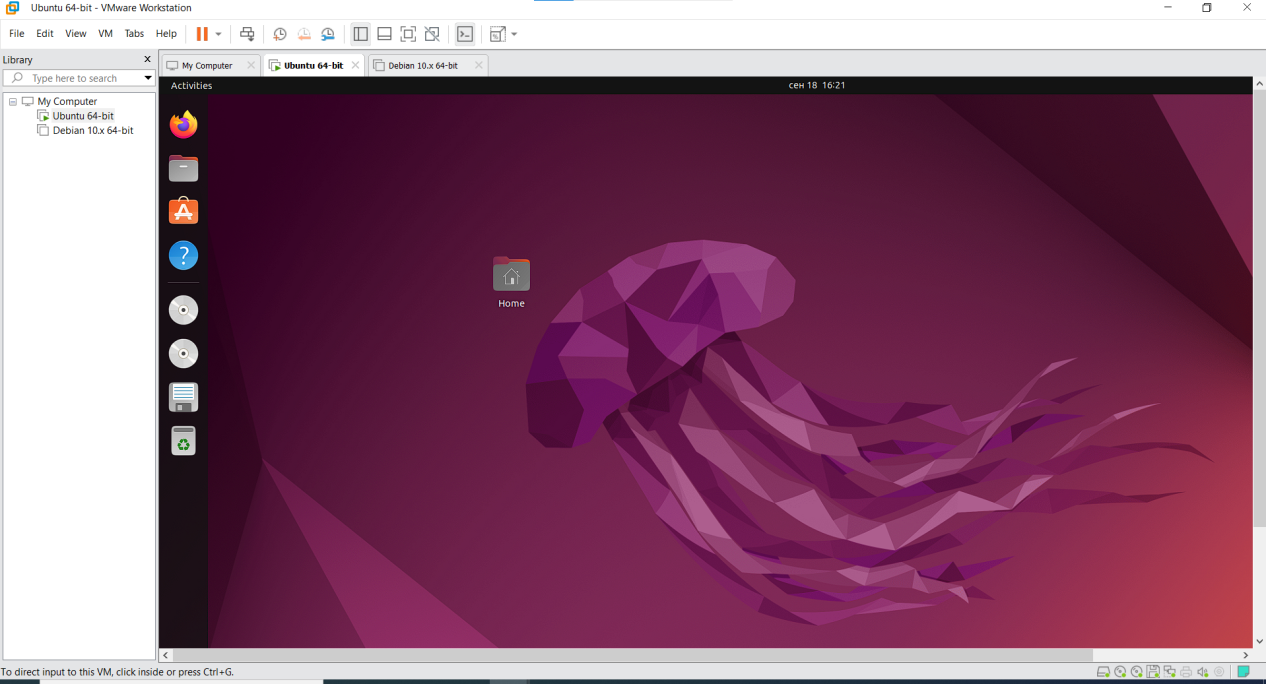
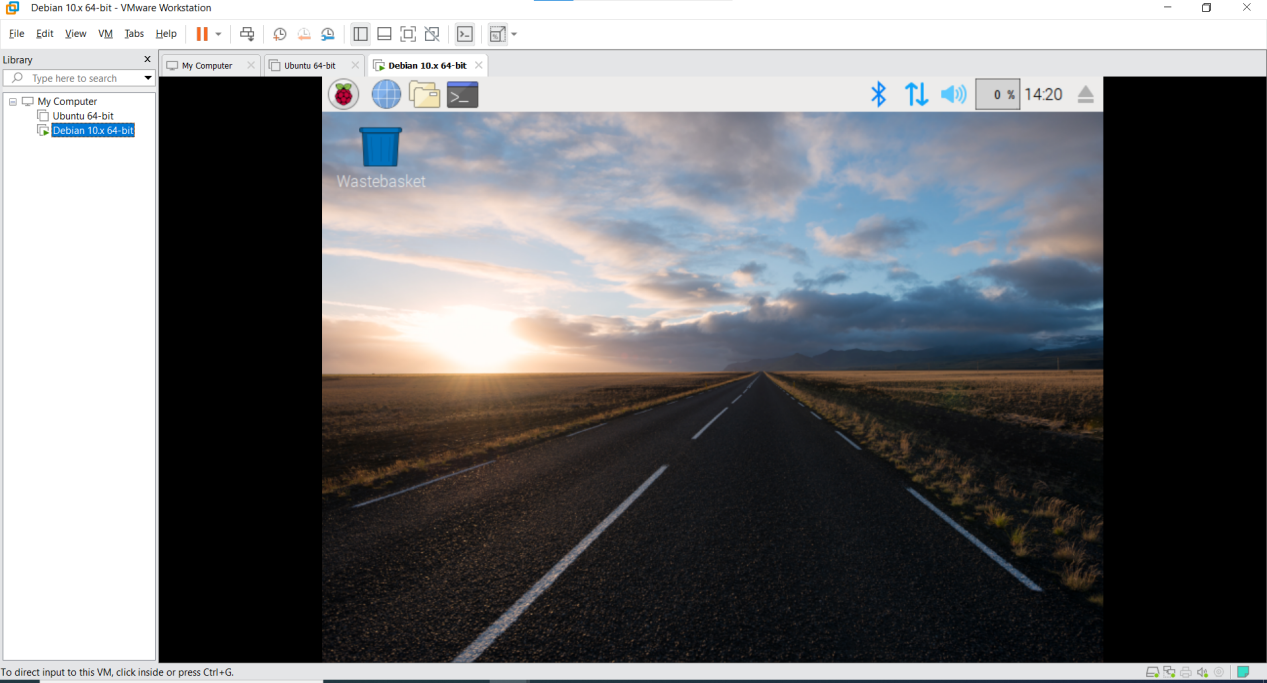
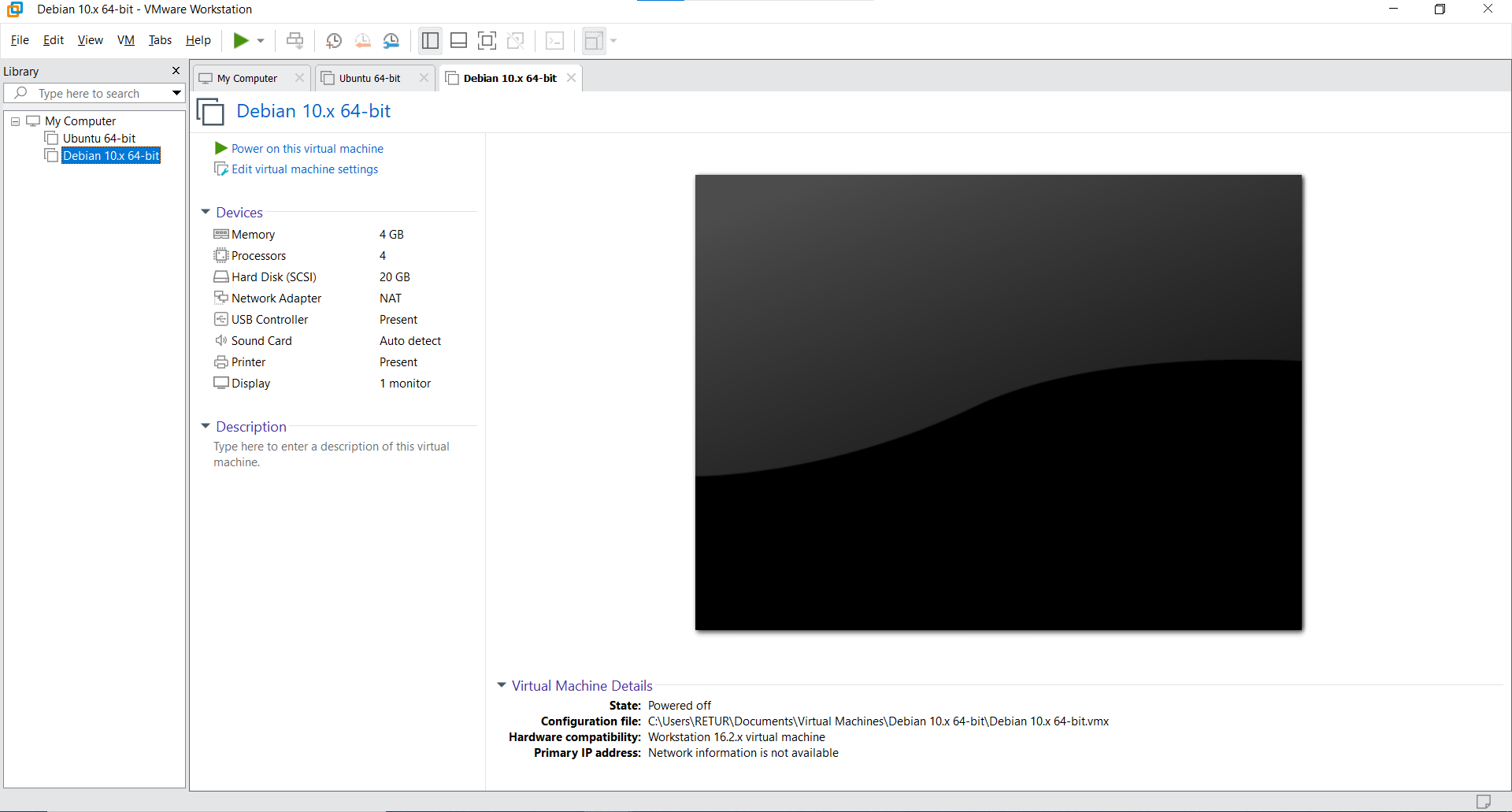
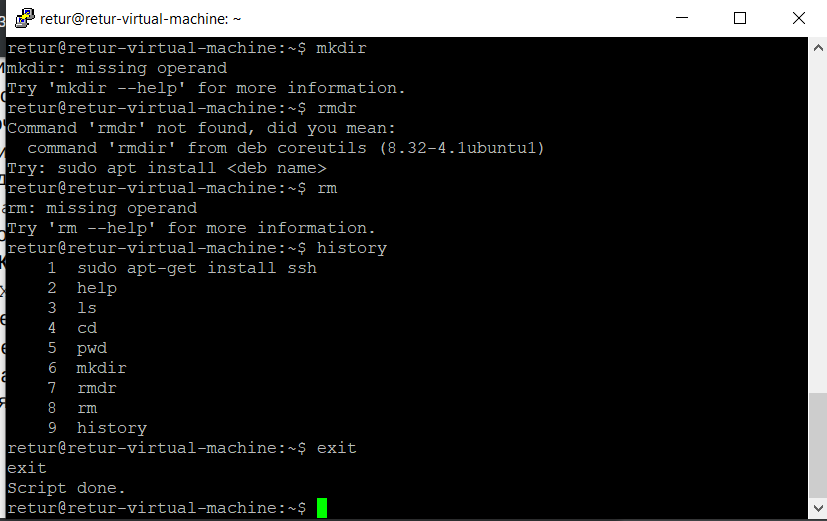
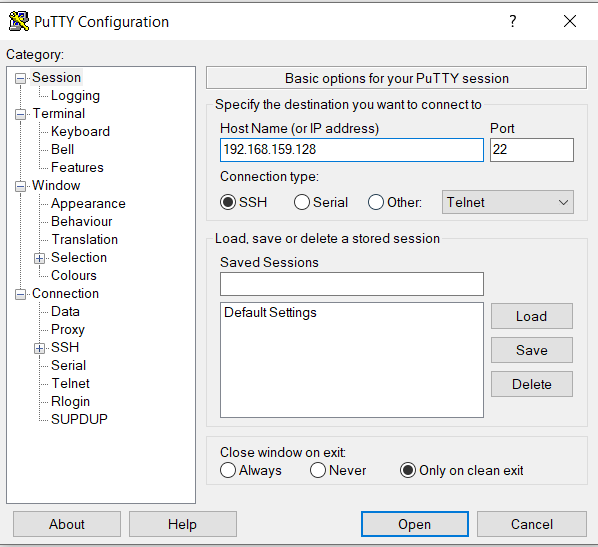
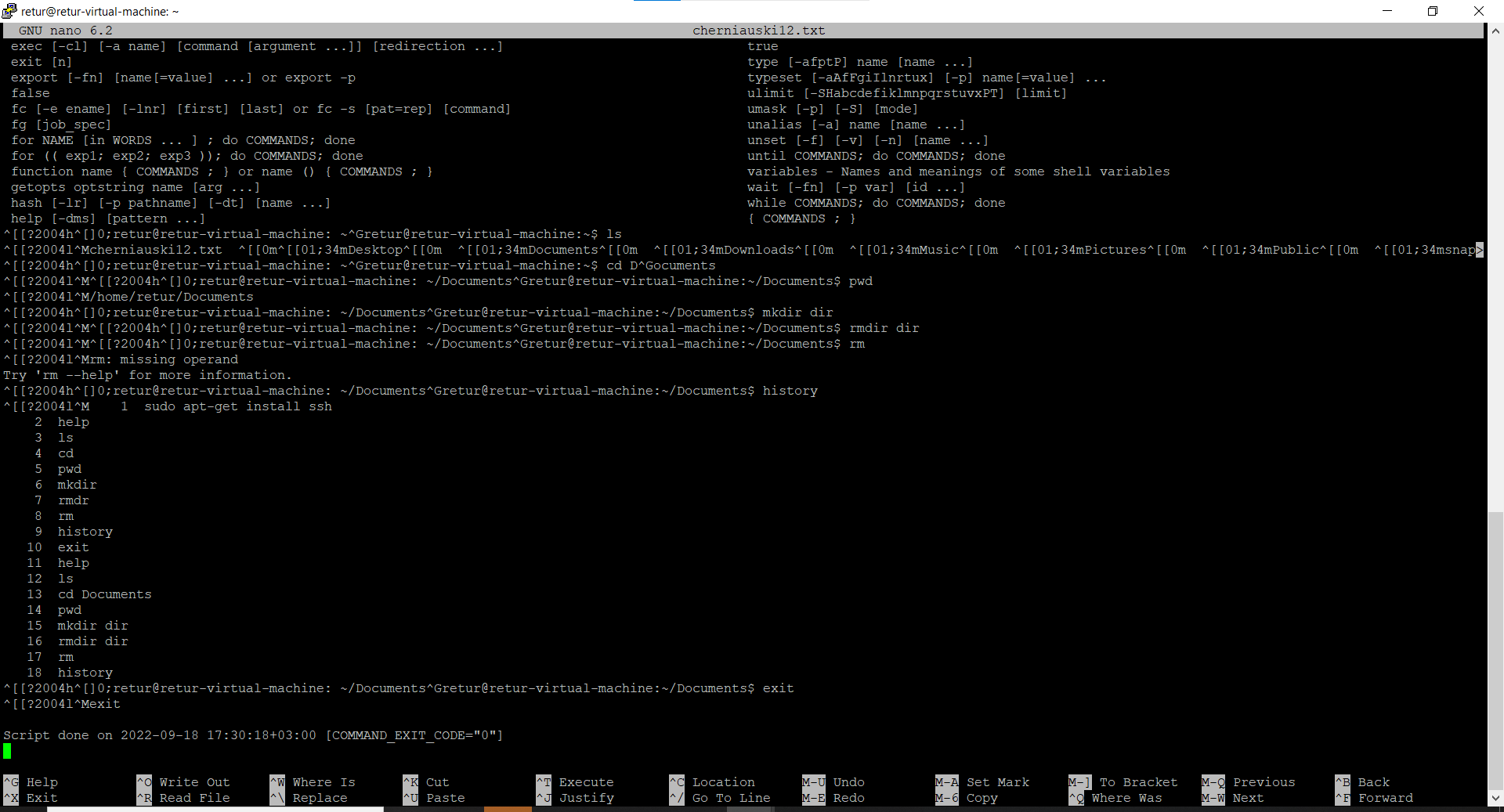
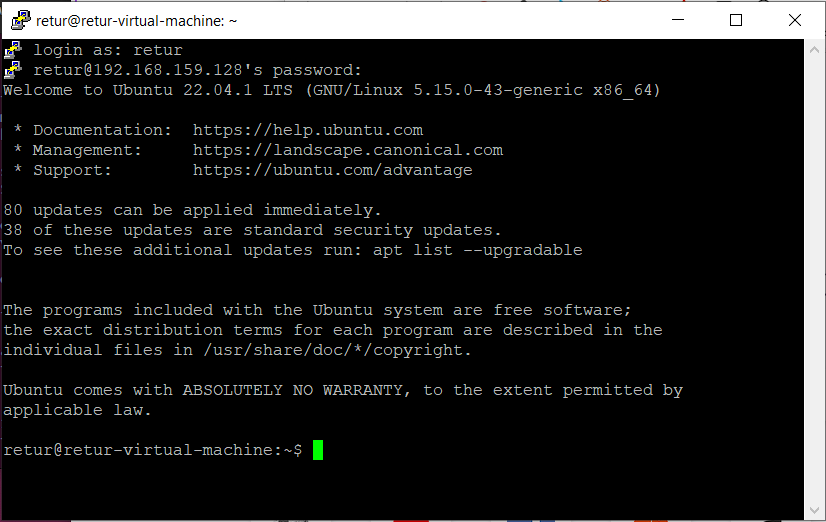
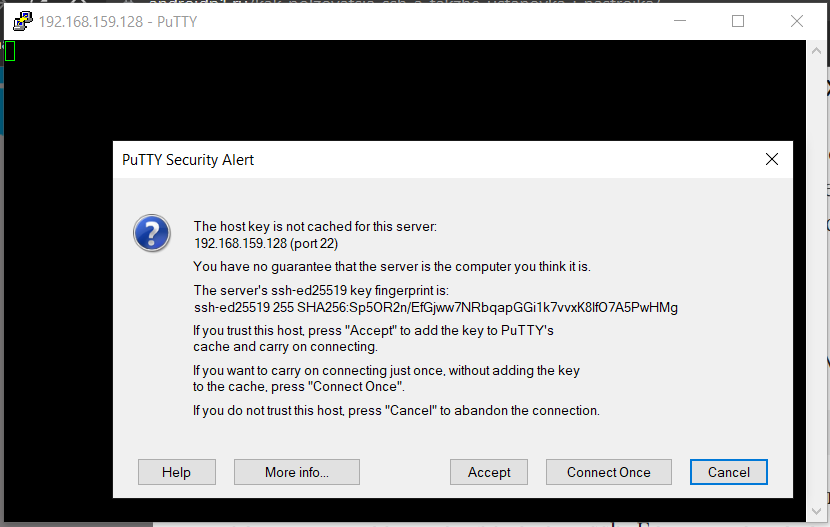
Установка

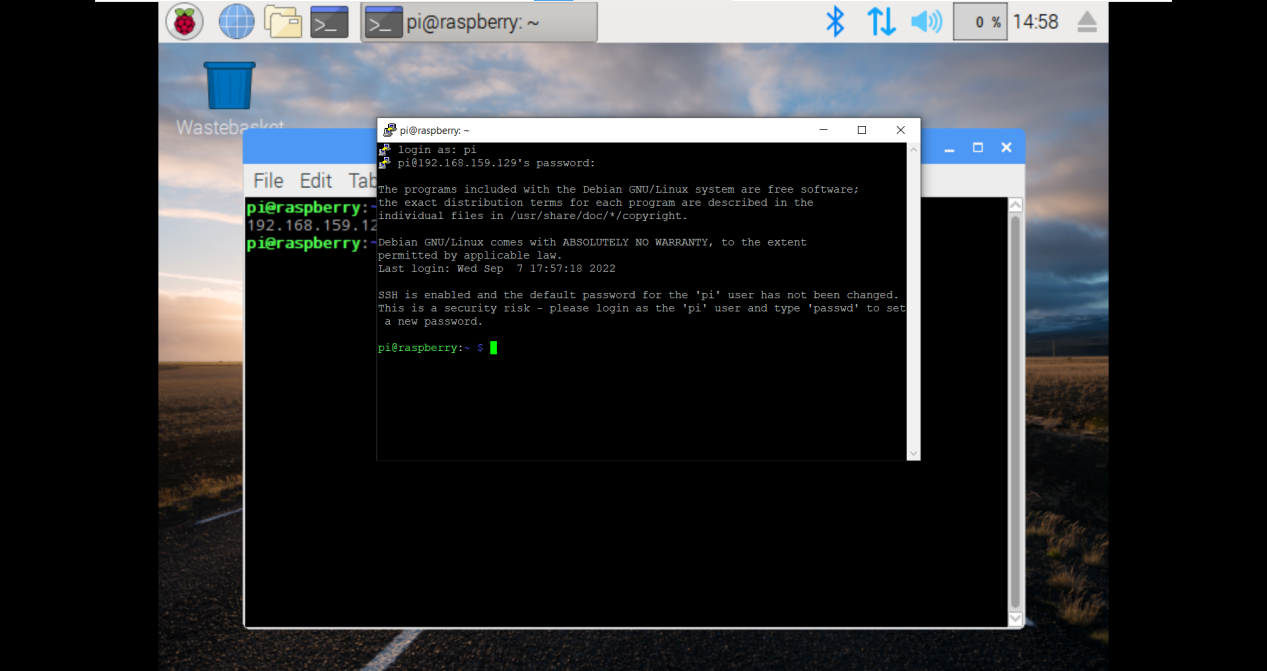


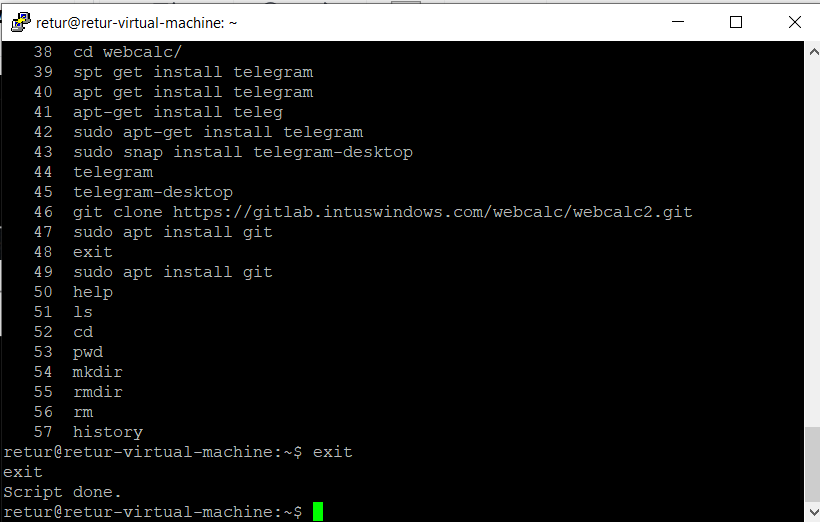
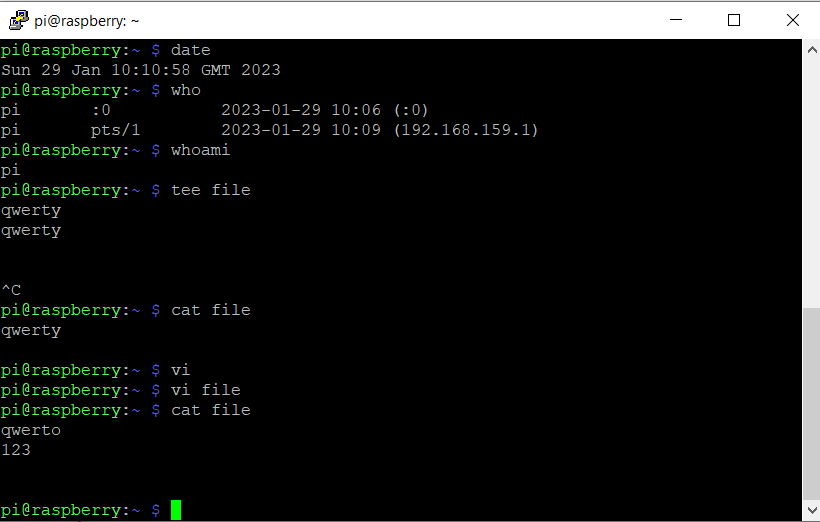
SSH

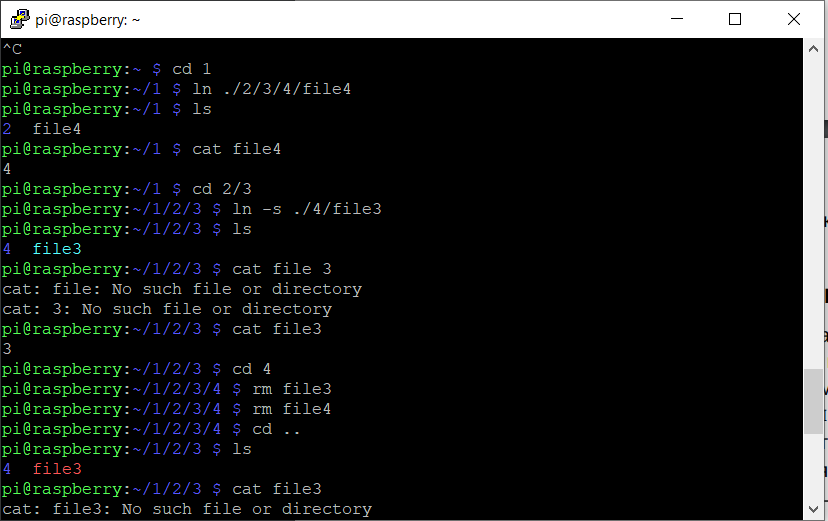






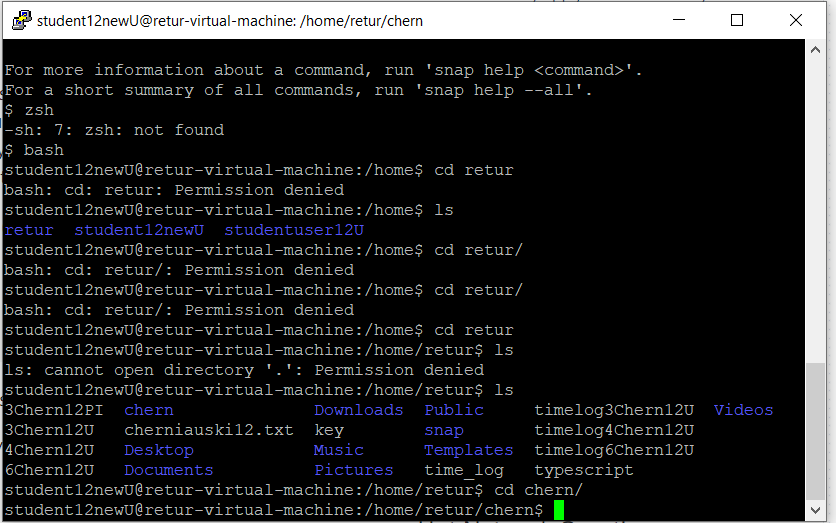


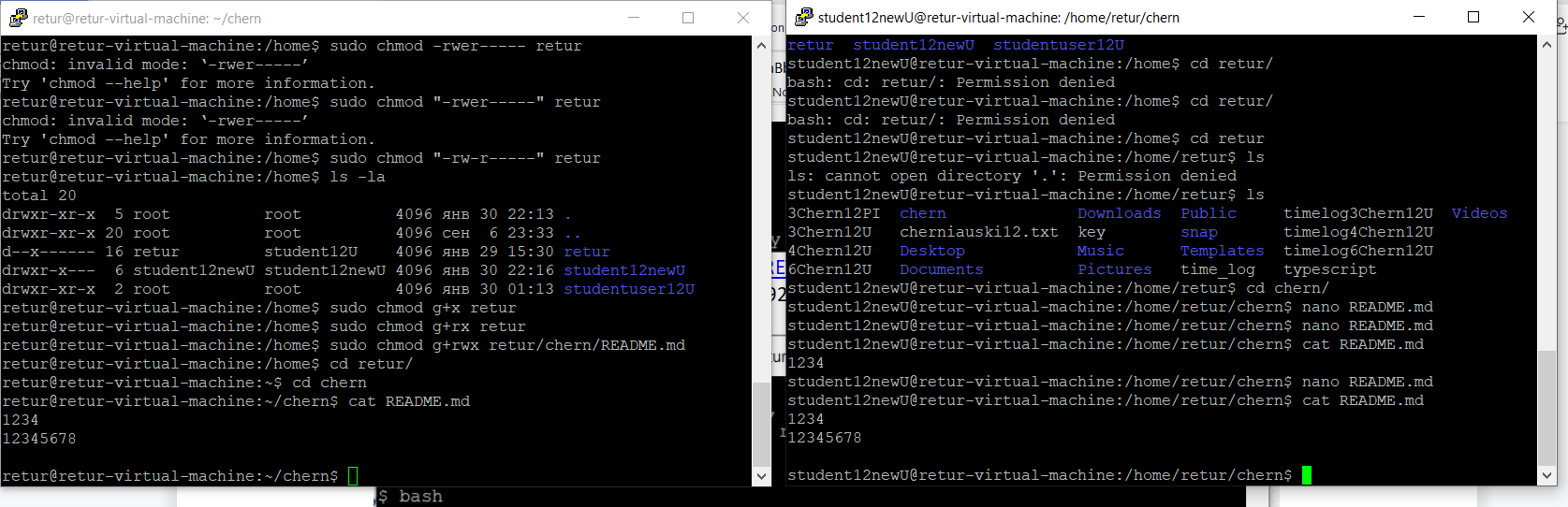




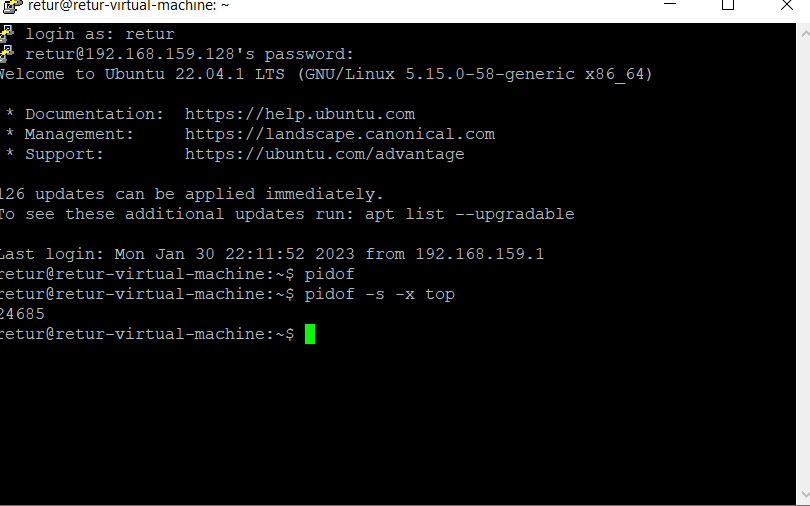
Задание 3, 4, 5 на гите <https://github.com/RETUR07/pvms_lab1/>

scp 3Chern12PItimelog 3Chern12PI retur@192.168.159.128:~/

Задание 6 



Задание 7 <https://github.com/RETUR07/pvms_lab1/>



Контрольные вопросы

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

**-a** - отображать все файлы, включая скрытые, это те, перед именем которых стоит точка;

**-A** - не отображать ссылку на текущую папку и корневую папку .

**--author** - выводить создателя файла в режиме подробного списка;

**-b** - выводить Escape последовательности вместо непечатаемых символов;

**--block-size** - выводить размер каталога или файла в определенной единице измерения, например, мегабайтах, гигабайтах или килобайтах;

**-B** - не выводить резервные копии, их имена начинаются с ~;

**-c** - сортировать файлы по времени модификации или создания, сначала будут выведены новые файлы;

**-C** - выводить колонками;

**--color** - включить цветной режим вывода, автоматически активирована во многих дистрибутивах;

**-d** - выводить только директории, без их содержимого, полезно при рекурсивном выводе;

**-D** - использовать режим вывода, совместимый с Emacs;

**-f** - не сортировать;

**-F** - показывать тип объекта, к каждому объекту будет добавлен один из специализированных символов \*/=>@|;

**--full-time** - показывать подробную информацию, плюс вся информация о времени в формате ISO;

**-g** - показывать подробную информацию, но кроме владельца файла;

**--group-directories-first** - сначала отображать директории, а уже потом файлы;

**-G** - не выводить имена групп;

**-h** - выводить размеры папок в удобном для чтения формате;

**-H** - открывать символические ссылки при рекурсивном использовании;

**--hide** - не отображать файлы, которые начинаются с указанного символа;

**-i** - отображать номер индекса inode, в которой хранится этот файл;

**-l** - выводить подробный список, в котором будет отображаться владелец, группа, дата создания, размер и другие параметры;

**-L** - для символических ссылок отображать информацию о файле, на который они ссылаются;

**-m** - разделять элементы списка запятой;

**-n** - выводить UID и GID вместо имени и группы пользователя;

**-N** - выводить имена как есть, не обрабатывать контролирующие последовательности;

**-Q** - брать имена папок и файлов в кавычки;

**-r** - обратный порядок сортировки;

**-R** - рекурсивно отображать содержимое поддиректорий;

**-s** - выводить размер файла в блоках;

**-S** - сортировать по размеру, сначала большие;

**-t** - сортировать по времени последней модификации;

**-u** - сортировать по времени последнего доступа;

**-U** - не сортировать;

**-X** - сортировать по алфавиту;

**-Z** - отображать информацию о расширениях SELinux;

**-1** - отображать один файл на одну строку

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

Команда *info* является альтернативой команде *man*. Для получения информации по отдельной команде надо задать в командной строке info с параметром, являющимся именем интересующей вас команды, например, *info man*

Информация, которую вы увидите, в большинстве случаев немного отличается от той, которую дает команда man.

Выдаваемая info информация представлена в гипертекстовом формате. В силу этого вы получаете возможность просматривать различные разделы помощи, не выходя из оболочки, предоставляемой командой info.

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

С ее помощью можно записывать все, что происходит в терминале в файл. Фактически вы составляете полный лог всех команд, которые вы вводите в терминале и то, что выводится на экран.

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

*ssh-keygen -t rsa*

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

Утилита для передачи файлов через ssh.

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

Команда *tee* считывает стандартный ввод (stdin), после чего записывает его в стандартный вывод (stdout) и одновременно копирует его в подготовленный файл или переменную tee [OPTIONS] [FILE]

Команда *cat* позволяет вывести на стандартный вывод (на экран) содержимое любого файла, однако она используется для этих целей очень редко, разве что для вывода очень небольших по объему файлов.

Команда *more* позволяет выводить содержимое больших файлов постранично. Перемещаться по страницам можно только вперёд.

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.

ln — создание ссылки на файл. Синтаксис: *ln [-f] файл1 [файл2 ...] целевой\_файл*

Описание: Команда ln делает целевой\_файл ссылкой на файл1. Файл1 не должен совпадать с целевым\_файлом. Если целевой\_файл является каталогом, то в нем создаются ссылки на файл1, файл2,... с теми же именами. Только в этом случае можно указывать несколько исходных файлов. Если целевой\_файл существует и не является каталогом, его старое содержимое теряется.

Опции:

-f удаление существующего целевого файла

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

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

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

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

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

**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] [--only-if-empty] group* - удаляет пользователя

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

*chmod* - изменение режима доступа к файлам.

Команда *chmod* изменяет права доступа к указанным файлам (среди

которых могут быть каталоги) в соответствии с указанным режимом. Режим может быть задан в абсолютном или символьном виде.

Абсолютный вид — восьмеричное число, являющееся поразрядным ИЛИ

следующих режимов (названы не все режимы):

00400 Доступен для чтения владельцем.

00200 Доступен для записи владельцем.

00100 Доступен для выполнения (в случае каталога — для просмотра)

владельцем.

00040 Доступен для чтения членами группы.

00020 Доступен для записи членами группы.

00010 Доступен для выполнения (просмотра) членами группы.

00004 Доступен для чтения прочими пользователями.

00002 Доступен для записи прочими пользователями.

00001 Доступен для выполнения (просмотра) прочими пользователями.

Символьный вид основан на однобуквенных обозначениях, которые определяют класс доступа и права доступа для членов данного класса. Права доступа к файлу зависят от идентификатора пользователя и идентификатора группы, в которую он входит. Режим в целом описывается в терминах трех последовательностей, по три буквы в каждой:

Владелец Группа Прочие

(u) (g) (o)

rwx rwx rwx

Для задания режима доступа в символьном виде используется синтаксис:

[кому] операция права

Часть кому есть комбинация букв u, g и o (владелец, члены группы и прочие пользователи соответственно). Если часть кому опущена или указано a, то это эквивалентно ugo.

Операция может быть: + (добавить право), — (лишить права), = (в пределах данного класса присвоить права абсолютно, то есть добавить указанные права и отнять неуказанные).

Права — любая осмысленная комбинация следующих букв (не все):

*r - Право на чтение.*

*w - Право на запись.*

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

Для просмотра прав доступа и контроля при их изменении используется команда *ls -l.*

*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 -czf atar1.tar.gz report.txt message.txt temp.txt* - архивация

*tar -czvf atar1.tar.gz report.txt message.txt temp.txt* - сжатие с архивацией

**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* - ищет id процесса

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

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