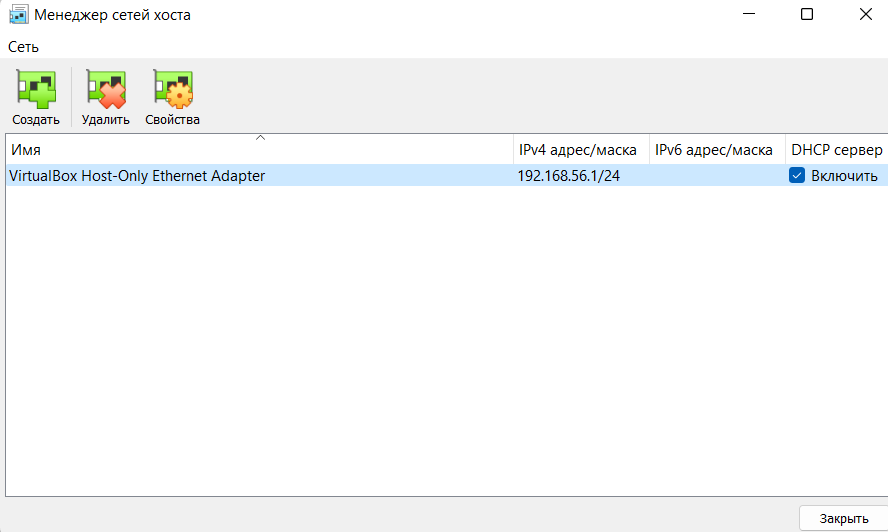
◦ Как настроить ssh-соединение в гостевой ОС в Virtualbox

◦ Как настроить ssh в ОС Raspbian ◦ Как настроить ssh ОС UBuntu

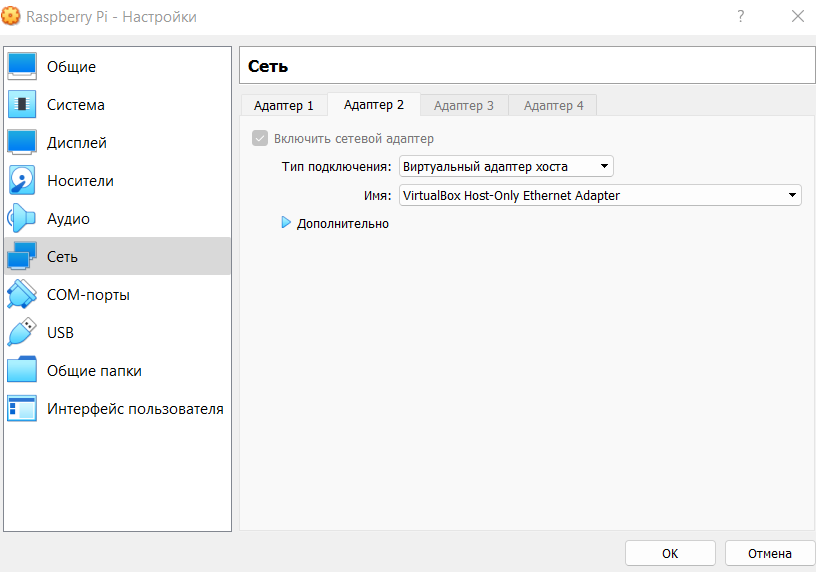
◦ Как пользоваться SSH? А также установка и настройка (Linux, Windows, macOS, Android, iOS)

2.2. Настроить ssh-соединение в гостевой виртуальной ОС Raspberry PI и Ubuntu. Основные этапы настройки продемонстрировать скриншотами в отчете.

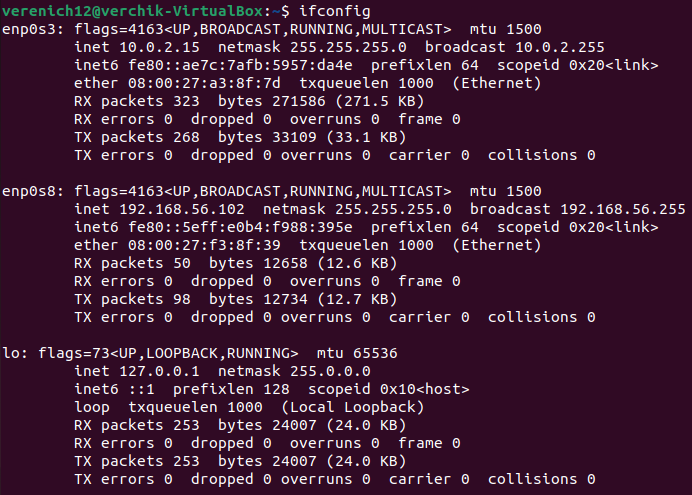
1) активируем DHCP-клиент для сетевого адаптера хоста(Ubuntu)



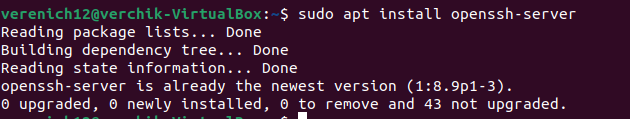
2) Добавим сетевой адаптер для хоста



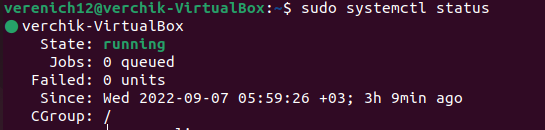
3) Проверяем настройки сетевых адаптеров в хосте



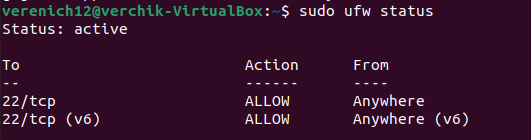
4) Установим SSH на виртуальную машину, которая будет сервером



Проверим статус

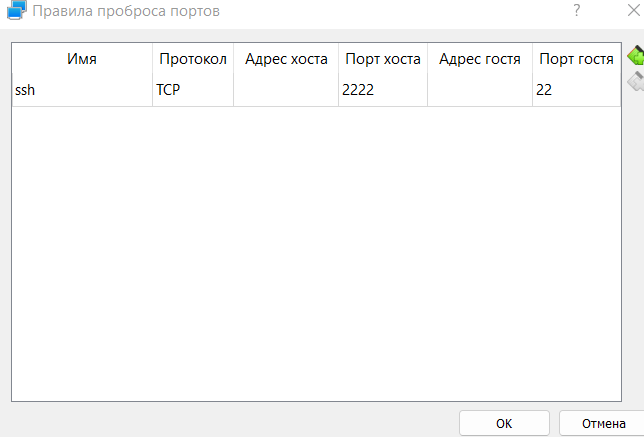


5) Откроем SSH-порт в хосте и проверим его состояние

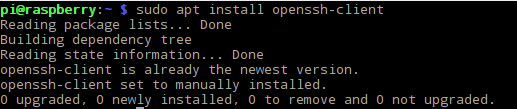


6) Теперь настроим сеть в виртуальной машине

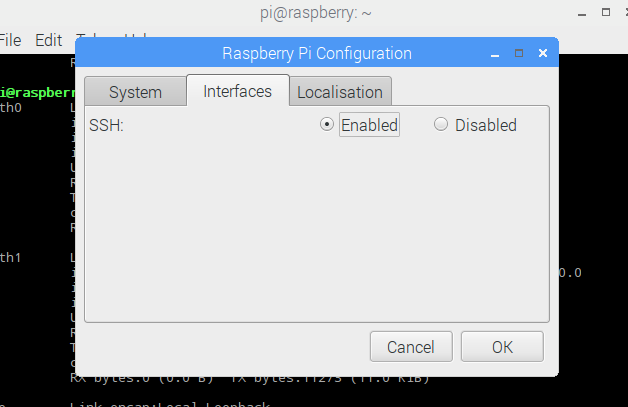




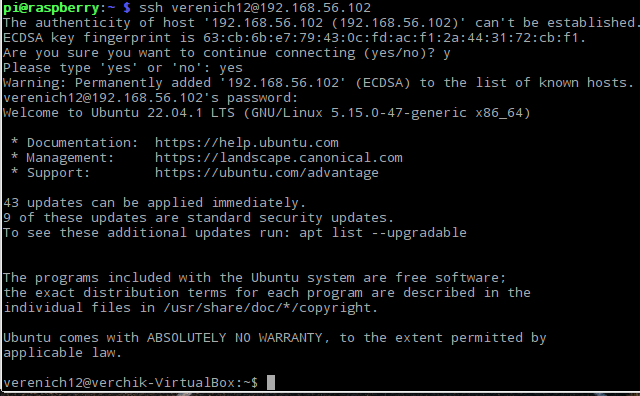
7) Установим SSH–клиент на виртуальной машине, которая будет клиентом



8) Включим SSH в гостевой виртуальной машине



9) подключимся из гостевой Raspberry PI OS в Ubuntu



2.3. Подключиться по ssh из ОС Raspberry PI в Ubuntu и продемонстрировать скриншотами в отчете.

СДЕЛАНО ВЫШЕ

2.4. В зависимости от вашей основной операционной системы изучите рекомендации:

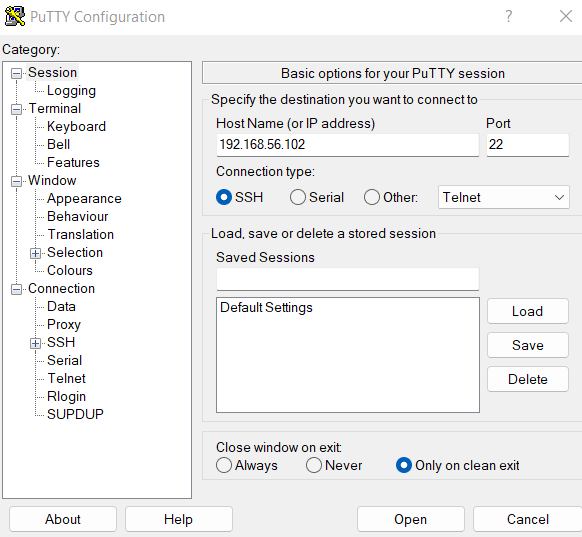
◦ Как пользоваться SSH? А также установка и настройка (Linux, Windows, macOS, Android, iOS)

◦ Для Windows: «Как активировать ssh-клиент в Windows 10» или установите клиент Putty для создания терминальных сессий, используя инструкцию «Как подключиться по ssh из Windows».

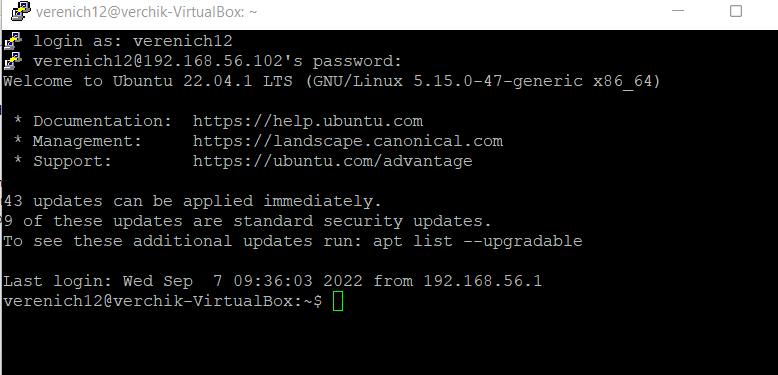
◦ Для macOS: «Как использовать встроенный ssh-клиент в macOS»

2.5. Открыть ssh-клиент в вашей хостовой системе Windows 10 или macOS. Подключитесь из хостовой машины в гостевые виртуальные машины под управлением ОС Raspberry PI и Ubuntu. Процесс подключения продемонстрировать скриншотами в отчете.

1) Настроим PuTTy(подключимся к уже настроенной выше хостовой системе Ubuntu)



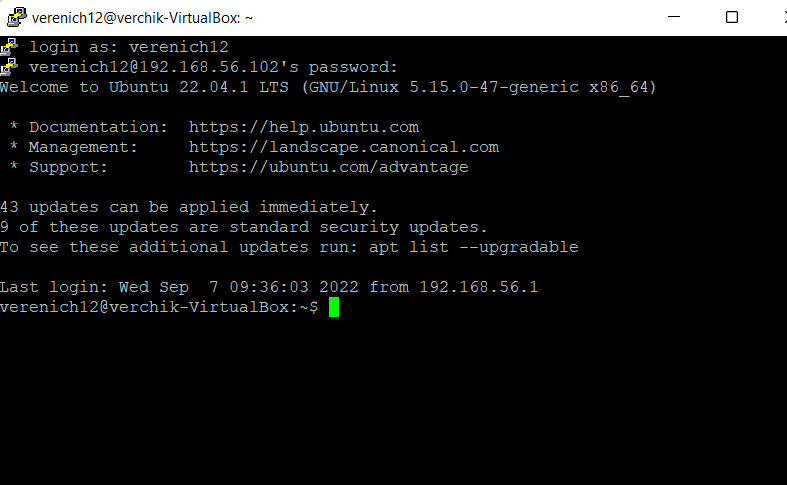
2) Результат



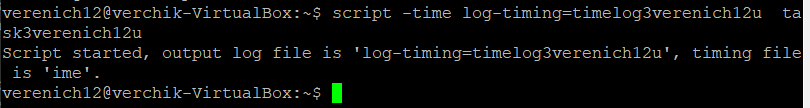
**Задание 3. Использование команды script для протоколирования командам**

3.1. Изучить руководство по команде script в материале Руководство по команде script из учебного курса. Обратить внимание на пример команды, которая позволяет протоколировать команды и вести журнал меток времени (пример 4).

3.2. Подключиться из хостовой ОС по ssh под пользователем с правами администратора в Ubuntu с именем ФамилияN, где Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13.

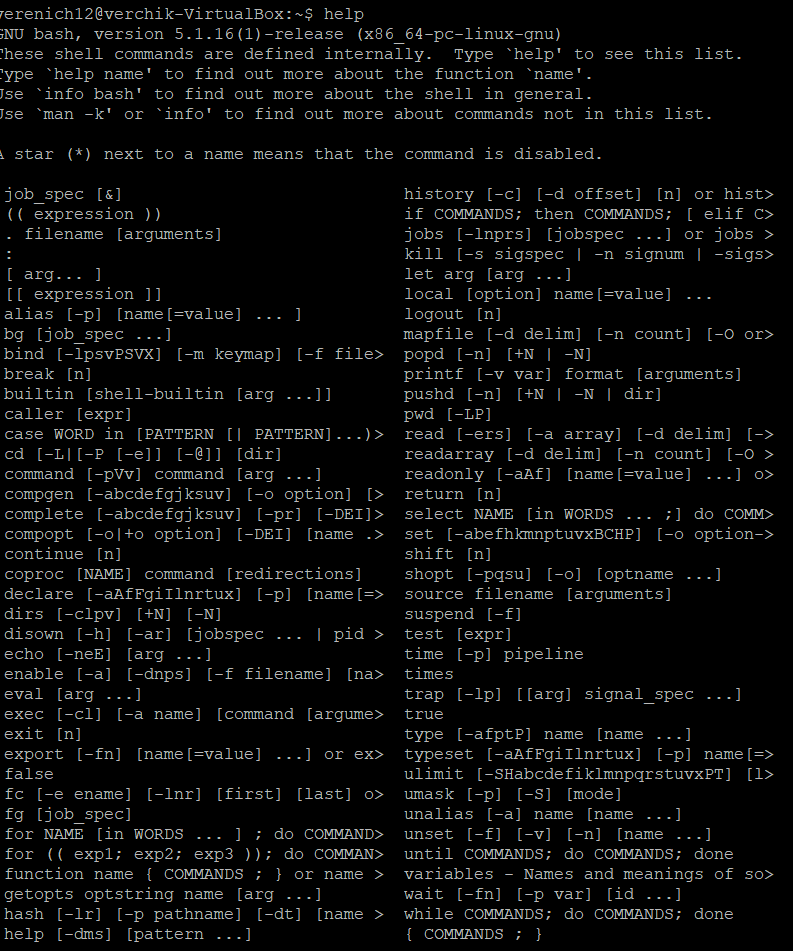
script

3.3. Включить ведение протокола командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.

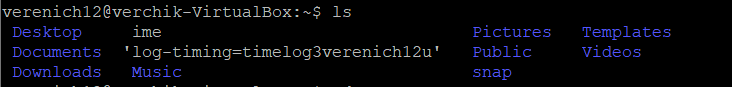


3.4. Выполните команды help, ls, cd, pwd, mkdir, rmdir, rm, history. Если требуется изучите описания команд с помощью man и info.

help



ls



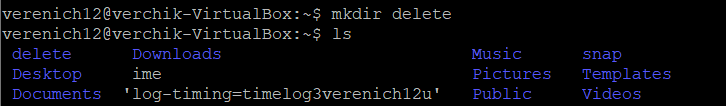
cd



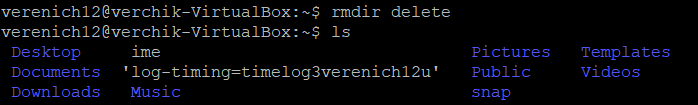
pwd



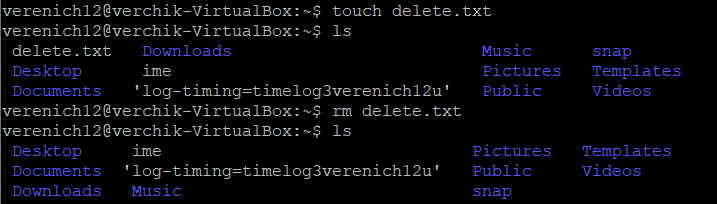
mkdir



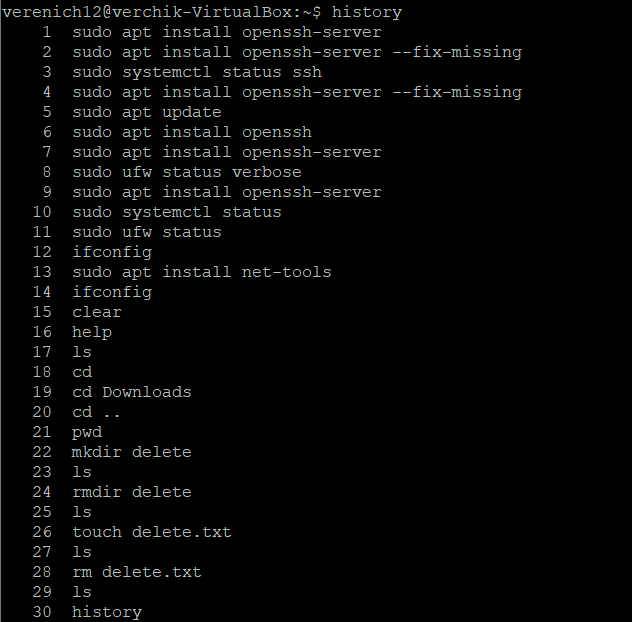
rmdir



rm



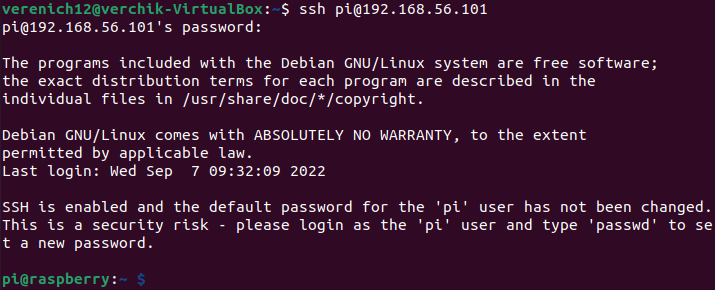
history



3.5. Завершить протоколирование команд в Ubuntu, т. е.ввести команду exit.



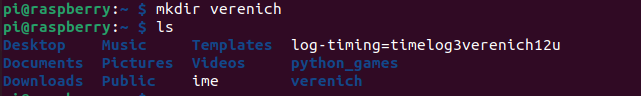
3.6. Не выключая Ubuntu, включите гостевую ОС Raspberry PI. Подключитесь по ssh к гостевой виртуальной ОС Raspberry PI под учетной записью пользователя pi.



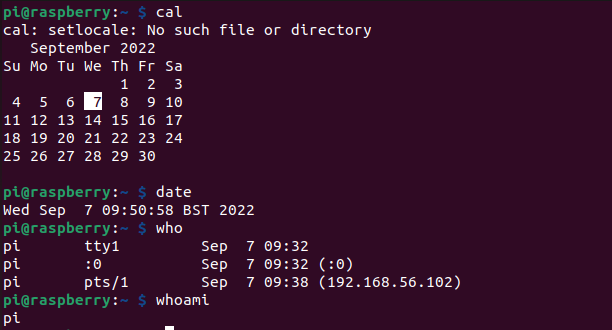
3.7. Включить ведение протокола командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.



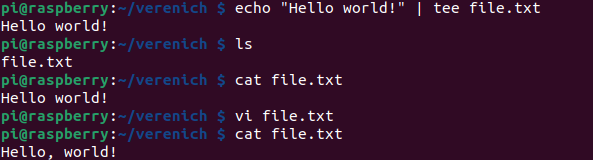
3.8. Создайте каталог, задав в качестве имени вашу фамилию латиницей строчными буквами. Данные каталог использовать для размещения результатов лабораторной работы в ОС Raspberry PI и сохраняйте в него результаты работы.



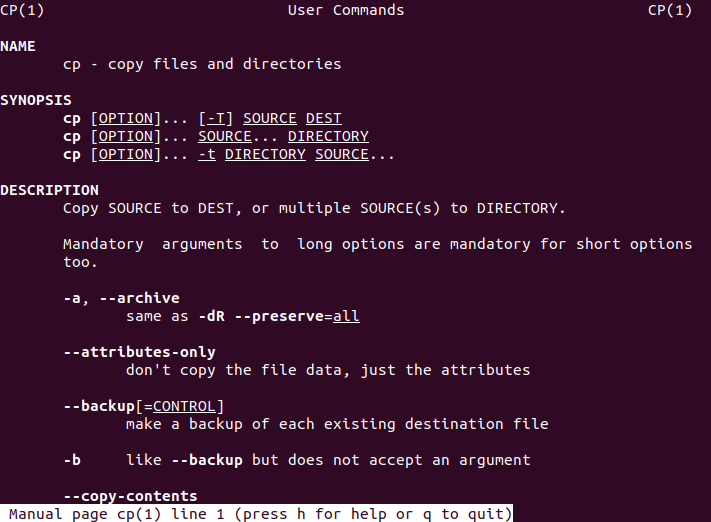
3.9. Выполнить команды cal, date, who, whoami.

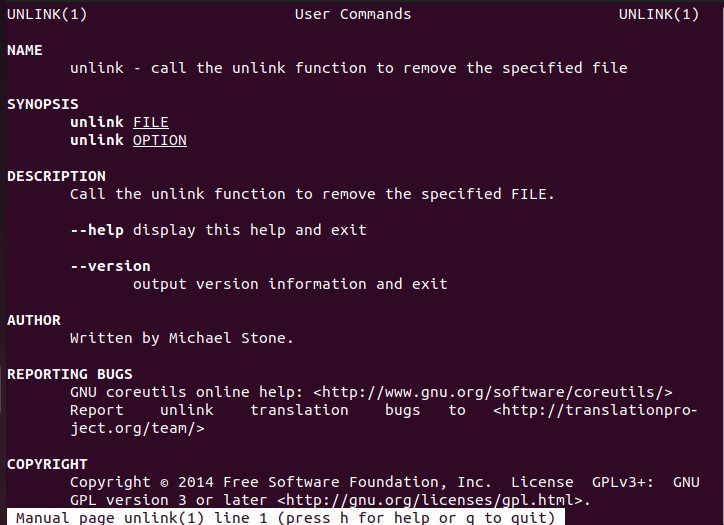


3.10. Создать текстовый файл в ОС Raspberry PI с произвольным содержанием при помощи команды tee. Просмотреть содержимое файла при помощи команды cat и исправить в нем ошибки при помощи текстового редактора vi.

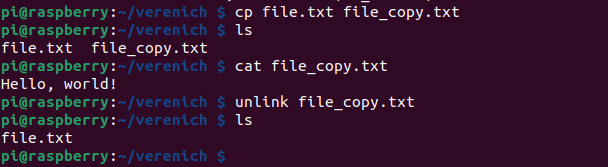


3.11. Изучить справку на команды cp, unlink.





3.12. Создать копию файла, созданного в п. 3.10, командой cp, удалить ее командой unlink.



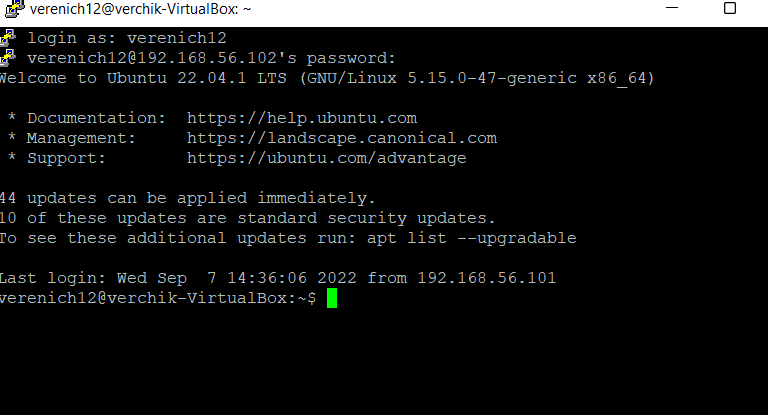
3.13. Завершить протоколирование команд в Raspberry PI, т. е. ввести команду exit.

Exit

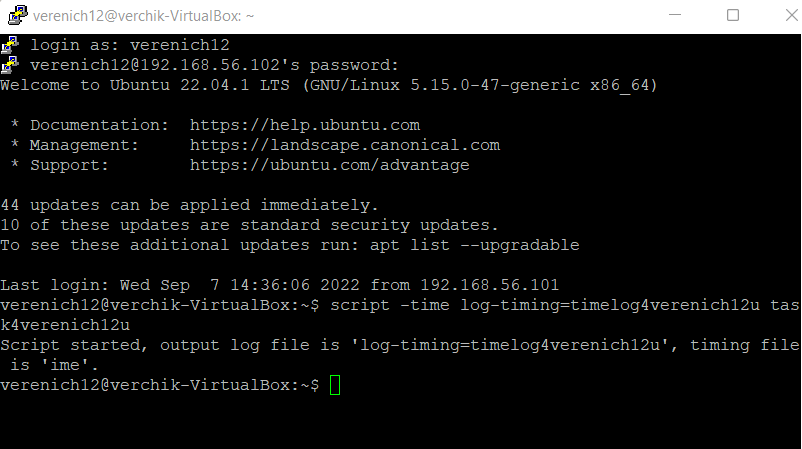


**Задание 4. Безопасное копирование файлов и публикация в репозиторий**

4.1. Если время действия ssh-сессии к гостевой ОС Ubuntu истекло, подключиться по ssh к ней.



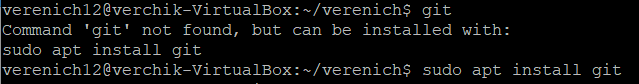
4.2. Включить ведение протокола командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.



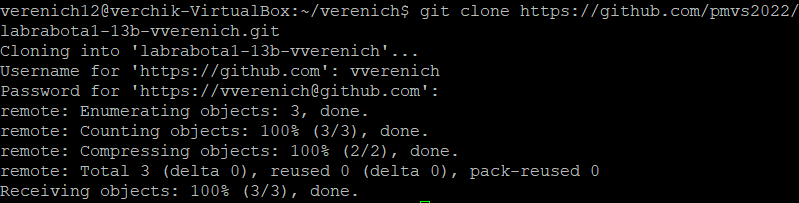
4.3. Создайте каталог, задав в качестве имени вашу фамилию латиницей строчными буквами. Данные каталог использовать для размещения результатов лабораторной работы в ОС Ubuntu.



4.4. Установить git, если требуется.



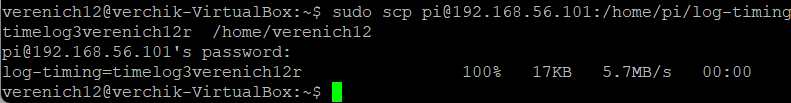
4.5. Перейдите в каталог, созданный в п 4.3. Склонируйте репозиторий из github. 7



4.6. Скопируйте в каталог, созданный в п. 4.3, результаты работы в Ubuntu, полученные в ходе выполнения задания 3.

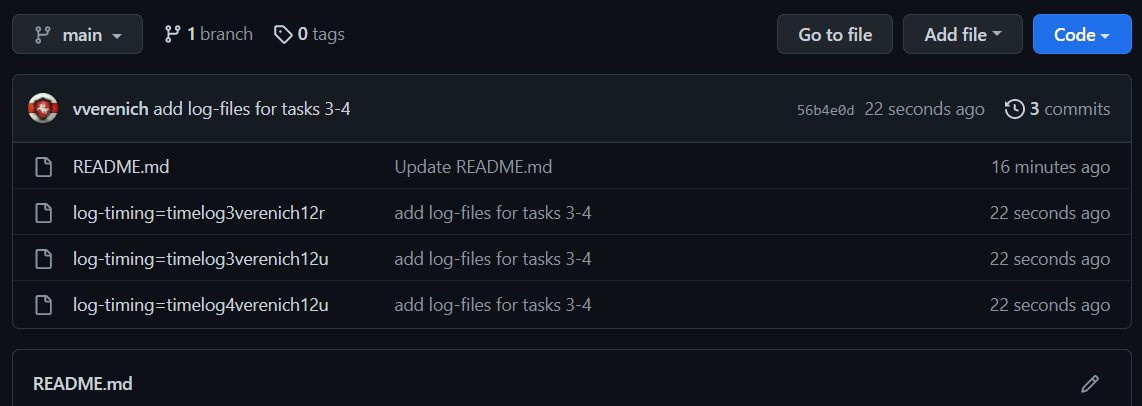


4.7. Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.



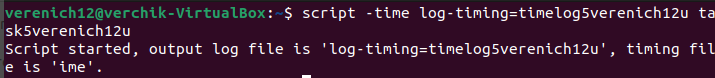
4.8. Завершите протоколирование команд с помощью script, т. е. введите команду exit.

4.9. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения заданий 3 и 4, в удаленный.

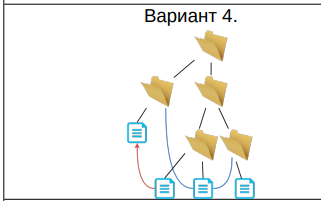


**Задание 5. Управление файлами и папками**

5.1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNKM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, K — номер варианта, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNKM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, K — номер варианта, M — r — для Raspberry PI, u — для Ubuntu.

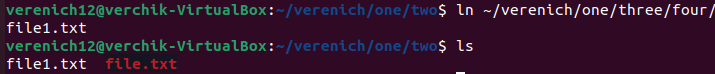


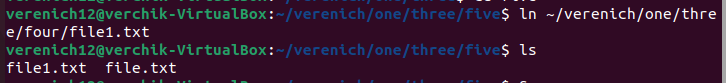
5.2. Создайте структуру каталогов в соответствии с вариантом в ОС Ubuntu. Черными линиями представлена вложенность файлов/подкаталогов в каталоги. Синими линиями представлены жесткие ссылки. Красными линиями — символические ссылки. Стрелка на красной линии указывает на целевой файл ссылки. Файлы создаются копированием ранее созданного файла командой cp с внесением в копии некоторых изменений. Ссылки создаются командой ln, символические ссылки — ей же, но с ключом -s:











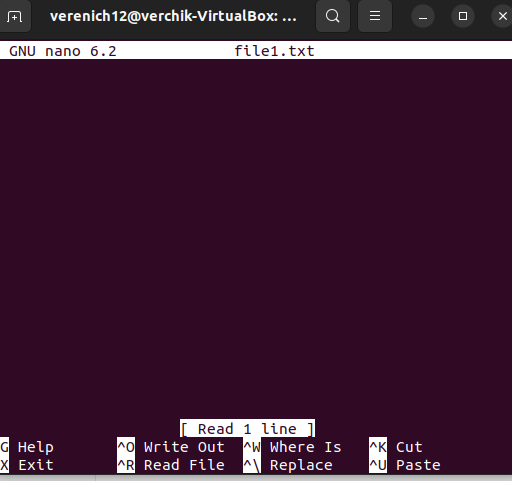
5.3. Для всех вариантов выполнить следующие действия:

◦ Создать ссылки (синие линии).

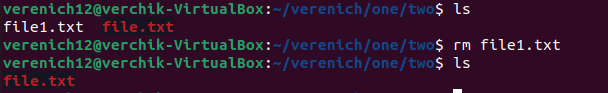
◦ Создать символические ссылки (красные линии).

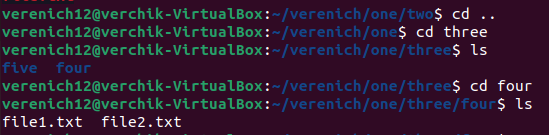
◦ Провести ряд экспериментов, иллюстрирующих доступ к файлам по основным именам, по ссылкам и по символическим ссылкам. Для доступа использовать команду cat или редактор vi.

Доступ к ~/verenich/one/three/four/file1.txt из ~/verenich/one/two



◦ Провести ряд экспериментов, иллюстрирующих реакцию системы на удаление файла, на который имеются ссылки, и файла, на который имеются символические ссылки. Проверять результаты командой ls -la.

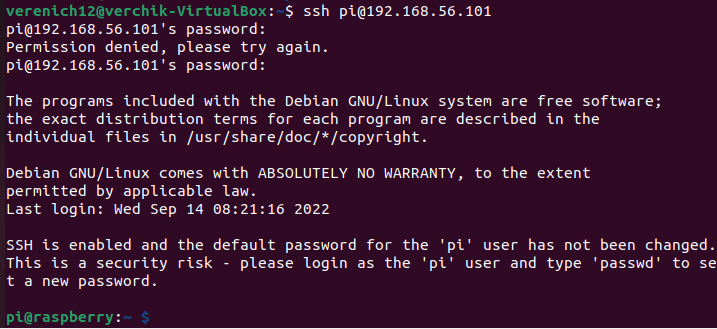




5.4. Завершите протоколирование команд с помощью script, т. е. введите команду exit.



5.5. Подключитесь по ssh из Ubuntu в ОС Raspberry PI и выполните шаги 5.1- 5.4.



Начинаю вести протокол



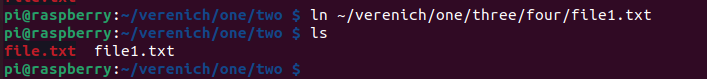
Создаю символическую ссылку

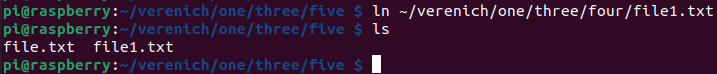


Проверяю создана ли ссылка

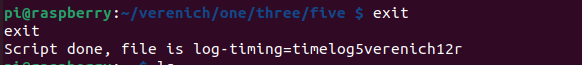


Создаю жесткие ссылки

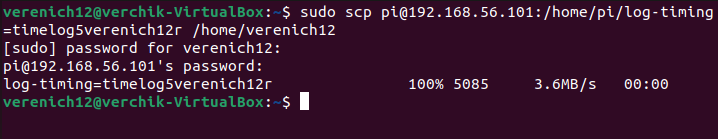




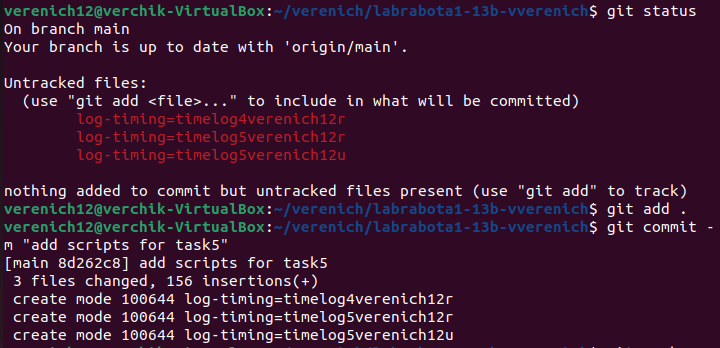
Завершаю протоколирование

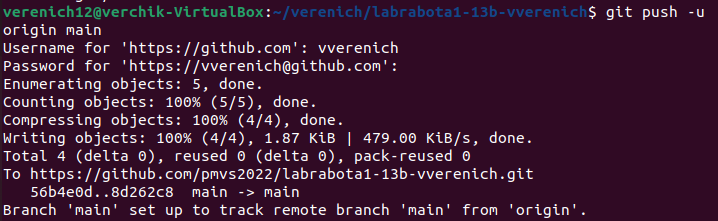


5.6. Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.



5.7. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 5, в удаленный.

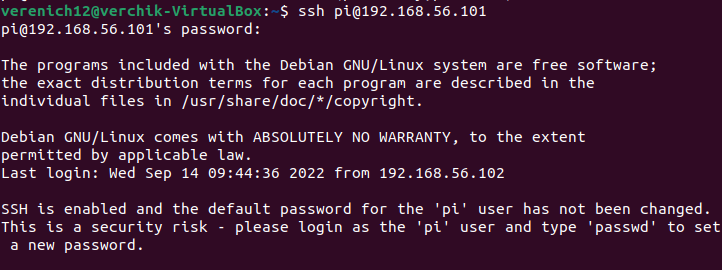




**Задание 6. Управление пользователями и права доступа**

6.1. Изучить справку к командам useradd, groupadd, chmod и chown.

6.2. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.



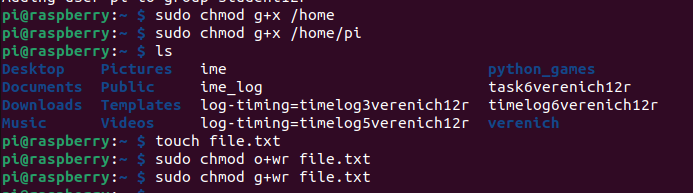
6.3. Добавить группу studentNM в ОС Ubuntu, где N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.



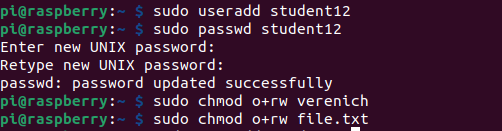
6.4. Добавить своего пользователя для каждой ОС в созданную группу.



6.5. Открыть для своей основной группы доступ к своему домашнему каталогу — для поиска в каталоге и к рабочей копии файла в домашнем каталоге — для чтения и записи.

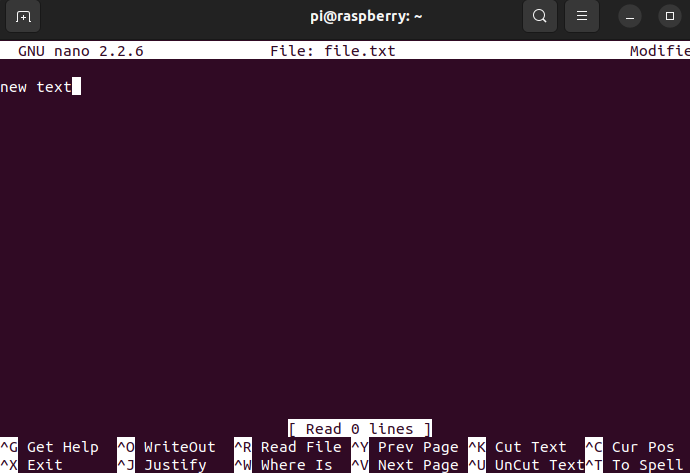


6.6. Создать пользователя studentNM в ОС Ubuntu, где N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Открыть доступ на чтение и запись к одному из свиох каталогов и одному из своих файлов для созданного пользователя.



6.7. Подключиться по ssh с созданной учетной записью или авторизоваться в гостевой ОС. И проверить права доступа на свой каталог и файл, внеся в них изменения.

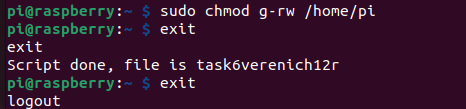




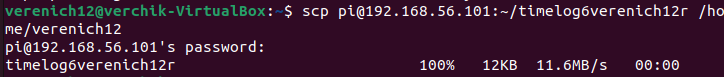
6.8. Переключиться в свою учетную запись. Проверить изменения.

6.9. Закрыть доступ к своему домашнему каталогу.

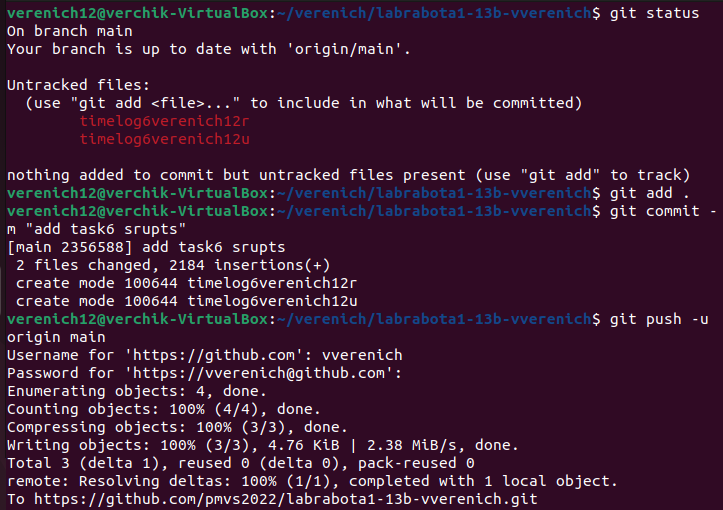
6.10. Завершите протоколирование команд с помощью script, т. е. введите команду exit.



6.12. Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.



6.13. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 6, в удаленный.

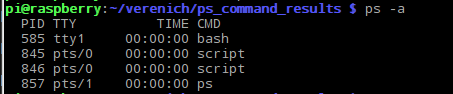


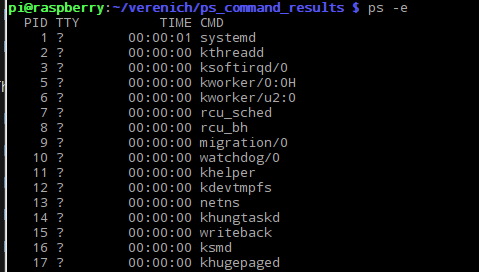
**Задание 7. Базовые команды для управления процессами**

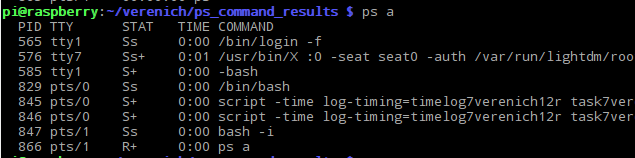
7.1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал 10 меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.



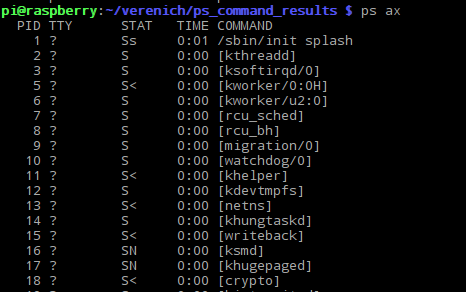
7.2. Изучите справку к команде ps и статью https://www.tecmint.com/pscommand-examples-for-linux-process-monitoring/, выполнить ее с ключами -a, -e, a, x, ax в ОС Raspberry PI, записать результаты в файл, например: ps -e > ps.log.



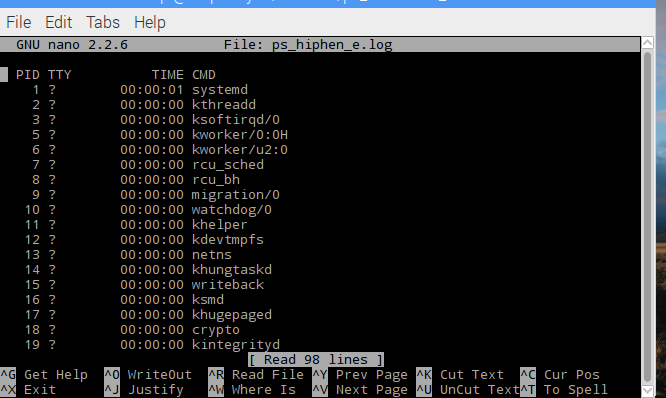




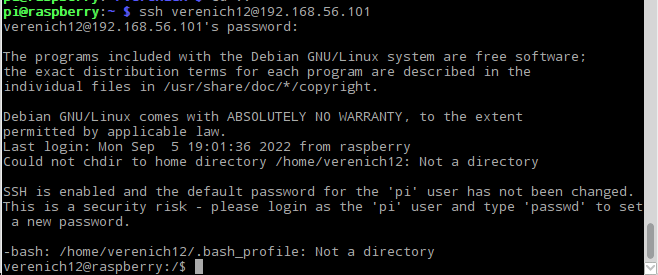


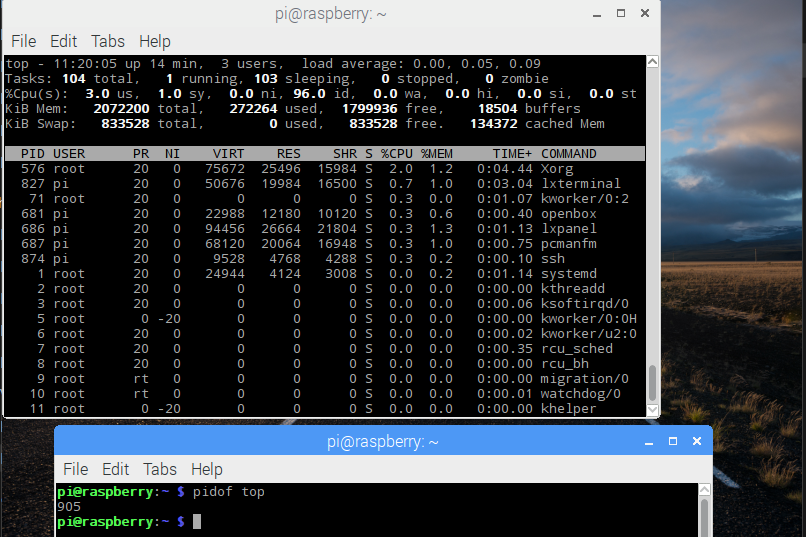






7.3. Авторизуйтесь в ОС Ubuntu по ssh. Изучите справку к команде pidof и статью https://www.tecmint.com/find-process-name-pid-number-linux/. Запустите команду top. А в другом терминале, используя pidof определите ее идентификатор процесса. Завершите выполнение команды top.

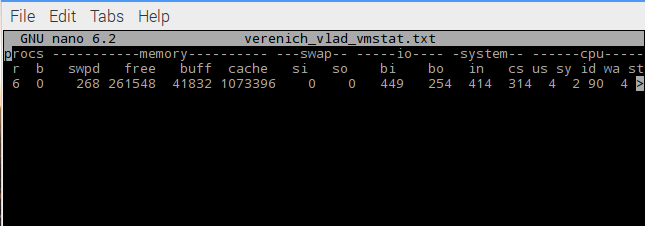




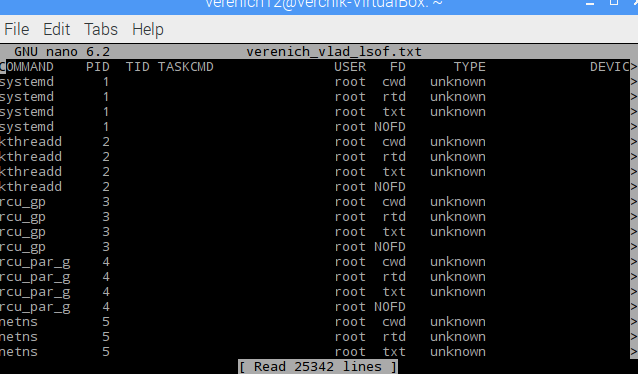
7.4. Авторизуйтесь в ОС Ubuntu по ssh. Изучите справку к команде fuser и статью https://www.tecmint.com/learn-how-to-use-fuser-command- with-examples-inlinux/. Откройте любой текстовый файл на редактирование. Определите, какие процессы используют ваш файл.



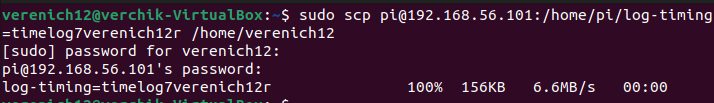
7.5. Авторизуйтесь в ОС Ubuntu по ssh. Изучите справку к команде vmstat. Результаты вывода команды запишите в файл с указанием в названии файла вашу фамилию и имя и название используемой команды.



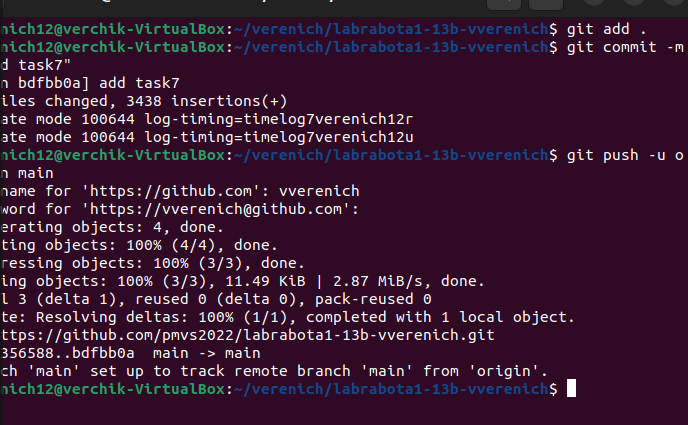
7.6. Авторизуйтесь в ОС Ubuntu по ssh. Изучите справку к команде lsof. Результаты вывода команды запишите в файл с указанием в названии файла вашу фамилию и имя и название используемой команды.



7.7. Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.



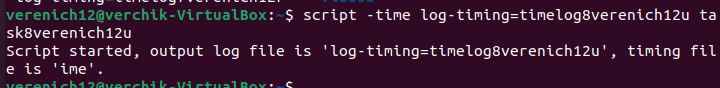
7.8. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 7, в удаленный.



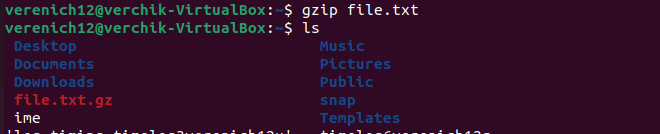
**Задание 8. Сжатие и архивация файлов**

8.1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu.

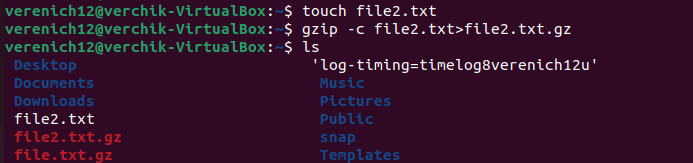
8.2. Включить запись протокола команд и меток времени командой script. 11



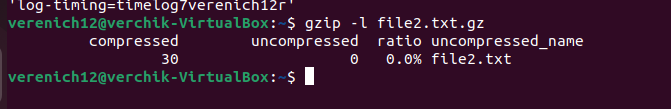
8.3. Подключиться по ssh со своей учетной записью к ОС Ubuntu или авторизоваться в виртуальной машине. Используя программу gzip, сжать файл любой текстовый файл в созданном каталоге согласно варианту из задания 5.



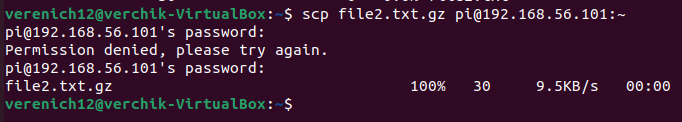
8.4. Используя программу gzip, сжать файл любой текстовый файл в созданном каталоге согласно варианту из задания 5, чтобы исходный файл остался в текущем каталоге.

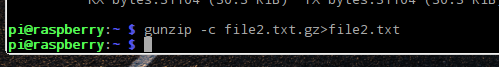


8.5. Проверить целостность сжатого файла.

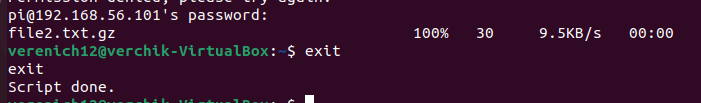


8.6. Скопировать полученный архив с помощью scp в домашний каталог своего пользователя в ОС Raspberry PI. Распаковать файл архивы \*.gz в домашний каталог в ОС Raspberry PI.

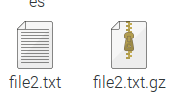




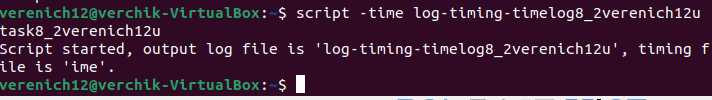
8.7. Завершить протоколирование команд.



8.8. Проверить содержимое домашнего каталога в ОС Raspberry PI. Утилита tar



8.9. Включить в ОС Ubuntu запись протокола команд и меток времени командой script.



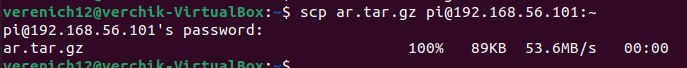
8.10. Создать в домашнем каталоге архив формата tar, включающий каталог и файлы.



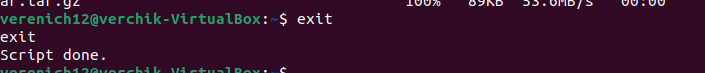
8.11. Сжать полученный архив.



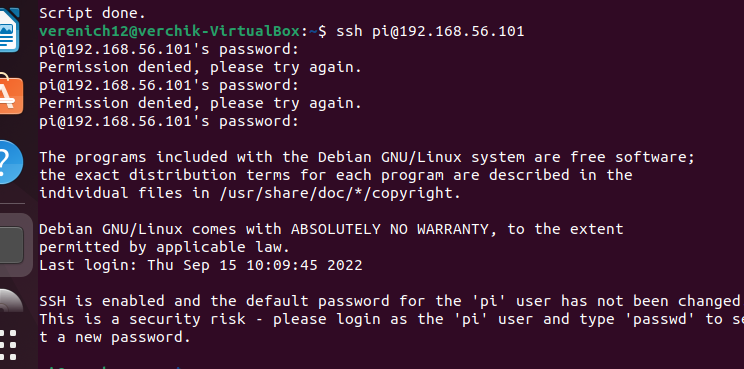
8.12. Скопировать полученный архив с помощью scp в домашний каталог своего пользователя в ОС Raspberry PI.

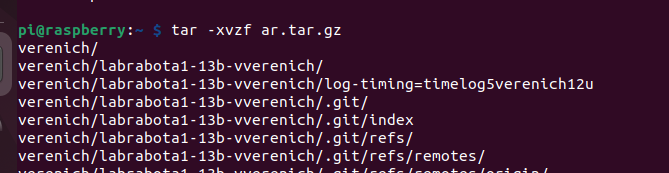


8.13. Завершить протоколирование команд.



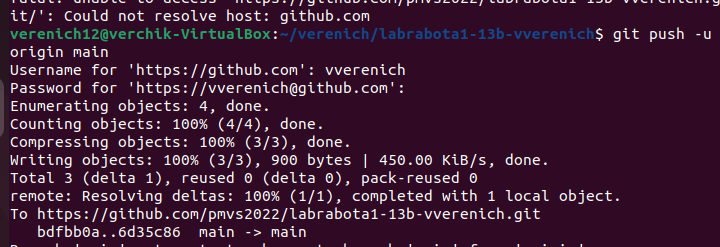
8.14. Подключиться к ОС Raspberry PI и распаковать архив.





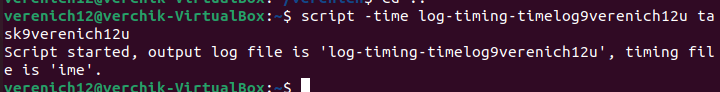
8.15. Скопируйте сохранённые протоколы работы и добавить в gitрепозиторий, созданный в ОС Ubuntu в п.4.5.

8.16. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 8, в удаленный.



**Задание 9. Синхронизация в Linux с помощью утилиты rsync**

9.1. Задание выполнить в ОС Ubuntu и ОС Raspberry PI. Для каждой ОС вести протокол командой script с журналом меток времени. Протокол назвать по следующему шаблону — taskXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. Журнал меток назвать по следующему шаблону — timelogXФамилияNM, где X — номер выполняемого задания, Фамилия — заменить на вашу фамилию латиницей и строчными буквами, N — номер группы, например 12 или 13, M — r — для Raspberry PI, u — для Ubuntu. 12



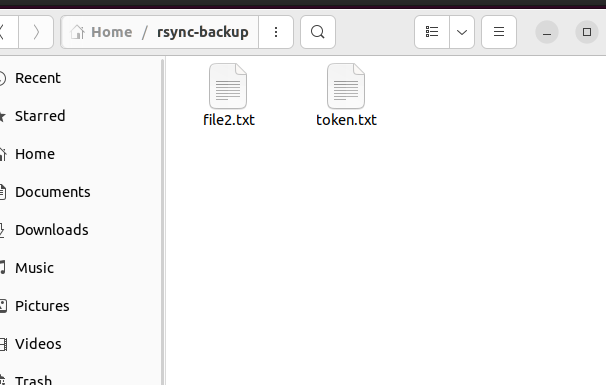


9.2. Создать в домашнем каталоге каталог rsync-backup в ОС Ubuntu и в ОС Raspberry PI.

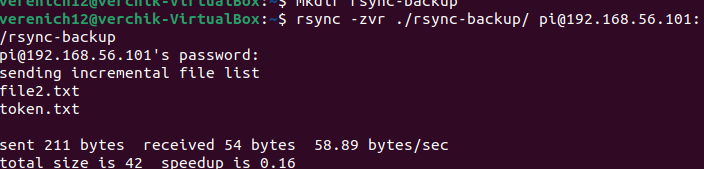




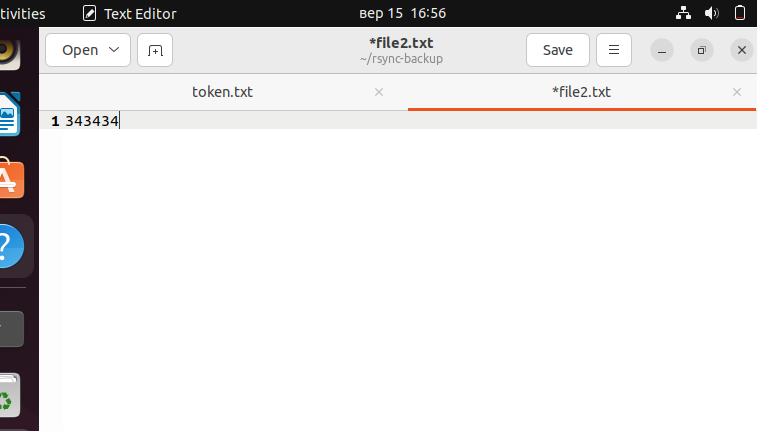
9.3. Скопировать несколько файлов в каталог rsync-backup в ОС Ubuntu.

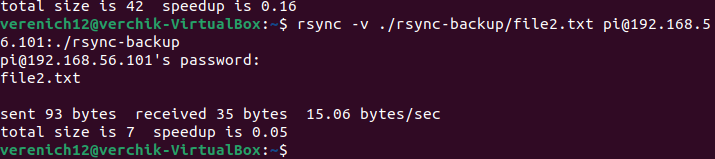


9.4. Используя rsync, скопировать содержимое каталога rsync-backup из ОС Ubuntu в каталог rsync-backup в ОС Raspberry PI.

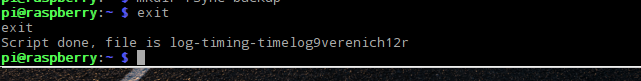


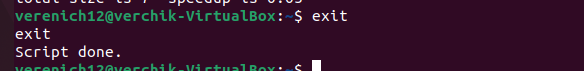
9.5. Внести изменения в любой из файлов в каталоге rsync-backup в ОС Ubuntu и синхронизировать изменения с каталогом в ОС Raspberry PI.



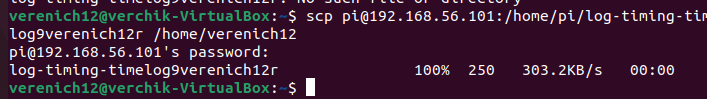


9.6. Скопируйте сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.





9.7. Опубликуйте изменения из локального репозитория, т. е. результаты выполнения задания 9, в удаленный.





**Задание 10. Подключение по ssh к контейнеру Docker**

10.1. Задание выполнить в хостовой системе. Для Windows и macOS найти документацию по установке контейнера Docker.

10.2. Просмотреть видео https://youtu.be/V7lTLVzsK5U с установкой docker на ОС Ubuntu 18.04.

10.3. Выполнить установку docker согласно документации, например для ОС Ubuntu <https://docs.docker.com/engine/install/ubuntu/>.

10.4. Установить docker и развернуть контейнер для ubuntu.

10.5. Изучить основные команды для работы с docker <https://youtu.be/XtO3hUWLKF8>.

10.6. Установить Docker Compose согласно документации <https://docs.docker.com/compose/install/>.

10.7. Изучить статью https://phoenixnap.com/kb/how-to-ssh-into-docker-container и <https://andreyex.ru/linux/kak-podklyuchitsya-po-ssh-k-kontejneru-docker/>.

10.8. Для контейнера ubuntu настроить подключение по ssh, скопировать ключ ssh с хостовой системы и подключиться к контейнеру docker ubuntu по ssh. Продемонстрировать подключение по ключу ssh (без применения пароля) и по нестандартному порту для ssh (стандартный пор для ssh — 22).

10.9. Изучить статьи для создания контенейра Docker для тестирования сборки для Android: https://andresand.medium.com/android-emulator-on-docker-containerf20c49b129ef https://betterprogramming.pub/build-a-lightweight-docker-container-forandroid-testing-2aa6bdaea422 <https://andresand.medium.com/building-android-with-docker-8dbf717f54d4>

10.10. Настроить контейнер docker для Android и продемонстировать запуск эмулятора.

Контрольные вопросы.

1. Какие ключи команды ls Вы знаете? Что они дают?

Команда ls выводит в консоль содержимое указанной директории.

Синтаксис: ls [OPTION] [FILE]

Возможные ключи:

-a, --all — выводит также скрытые файлы (начинаются с .)

--author — с -l выводит автора каждого файла

-c — c -lt сортирует вывод по времени последнего изменения, с -l — показывает время последнего изменения и сортирует по имени, иначе сначала более новые файлы.

-d, --directory — показывает директории без их содержимого

-h, --human-readable — c -l, -s показывает размер файла.

-l — выводит более подробную информацию о файлах (права доступа, владельца и автора, дату последнего изменения)

-r, --reverse — обратный порядок при сортировках

-R, --recursive — рекурсивно заходит в поддиректории.

--sort — сортирует по указанному параметру.

2. Чем отличаются man и info? Как с ними работать?

Обе утилиты помогают получить информацию о какой-либо команде. Отличие в том, что man использует less для рендеринга текста и комбинации клавиш из vim, а info — комбинации из emacs.

3. Команда script — назначение и применение.

script используется для логирования команд, вводимых в консоль.

Синтаксис: script [options] [file]

можно использовать опцию -t для записи времени. Эта информация может быть полезна для воспроизведения скрипта.

Опция -a позволяет дополнить существующий скрипт.

4. Какая команда используется для генерации ssh-ключа?

ssh-keygen

5. Для каких задач применяется команда scp?

Для копирования файлов с других компьютеров по ssh-соединению

6. Команды tee и cat. Назначение и применение. Чем cat отличается от more и less?

Команда tee читает из стандартного потока ввода и записывает в стандартный поток вывода и файлы.

tee [OPTION] [FILE]

Команда cat конкатенирует файлы и показывает их содержимое. Если указать один файл в качестве параметра, то просто выведет его содержимое.

more будет отображаться постранично, в отличие от cat.

less также позволяет использовать PageUp, PageDown для перемещения по файлу.

7. Основные команды редактора vi.

* esc — обычный режим
* i — режим ввода
* : — командный режим.
* q — закрыть.
* :q! - выйти без сохранения;
* :wq - записать файл и выйти;
* i — перейти в режим ввода с текущей позиции
* a — перейти в режим ввода после курсора
* o — перейти в режим ввода с новой строки под курсором
* x — удалить символ под курсором
* dw — удаляет символы с текущего до конца слова, включая пробел после слова
* diw — удаляет слово под курсором,
* dd — удалить текущую строку (вырезать);
* u — отмена последней команды;

8. Ссылки и символические ссылки. Создание и применение.

Ссылки создаются командой ln. Бывают жёсткие и символические. Жёсткие указывают на адрес файла на жёстком диске, поэтому остаются активными после перемещения или удаления файла. Символические (создаются ln -s) — на его имя, поэтому неактивны после перемещения/удаления/переименования файла, на который ссылаются.

9. Создание и копирование файлов и папок в Linux.

Создание папки - mkdir

Создание пустого файла - touch

Копирование – cp

10.Перемещение и удаление файлов и папок в Linux.

Перемещение - mv

Удаление файлов - rm (-r - рекурсивно, -f - принудительно)

Удаление пустых каталогов - rmdir

11.Команды useradd, adduser, userdel, deluser, groupadd, addgroup, groupdel, delgroup . Назначение и применение.

useradd — создаёт нового пользователя,

adduser — аналогично + добавляет и настраивает домашнюю папку.  
userdel — удаляет пользователя,

deluser — аналогично + можно настроить конфигурационный файл с параметрами удаления.

groupadd — добавляет новую группу,

addgroup — аналогично, более высокоуровневая.

groupdel — удаляет группу,

delgroup — аналогично, более высокоуровневая.

12.Команды chmod и chown. Назначение и применение.

chmod — изменяет режим доступа к файлу.

chown — задает владельца файла.

13.Какие права доступа Вы имеете к своему домашнему каталогу, каталогам /home и / ?

drwxr-xr-x к /home

drwxr-xr-x к /

drwxr-xr-x к домашнему каталогу

14.Как вы считаете, почему в Linux разделены команды для сжатия и команды архивирования? Приведите их примеры.

tar — архивирование, gzip — сжатие.

Архивирование - это объединение нескольких небольших файлов в один, с целью более удобной последующей передачи, хранения, шифрования или сжатия.

Сжатие — уменьшение размера файла. Архивировать можно не только сжатые файлы, поэтому в линуксе эти понятия разделяют.

15.Команда ps. Назначение и применение. Ключи команды.

ps — возвращает список процессов.

-a — все процессы, кроме процессов, не связанных с терминалом и текущей сессией.

-e — все процессы

-x — в BSD-стиле.

16.Какой алгоритм сжатия использует gzip?

Deflate

17.В каком году был создан архиватор xz? Для каких задач применяется? Какой алгоритм сжатия использует?

2009,

позволяет автоматически сжимать и распаковывать архивы tar,

LZMA2

18.Какой архиватор, созданный в 2015 году, поддерживает поддерживает 22 уровня сжатия.

FreeArc

19.Команда pidof. Назначение и применение. Ключи команды.

pidof определяет id процесса по названию программы.

-s — только один pid

-o — пропустить какие-то pid при выводе

20.Команда fuser. Назначение и применение. Ключи команды.

fuser — выводит процессы, которые используют файлы или сокеты.

-a — все

-k — убивает все процессы, которые нашёл.

-i — запрашивает разрешение пользователя на убийство процесса.