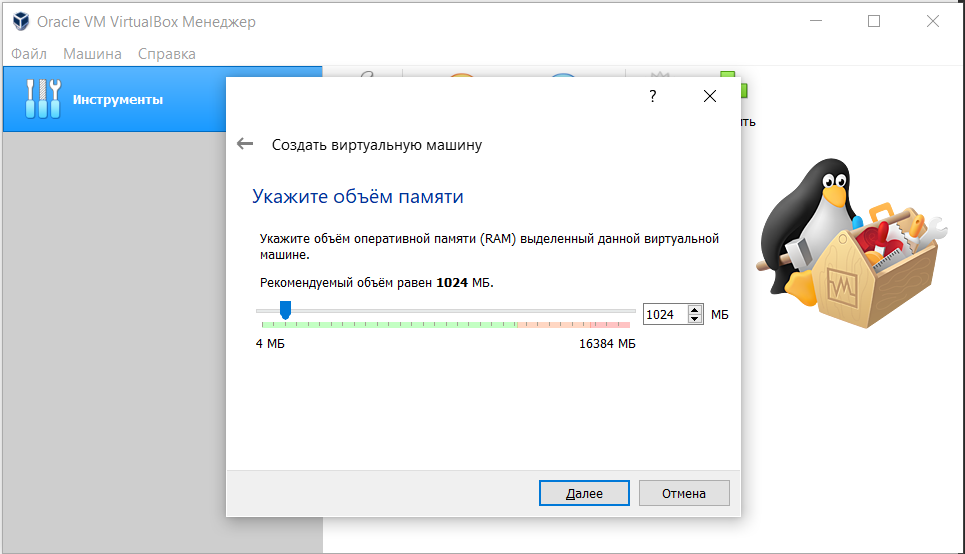
Лабораторная работа №1 Доскоч Роман. Вариант 8.

**Цель работы.**

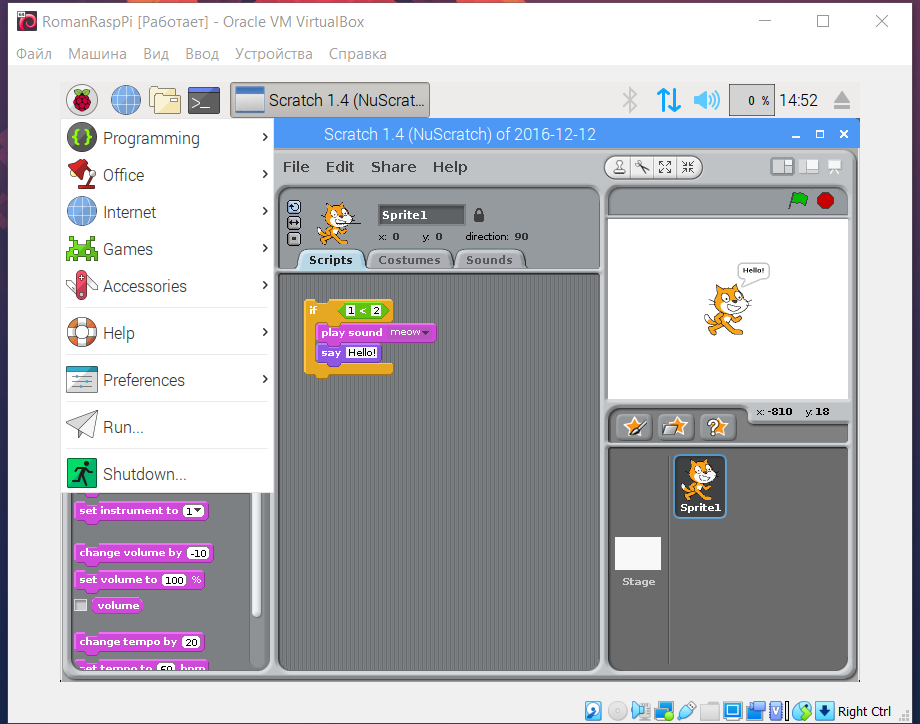
Ознакомление с операционной системой ОС Unix/Linux и основными командами работы с файлами, и каталогами, правами доступа и процессами ОС Unix/Linux.

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

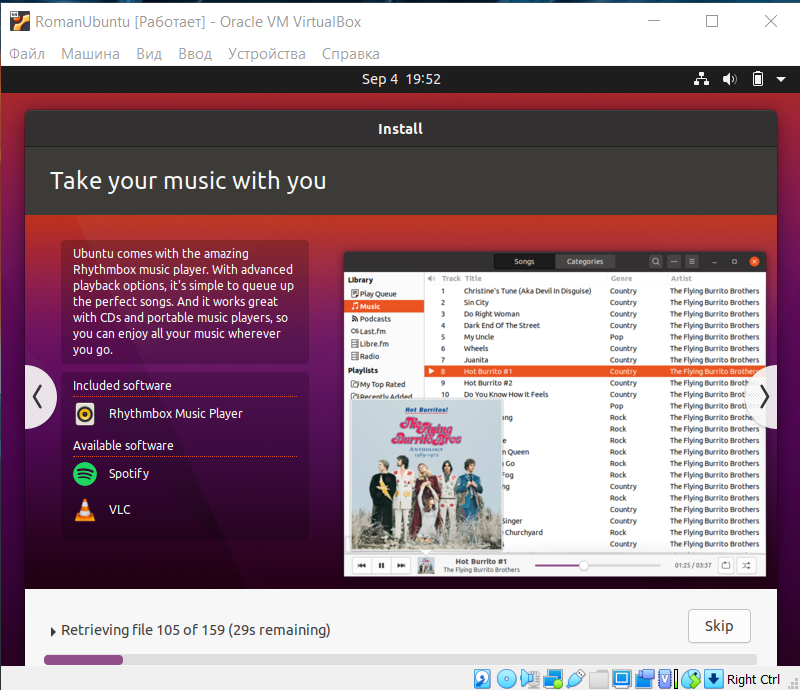
Начал установку Rasbian Pi



Закончил установку



Процесс установки Ubuntu.



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

2.2. Настроить ssh-соединение в гостевой виртуальной ОС Raspberry PI и Ubuntu.

**Установка**

sudo apt install openssh-server

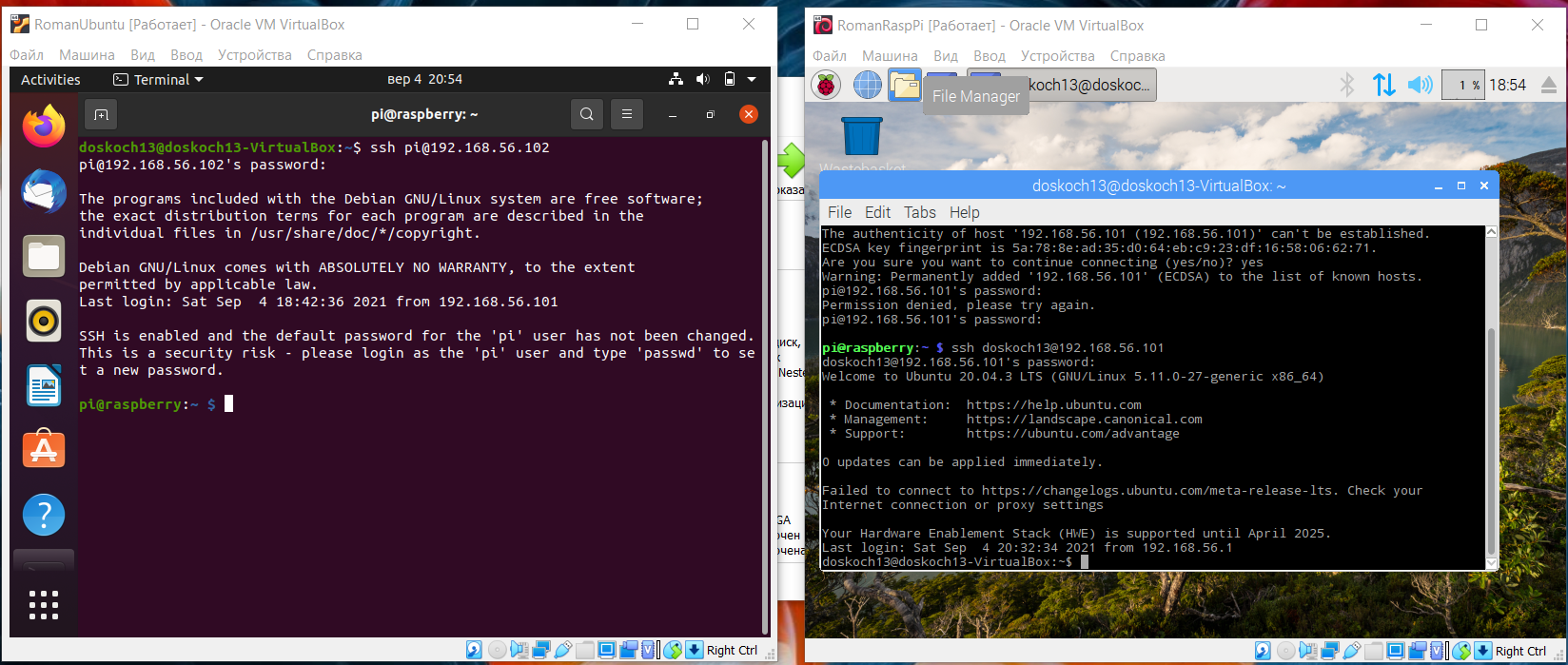
sudo systemctl enable ssh

**Проверка**

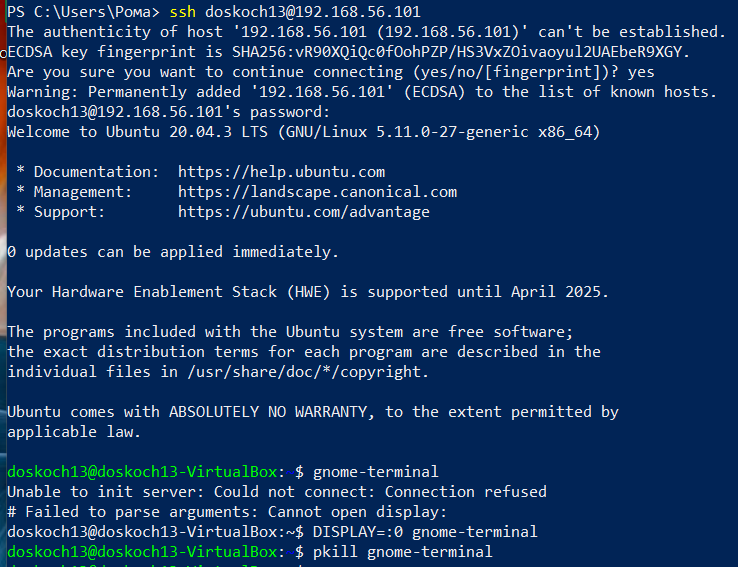
sudo systemctl status | start | restart | stop ssh для SYSTEMD

sudo server ssh status для SYSVINIT

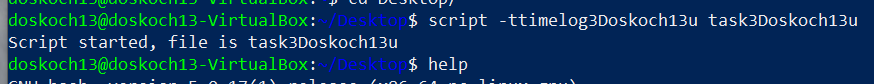
2.3. Подключиться по ssh из ОС Raspberry PI в Ubuntu.

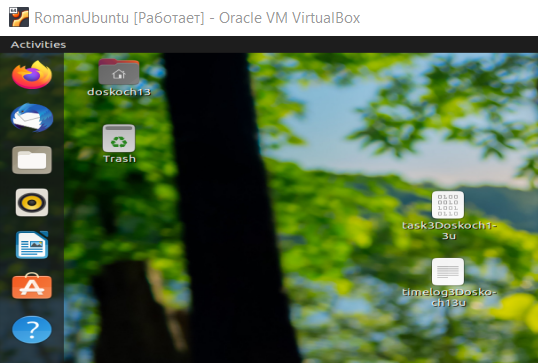


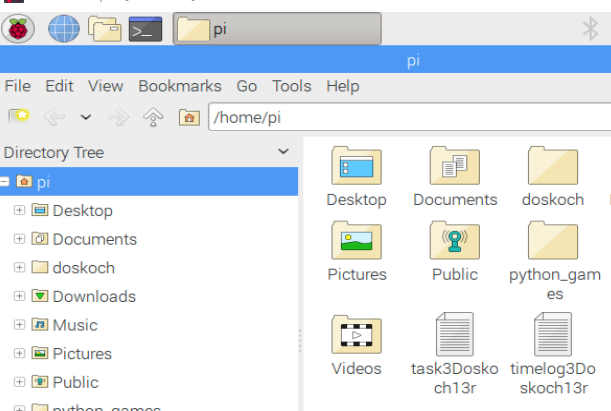
2.5. Открыть ssh-клиент в Windows 10. Подключитесь из хостовой машины в гостевые виртуальные машины под управлением ОС Raspberry PI и Ubuntu. Процесс подключения продемонстрировать скриншотами в отчете



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

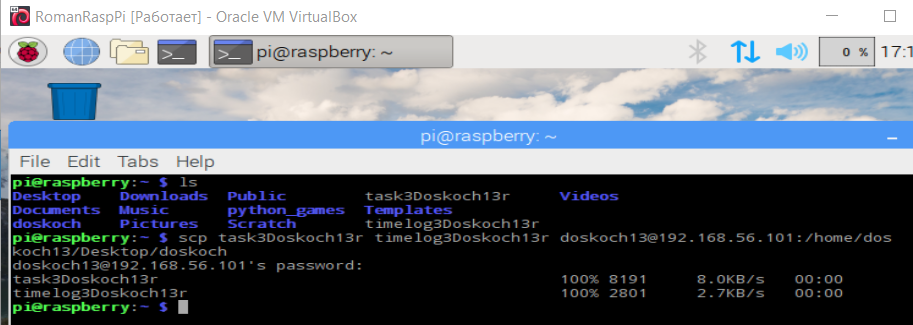
****

****

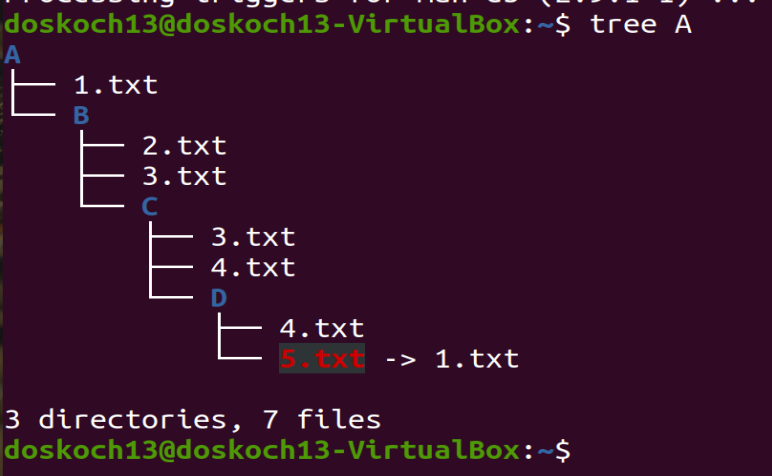
****

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

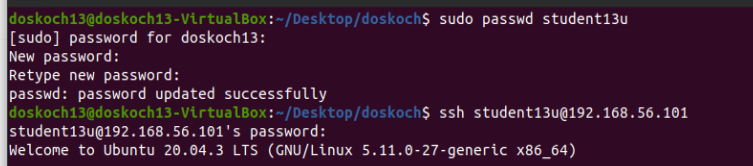
Скопируйте с помощью scp из Raspberry PI сохранённые протоколы работы и добавить в git-репозиторий, созданный в ОС Ubuntu в п.4.5.

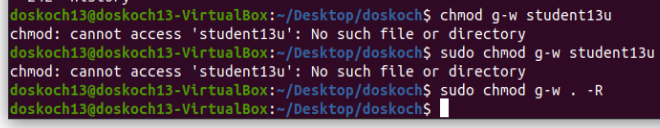
****

**Задание 5. Управление файлами и папками**

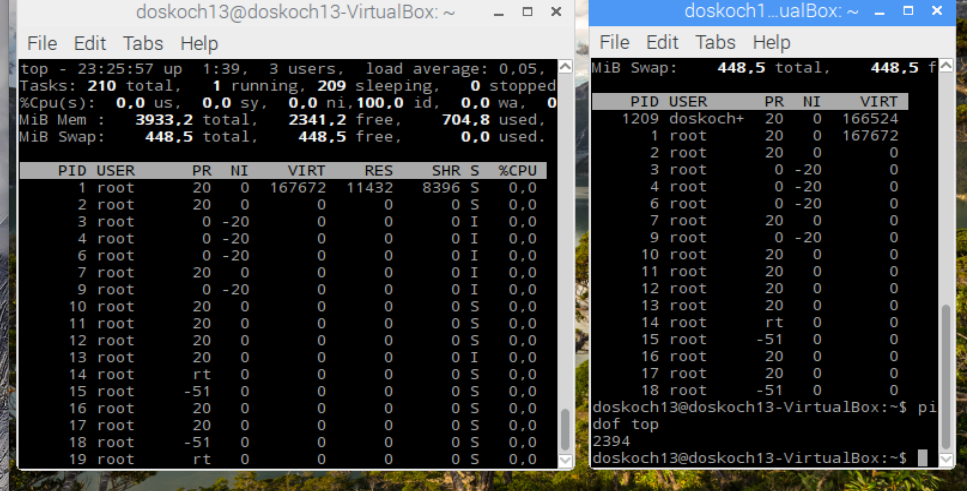
****

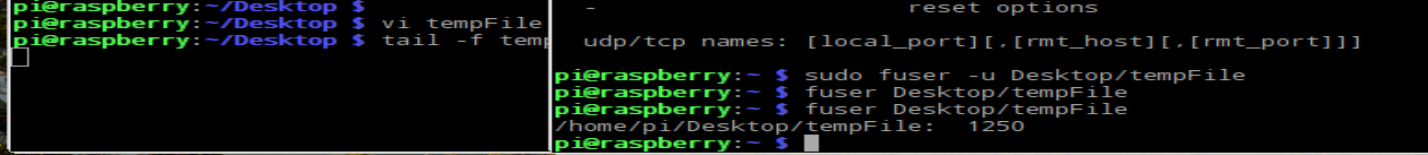
**Задание 6. Управление пользователями и права доступа**

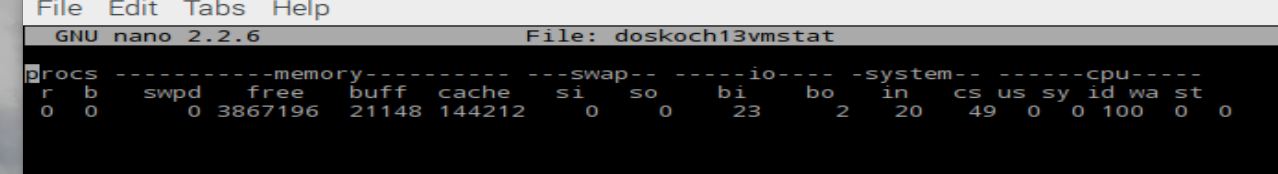
****

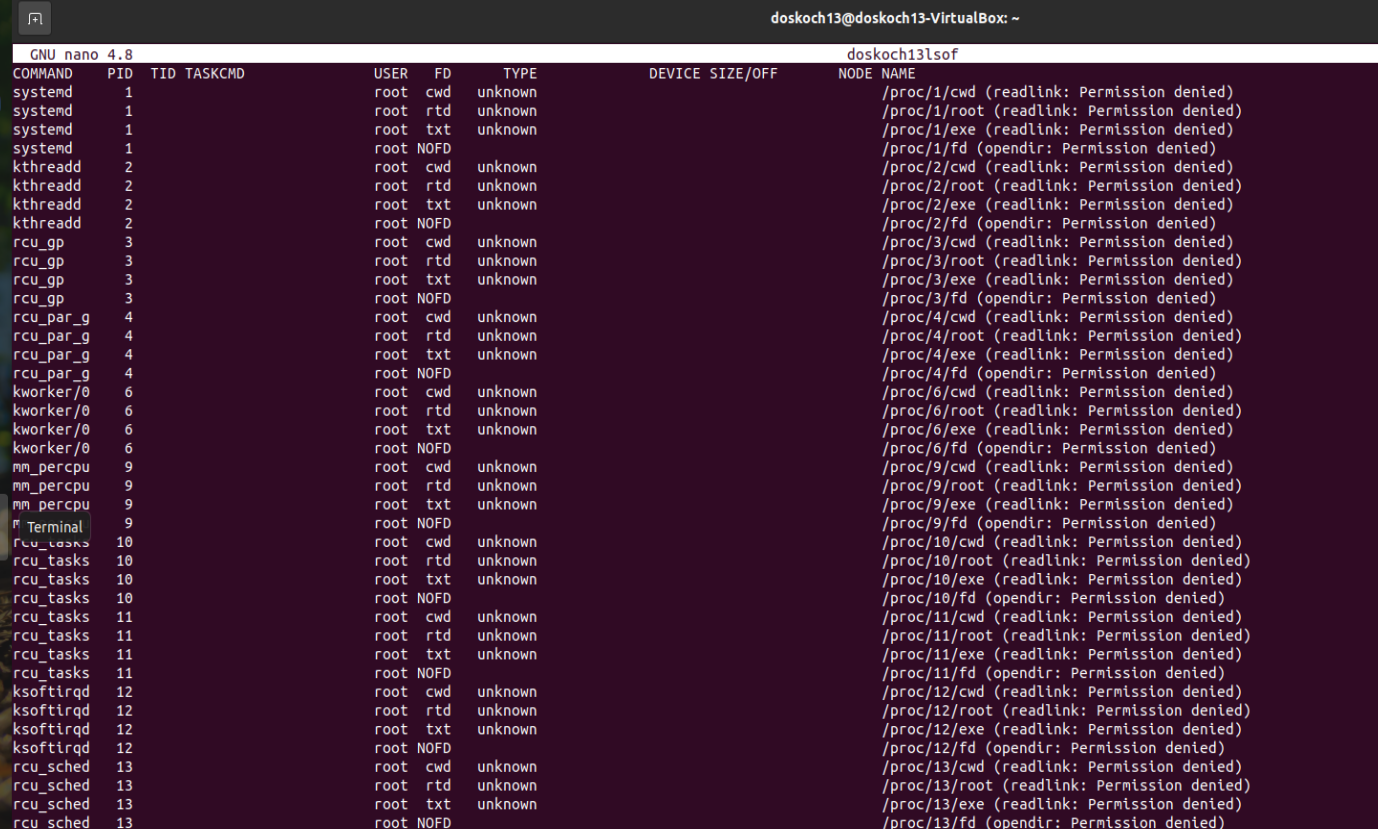
****

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

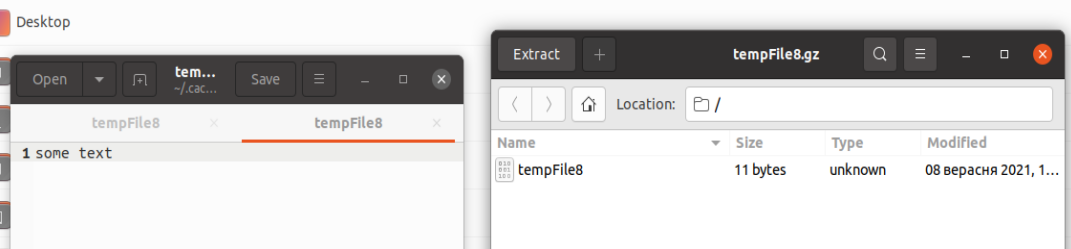
****

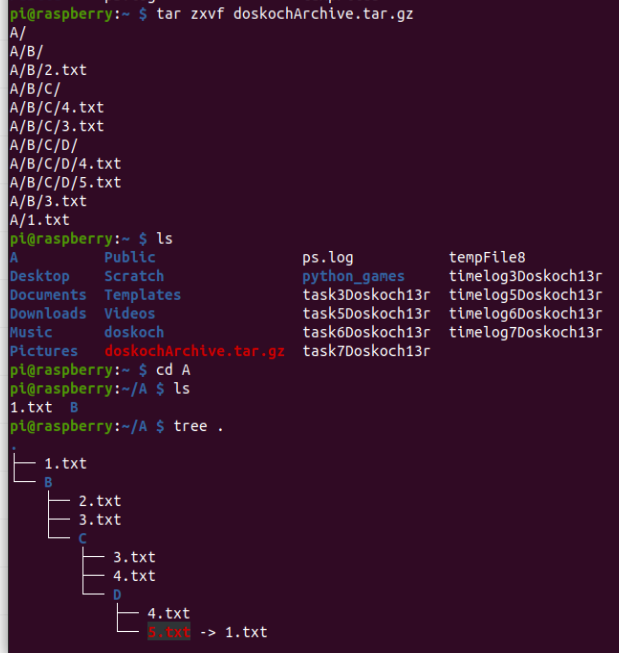
****

****

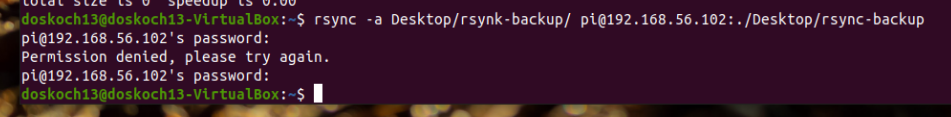
****

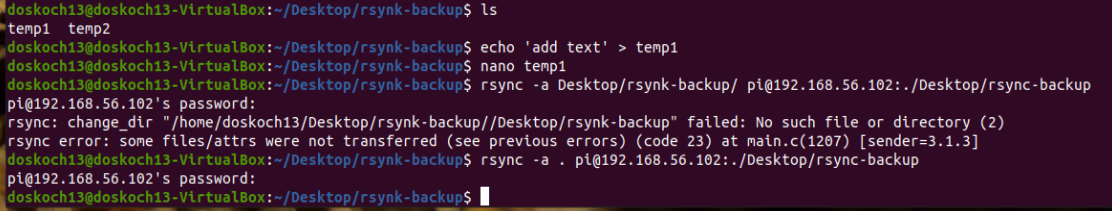
**Задание 8. Сжатие и архивация файлов**

****

****

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

****

****

**Контрольные вопросы.**

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

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

Info = гиппер ссылки.

Man = страницы.

**3**. Команда script — назначение и применение.

Запись всех действий терминала в файл. Так же можно воспроизвести в реальном времени «запись работы терминала»

**4**. Какая команда используется для генерации ssh-ключа?

*ssh-keygen -t rsa*

**5**. Для каких задач применяется команда scp?

Для передачи файлов между серверами по ssh.

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

Команда *tee* считывает стандартный ввод, после чего записывает его в стандартный вывод и одновременно копирует его в подготовленный файл или переменную

Команда *cat* позволяет вывести на стандартный вывод содержимое любого файла.

Команда *more* позволяет выводить содержимое больших файлов постранично.

Команда *less* работает аналогично, но позволяет двигаться по файлу в обоих направлениях.

**7**. Основные команды редактора vi.

:q - выход из программы

:!command - выполнение команды (например, git)

:p - вставить

:w [filename] - сохранить

**8.** Ссылки и символические ссылки. Создание и применение.

Синтаксис: *ln файл1 [файл2 ...] целевой\_файл*

Команда ln делает целевой\_файл ссылкой на файл1.

-s создание символической ссылки (по умолчанию создается жесткая ссылка)

**9**. Создание и копирование файлов и папок в Linux.

Nano file\_name

*touch [option] file\_name(s)*

*mkdir [options...] [directories ...]*

*cp [option]... source... directory*

**10**.Перемещение и удаление файлов и папок в Linux.

*mv [filename] [new\_filename]*

*rm [option]... FILE...*

*rmdir [option]... DIRECTORY...*

**11**.Команды useradd, adduser, userdel, deluser, groupadd, addgroup,

groupdel, delgroup . Назначение и применение.

*useradd [options] username* - добавляет нового пользователя в систему

*userdel [options] login -* удаляет пользователя и принадлежащие ему файлы

*groupadd [options] group -* добавляет группу пользователей

*groupdel group -* удаляет группу

*delgroup [options] group* - удаляет пользователя

**12**.Команды chmod и chown. Назначение и применение.

*chmod* - изменение режима доступа к файлам.

Режим в целом описывается в терминах трех последовательностей, по три буквы в каждой:

Владелец Группа Прочие

(u) (g) (o)

rwx rwx rwx

+ добавить команду

*r - Право на чтение.*

*w - Право на запись.*

*x - Право на выполнение (поиск в каталоге).*

*chown* — изменение владельца и группы файлов.

Команда *chown* изменяет владельца и/или группу для каждого заданного

файла. В качестве имени владельца/группы берется первый аргумент, не

являющийся опцией. Если задано только имя пользователя (или числовой

идентификатор пользователя), то данный пользователь становится владельцем каждого из указанных файлов, а группа этих файлов не изменяется. Если за именем пользователя через двоеточие следует имя группы (или числовой идентификатор группы), без пробелов между ними, то изменяется также и группа файла.

**13**.Какие права доступа Вы имеете к своему домашнему каталогу, каталогам

/home и / ?

Полные права.

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

tar -cvf filename.tar files/directories

gzip file.txt

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

*ps* - вывод информации о состоянии процессов.

Команда ps выводит в стандартный вывод информацию о текущем

состоянии процессов.

Опции:

-a все процессы, кроме лидеров групп и процессов, не ассоциированных с

терминалом.

-d все процессы, кроме лидеров групп.

-e все процессы.

-g список выбирать процессы по списку лидеров групп.

-p список выбирать процессы по списку идентификаторов процессов.

-t список выбирать процессы по списку терминалов.

-u список выбирать процессы по списку идентификаторов пользователей.

-f генерировать полный листинг.

-l генерировать листинг в длинном формате.

**16.**Команда pidof. Назначение и применение. Ключи команды.

*pidof [options] command* - выводит pid процесса

**17**.Команда fuser. Назначение и применение. Ключи команды.

 *fuser [options] [file|socket] -* ищет процесс по файлам или сокетам

Команда **fuser** предназначена для задействования одноименной утилиты, выводящей список процессов, работающих с заданным файлом, сетевым портом или файловой системой и, в случае необходимости, позволяющая автоматически завершать их работу.