LAB 2

LINUX BASIC - SHELL SCRIPTING

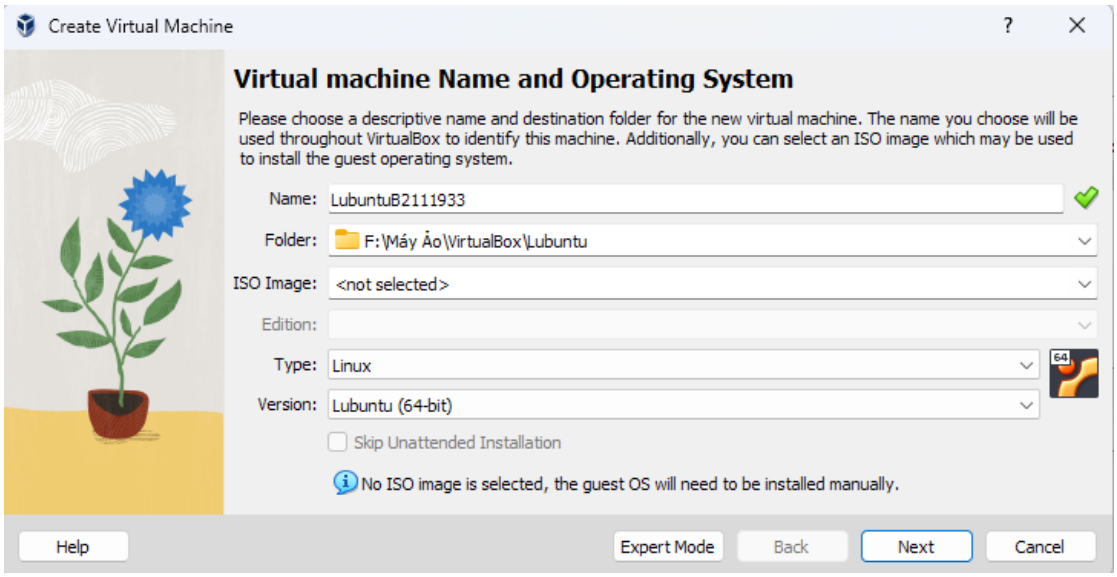
Fullname: Trương Đặng Trúc Lâm

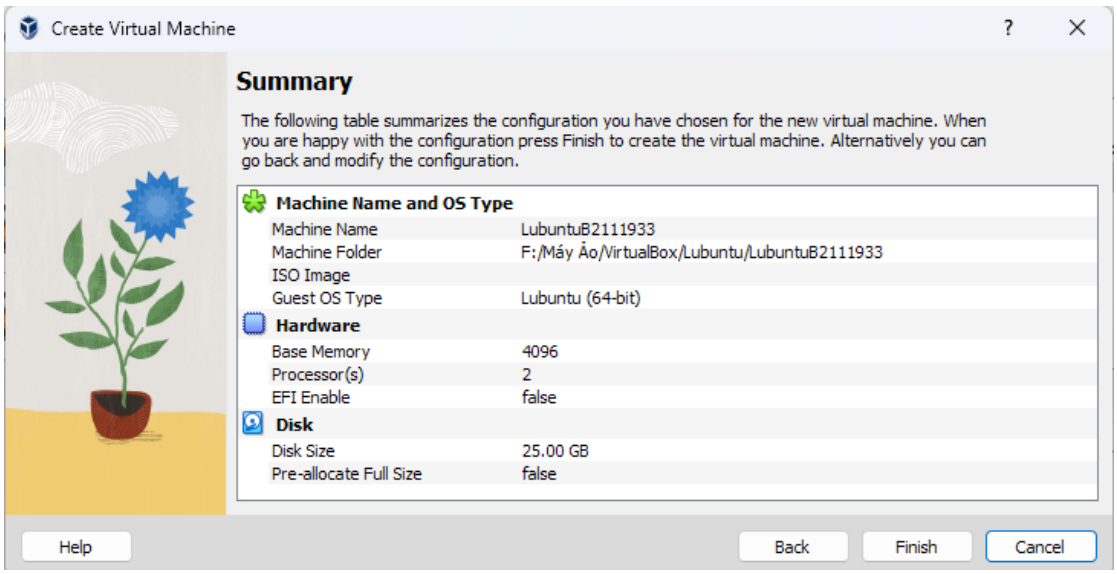
Student ID: B2111933

- Note: screenshots need to be clear and good-looking; submissions must be in PDF format.

1. Lubuntu/Ubuntu installation

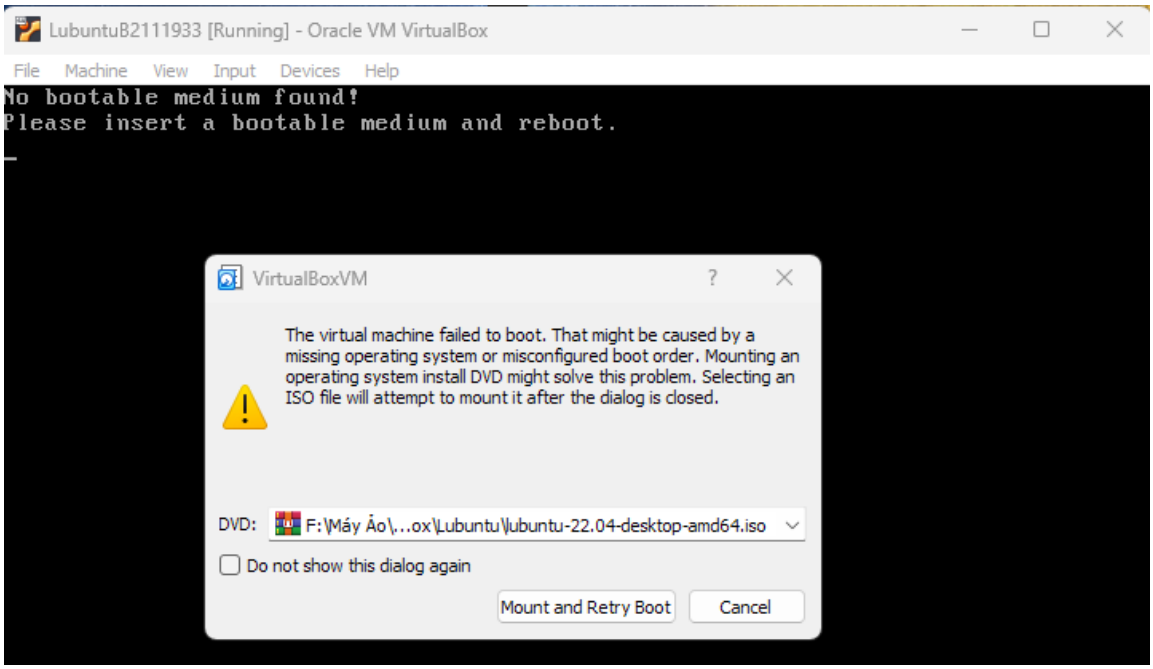
1.1. Create a virtual machine using VirtualBox/VMWare.



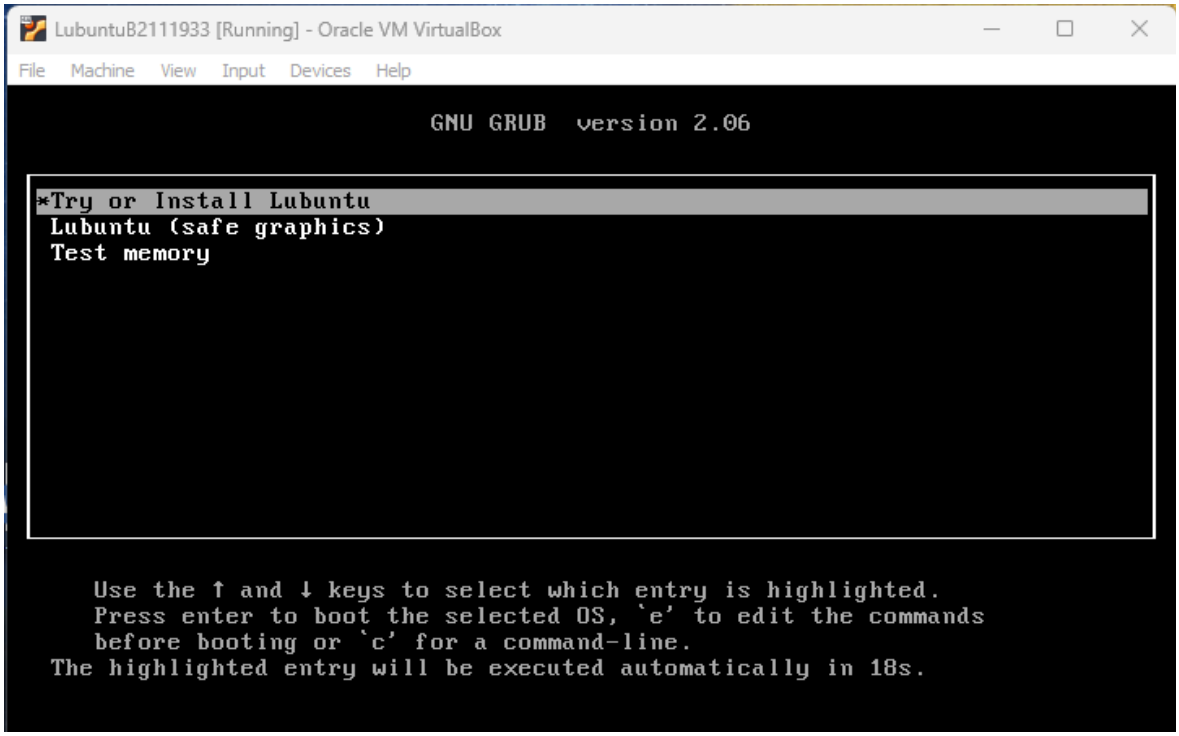


Create a **Virtual Machine**

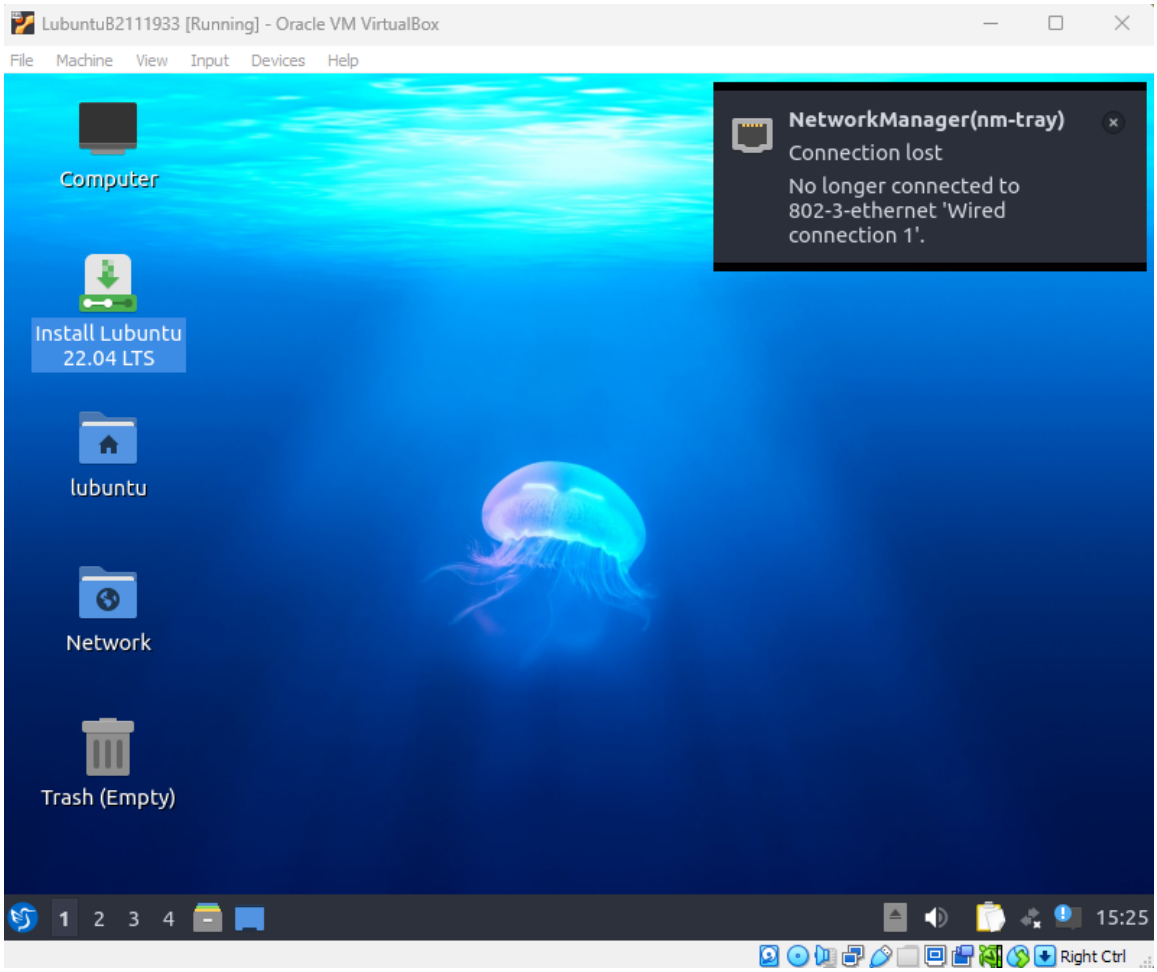
1.2. Install Lubuntu Desktop 22.04 LTS (or any other Linux distributions) as the OS of the VM. (take a screenshot after finishing the installation)
+ Please disable networking before installing the OS.

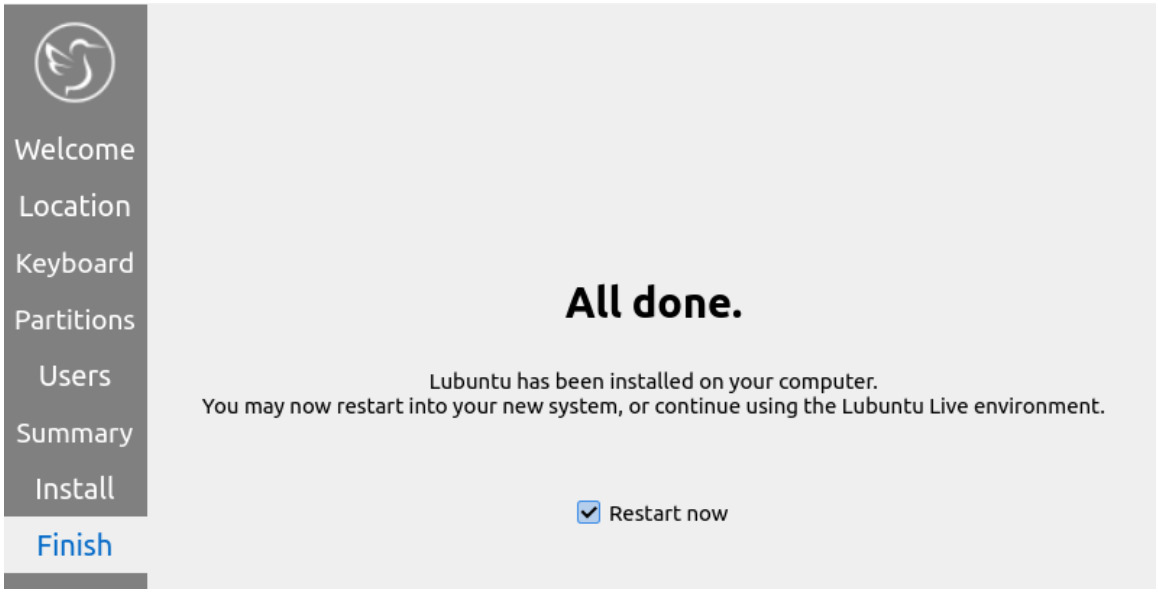


Choose a disk for the **VM**

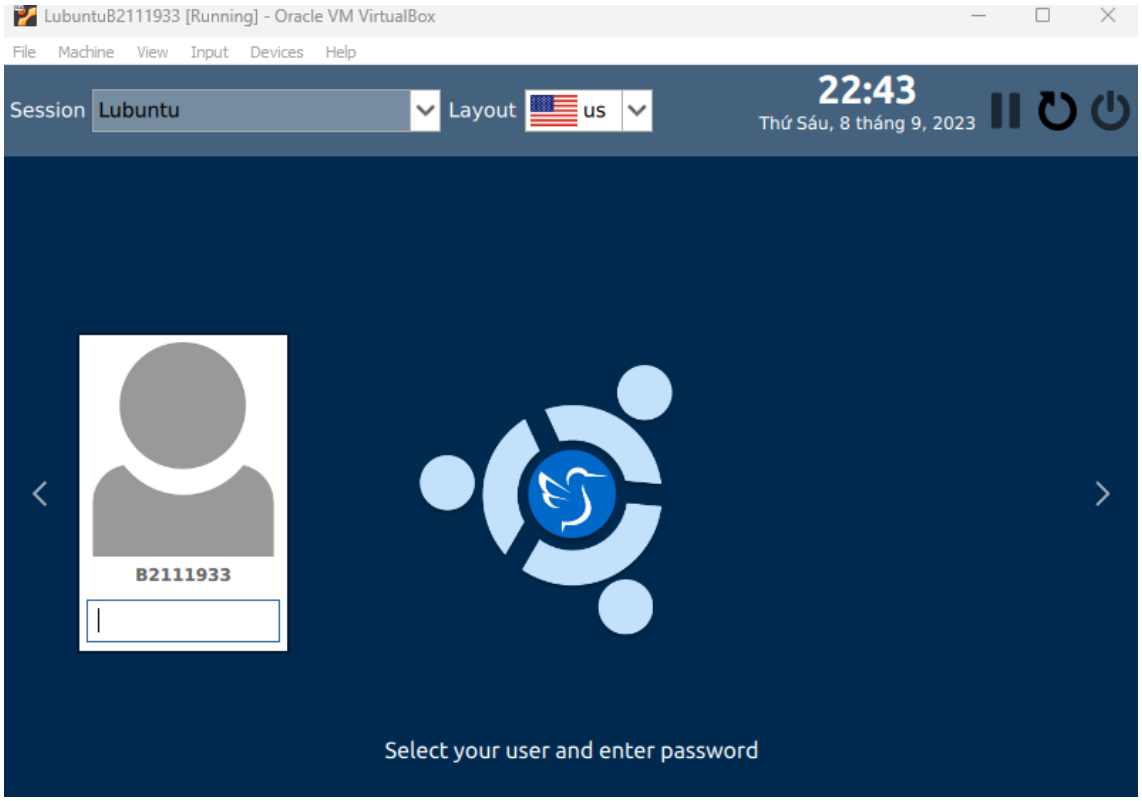


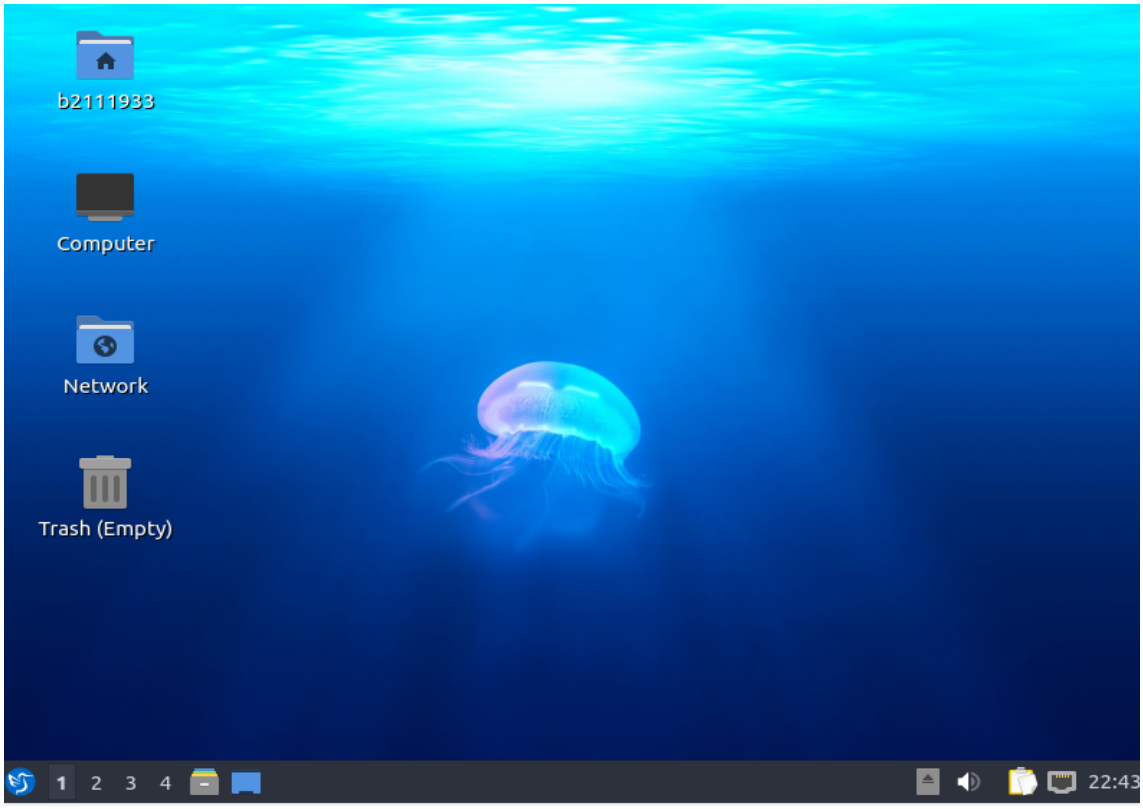
Start the created **VM**





Restart **Lubuntu** to complete the installation





Lubuntu is installed

1.3. Configure network and proxy settings using graphical user interface (if necessary)

1.4. Setting up proxy for command line (if necessary)

```
$sudo featherpad /etc/environment
```

- Appending below lines to the file

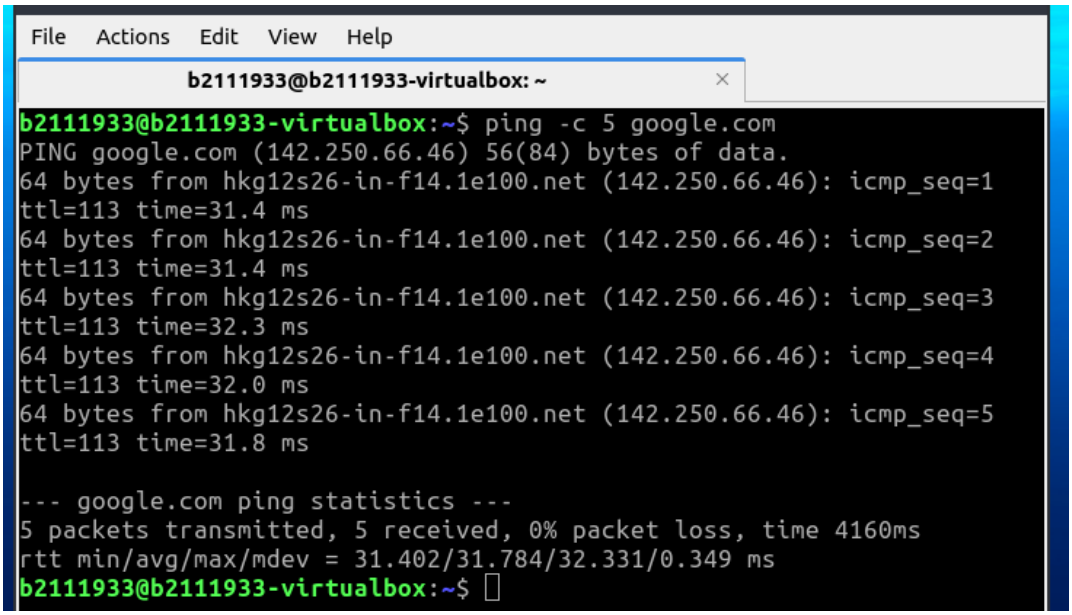
```
http_proxy="http://proxy.ctu.edu.vn:3128"
https_proxy="http://proxy.ctu.edu.vn:3128"
ftp_proxy="http://proxy.ctu.edu.vn:3128"
no_proxy=localhost,127.0.0.1
```

```
$source /etc/environment
```

Skip 1.3 & 1.4

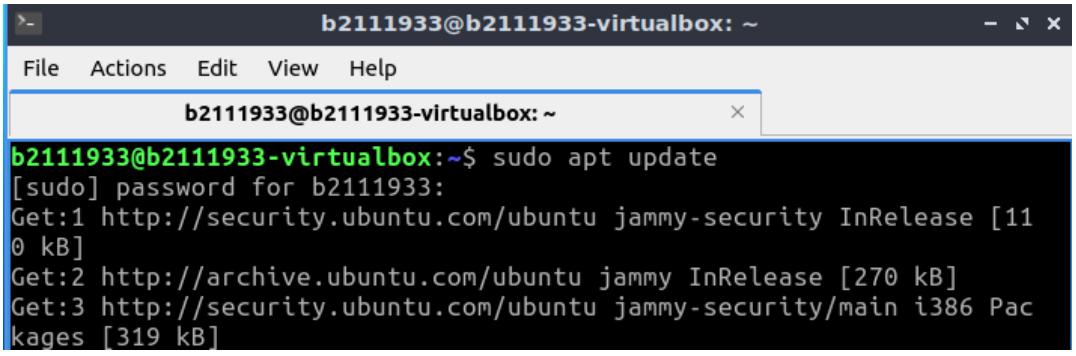
1.5. Verifying network connections

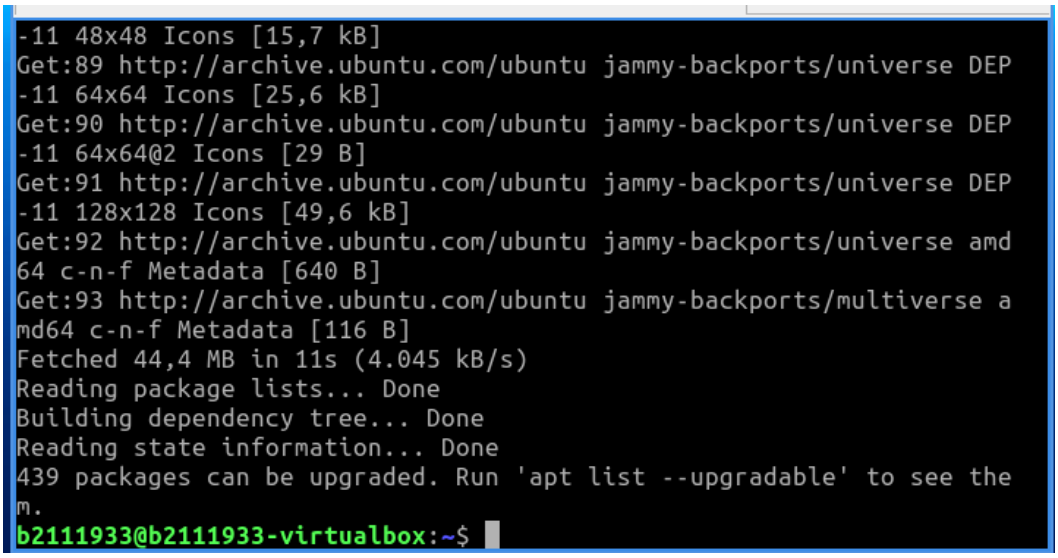
```
$ping -c 5 google.com
```



The **VM** can connect to **Google**

```
$sudo apt update
```





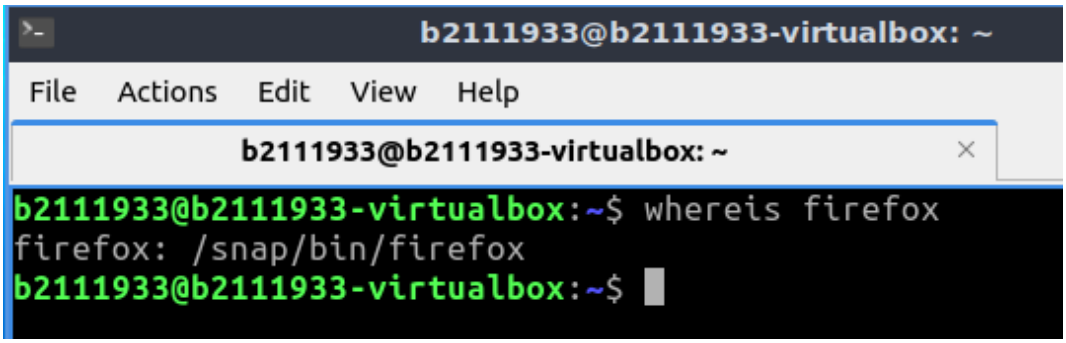
Updated successfully

2. Basic operations

2.1. Accessing Directories

- Locate the application titled "**firefox**" using **whereis** command.

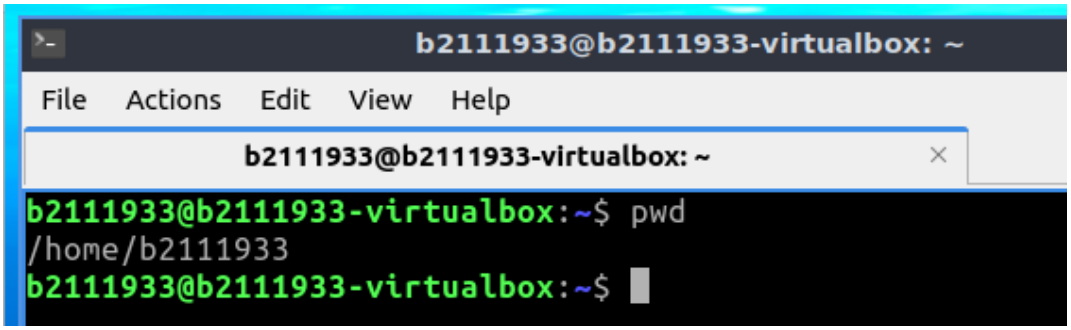
```
$whereis firefox
```



Located **“firefox”** application

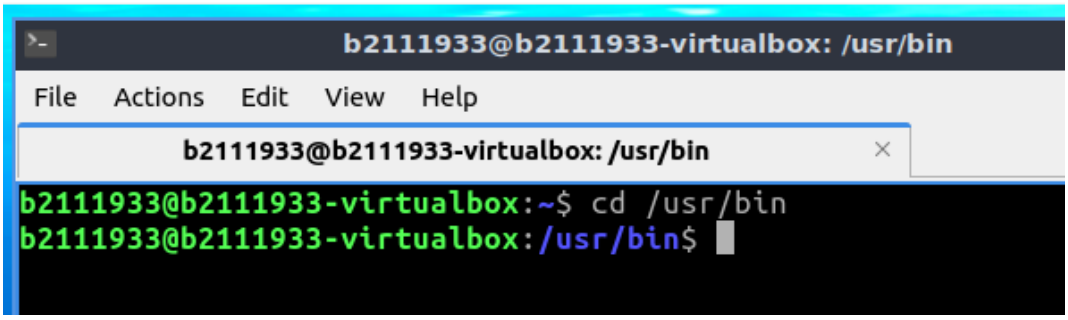
- Display the present working directory.

```
$pwd
```



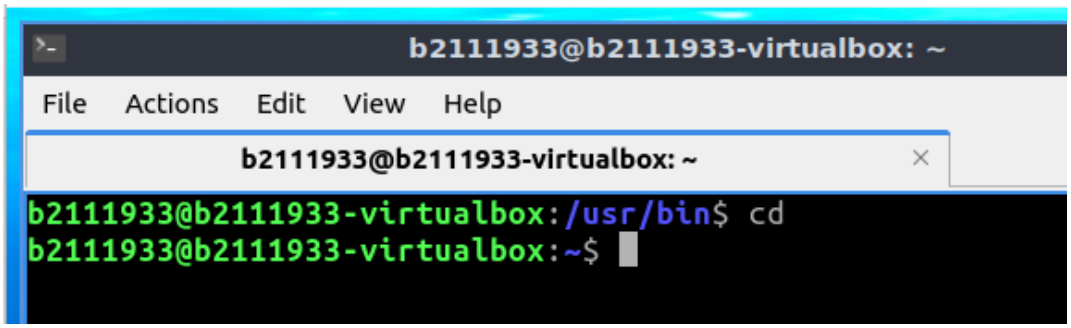
The present working directory is: **/home/b2111933**

- Change the current working directory to **/usr/bin** directory.
  `$cd /usr/bin`



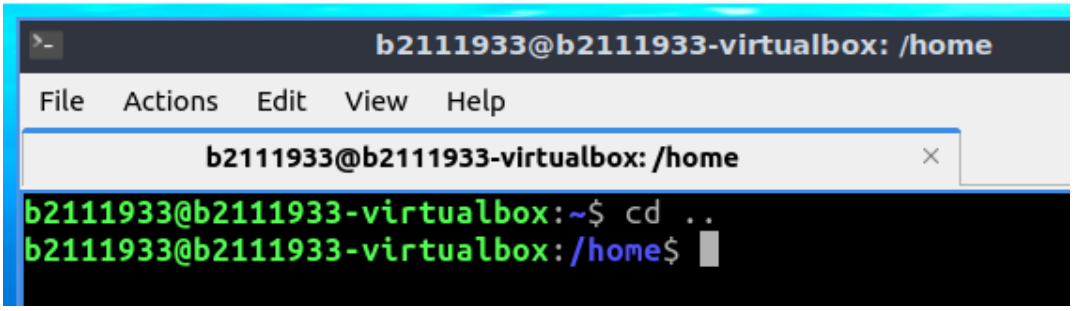
Now the **current directory** is **/usr/bin**

- Change the current working directory to **$HOME** directory.
  `$cd ~`



The **current working directory** returned to **$HOME directory** (**/home/b2111933)**

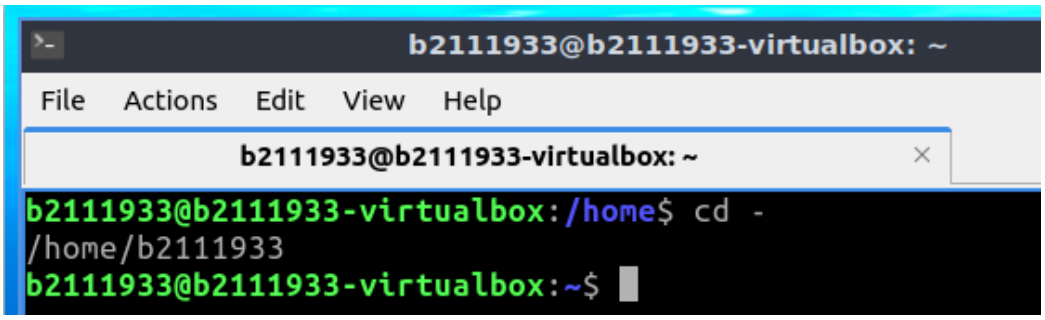
- Move to the parent directory.
  `$cd ..`



Move to the parent directory of **/home/b2111933** directory

- Go to the previous directory by the shortcut method, i.e., using '-' operator.

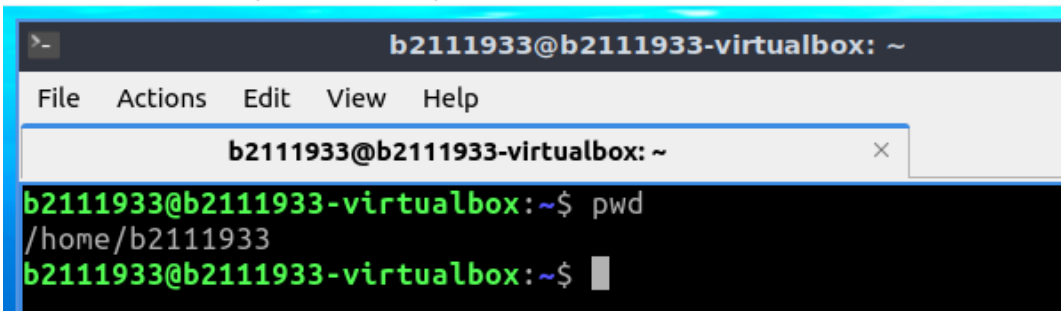
```
$cd -
```



Get back to the **previous directory** (**$HOME)** by the **shortcut method (-)**

- Display the present working directory.

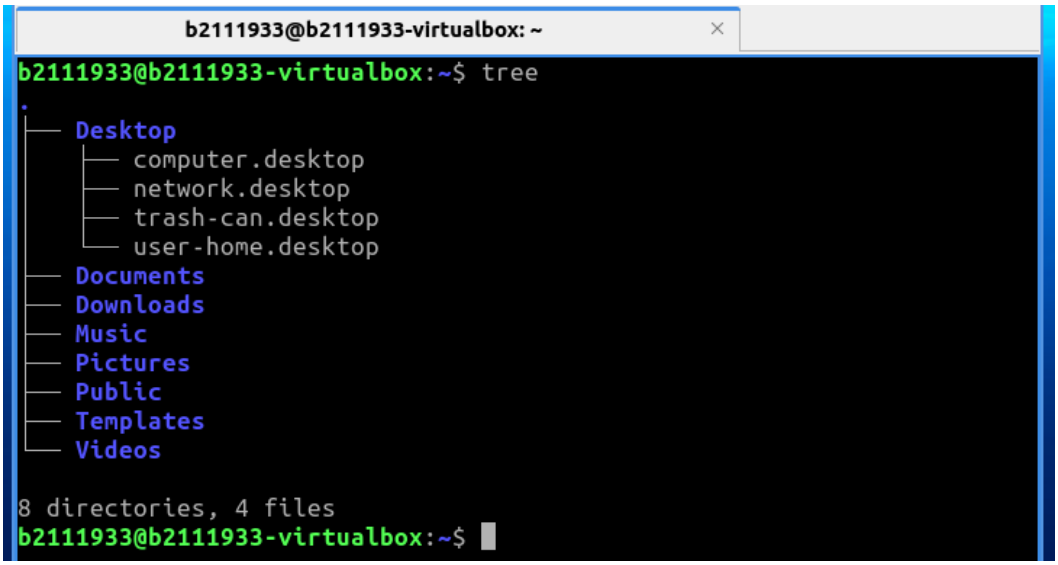
```
$pwd
```



The present working directory is: **/home/b2111933**

- Display filesystem tree of current directory

```
$tree
```



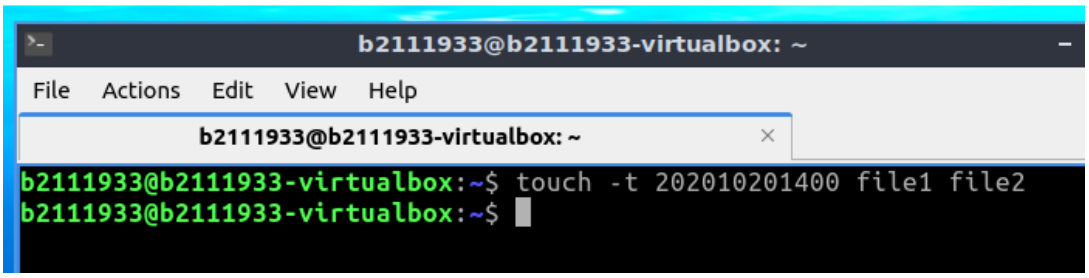
Display **filesystem tree** of **$HOME directory**

(take a screenshot of the console)

2.2. Working With Files and Directories

- Using **touch**, create **file1** and **file2** (two empty files) with timestamp: 20 October 2020 2:00 PM.

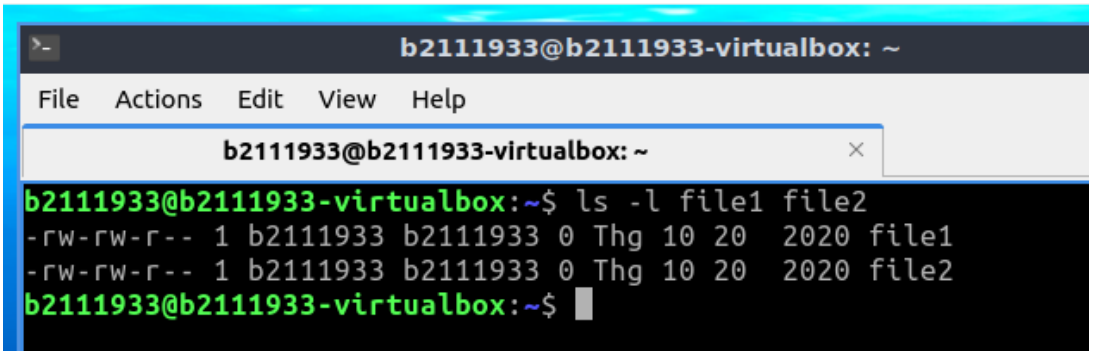
```
$touch -t 202010201400 file1 file2
```



Command **touch -t** will create files with timestamp

- Check for the existence of **file1** and **file2** using **ls -l** .

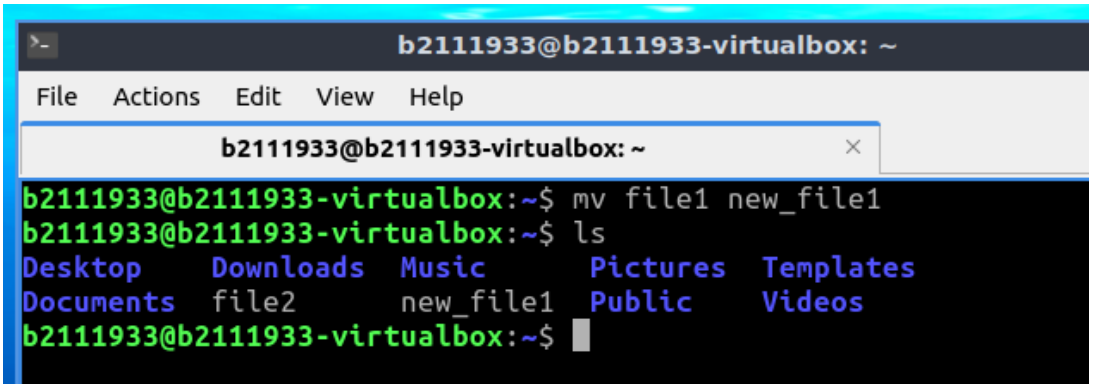
```
$ls -l file1 file2
```



List **file1** & **file2** with full details

- Rename **file1** to **new_file1** using **mv**.

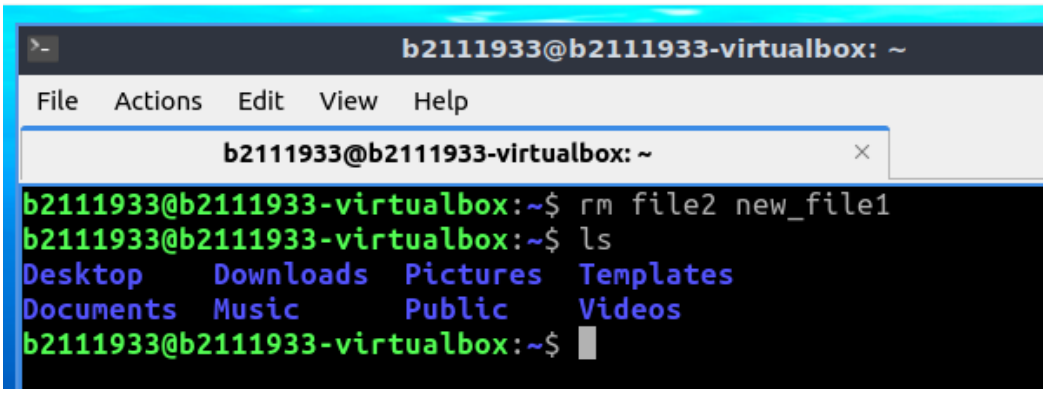
```
$mv file1 new_file1
```



Rename **file1** to **new_file1** using **mv**

- Remove **file2** and **new_file**1 using **rm** without any options.

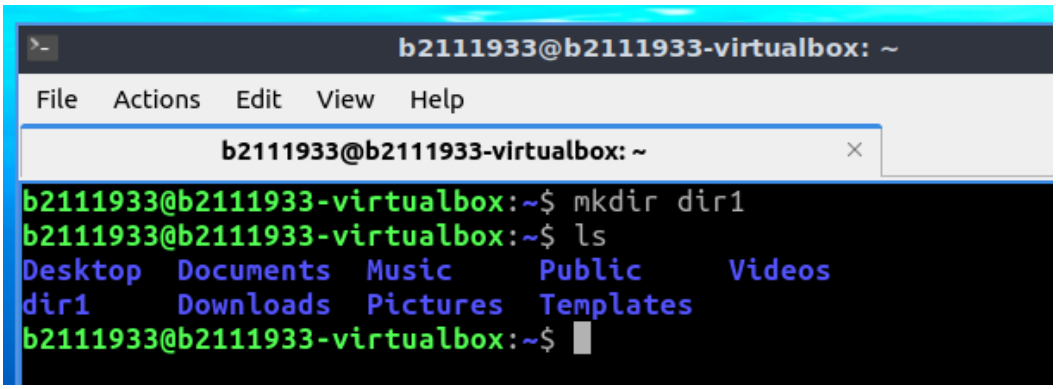
```
$rm file2 new_file1
```



Removed **file2** & **new_file**1 successfully

- Create a directory named **dir1**, using **mkdir**.

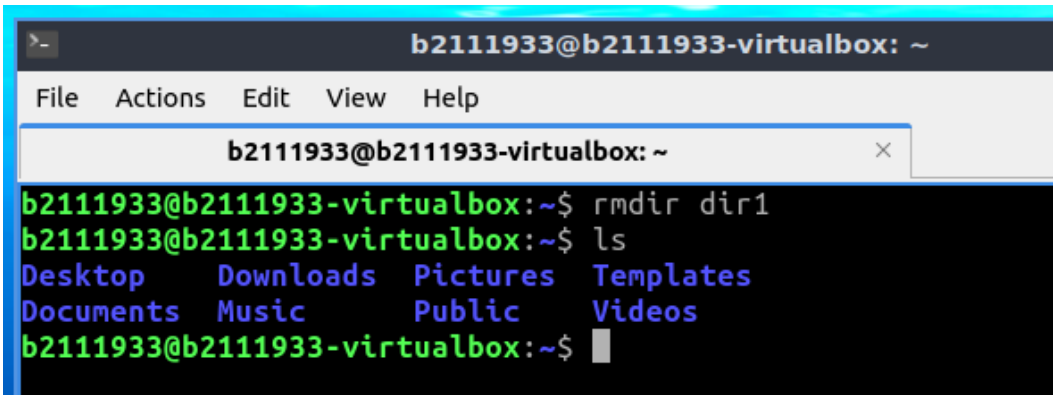
```
$mkdir dir1
```



Created a directory named **dir1**

- Remove **dir1** using **rmdir** without any options

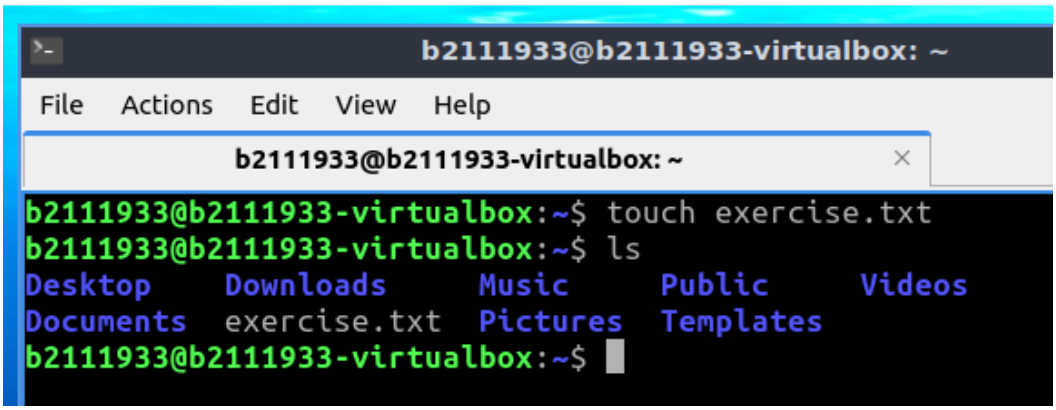
```
$rmdir dir1
```



Removed **dir1** using **rmdir**

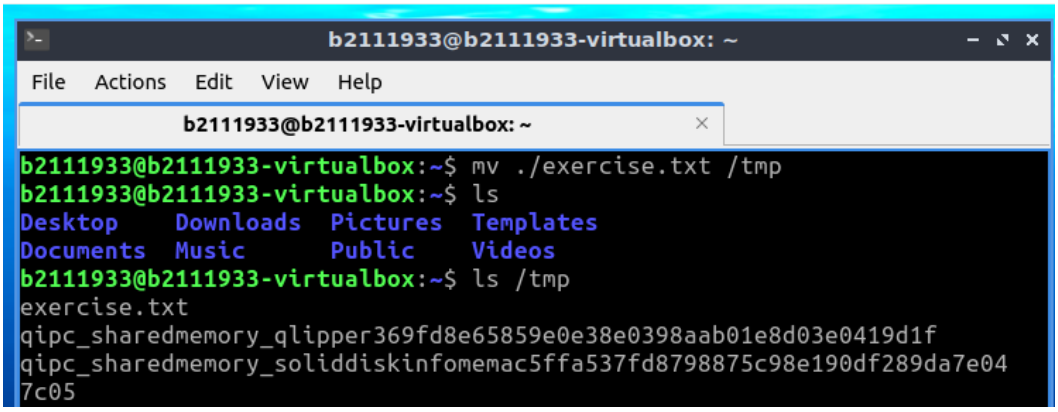
- Create an empty file named **exercise.txt** and move it to **/tmp** using relative pathname. Then delete this file using an absolute pathname

```
$touch exercise.txt
```



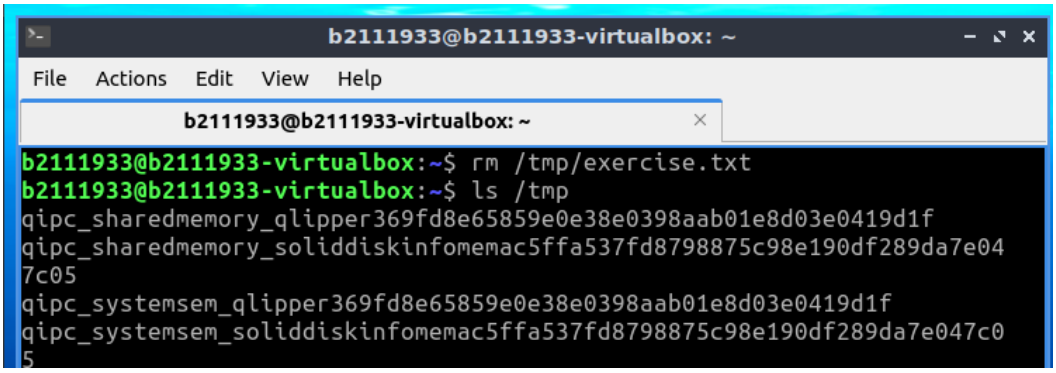
Created an empty file named **exercise.txt**

```
$mv ./exercise.txt /tmp
```



Moved file named **exercise.txt** to **/tmp** using relative pathname

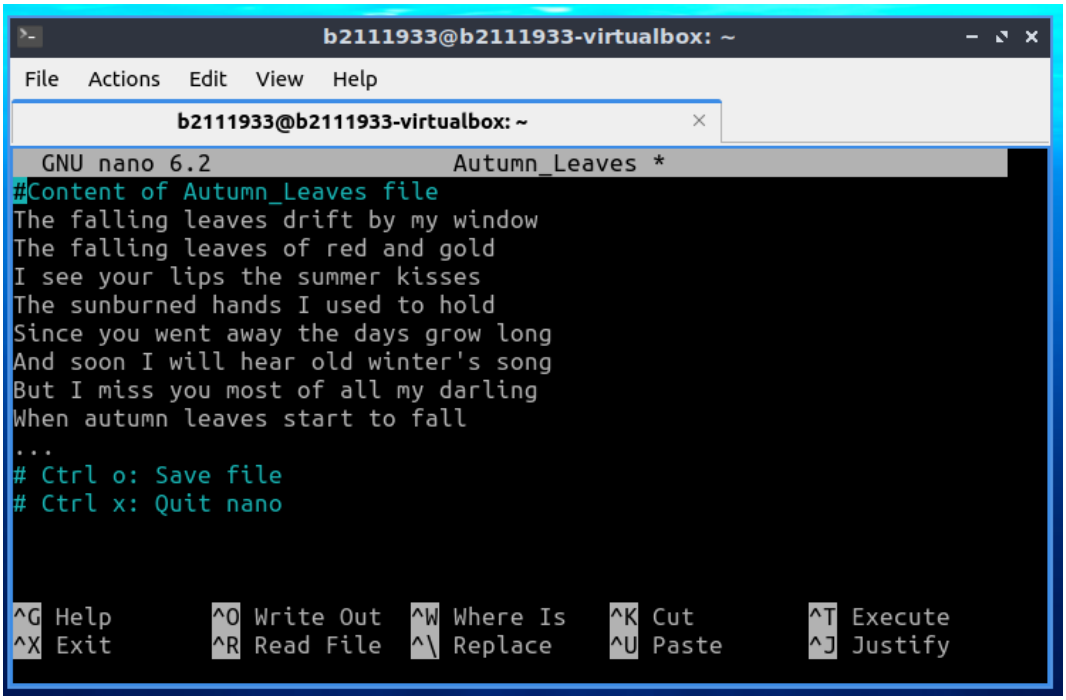
```
$rm /tmp/exercise.txt
```



Deleted file **exercise.txt** using absolute pathname **/tmp/exercise.txt**

- Create a file **Autumn_Leaves** with its content as below using **nano** tool.

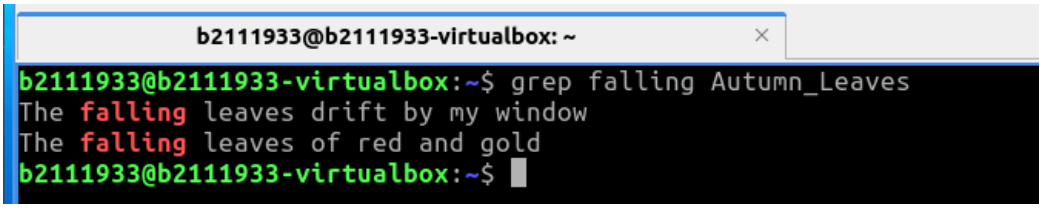
```
$nano Autumn_Leaves
#Content of Autumn_Leaves file
The falling leaves drift by my window
The falling leaves of red and gold
I see your lips the summer kisses
The sunburned hands I used to hold
Since you went away the days grow long
And soon I will hear old winter's song
But I miss you most of all my darling
When autumn leaves start to fall
...
# Ctrl o: Save file
# Ctrl x: Quit nano
```



Create a file **Autumn_Leaves** with its content using **nano** tool

- Search for the string "**falling**" in the file **Autumn_Leaves**

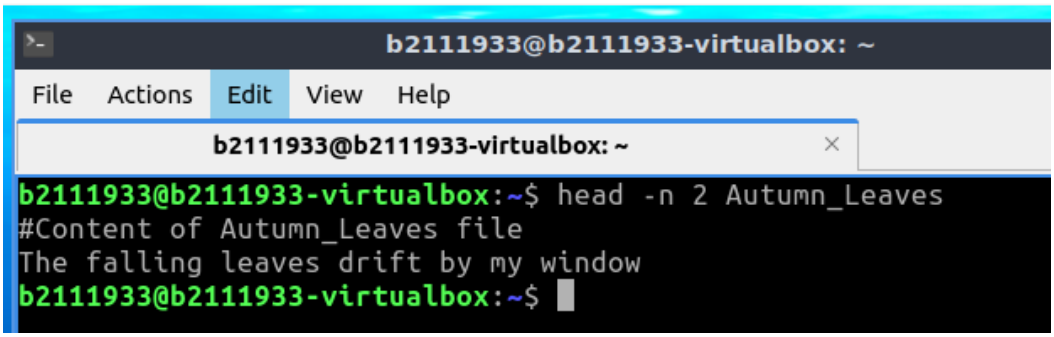
```
$grep "falling" Autumn_Leaves
```



There are 2 strings "**falling**" in the file **Autumn_Leaves**

- Display the first 2 lines of the file **Autumn_Leaves**

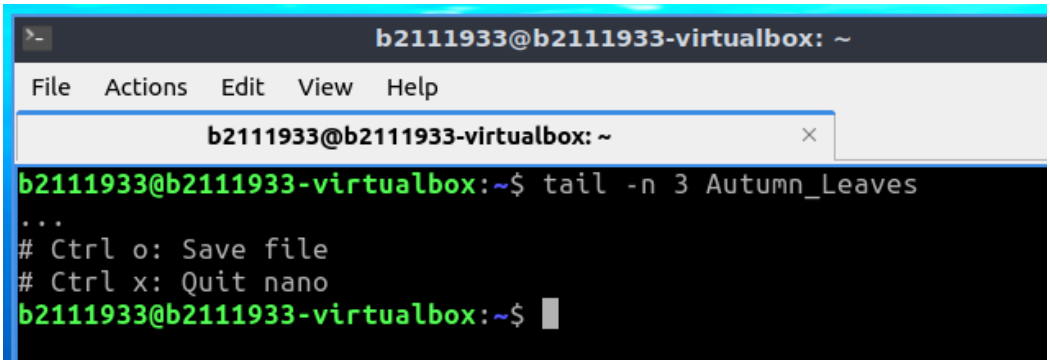
```
$head -n 2 Autumn_Leaves
```



Displayed the first 2 lines of the file **Autumn_Leaves**

- Display the last 3 lines of the file **Autumn_Leaves**

```
$tail -n 3 Autumn_Leaves
```



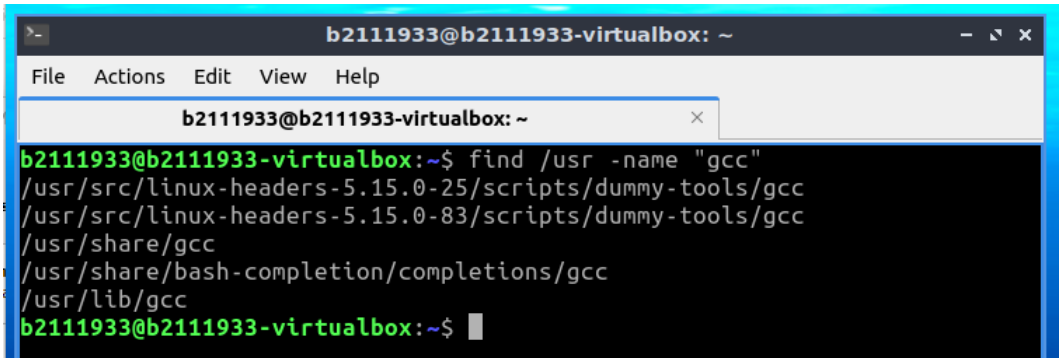
Displayed the last 3 lines of the file **Autumn_Leaves**

(take a screenshot of the console)

2.3. Finding files in a directory

- Search for a file with the name **gcc** in the **/usr** folder using **find** command.

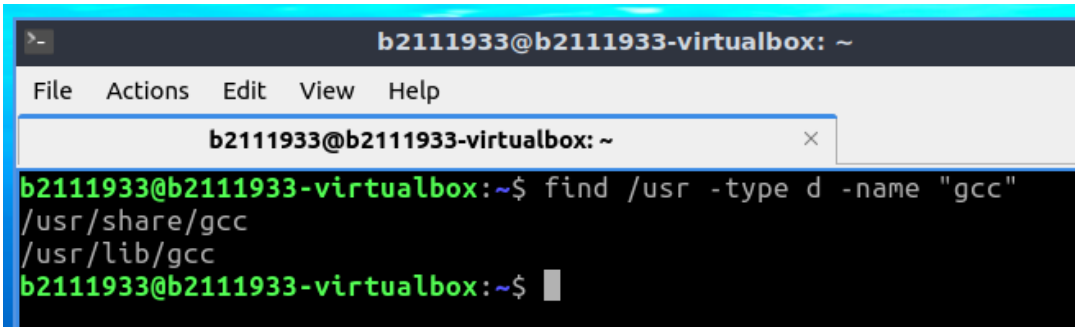
```
$find /usr -name "gcc"
```



Searched for files with the name “**gcc”** in the **/usr** folder using **find** command

- Search for a directory with the name **gcc**, in the **/usr** folder using **find** command.

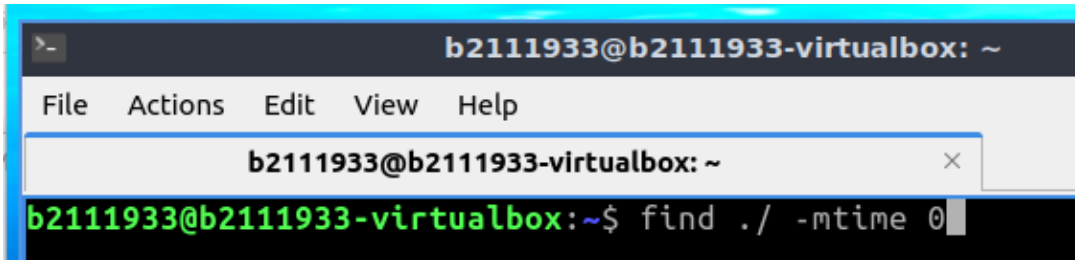
```
$find /usr -type d -name "gcc"
```

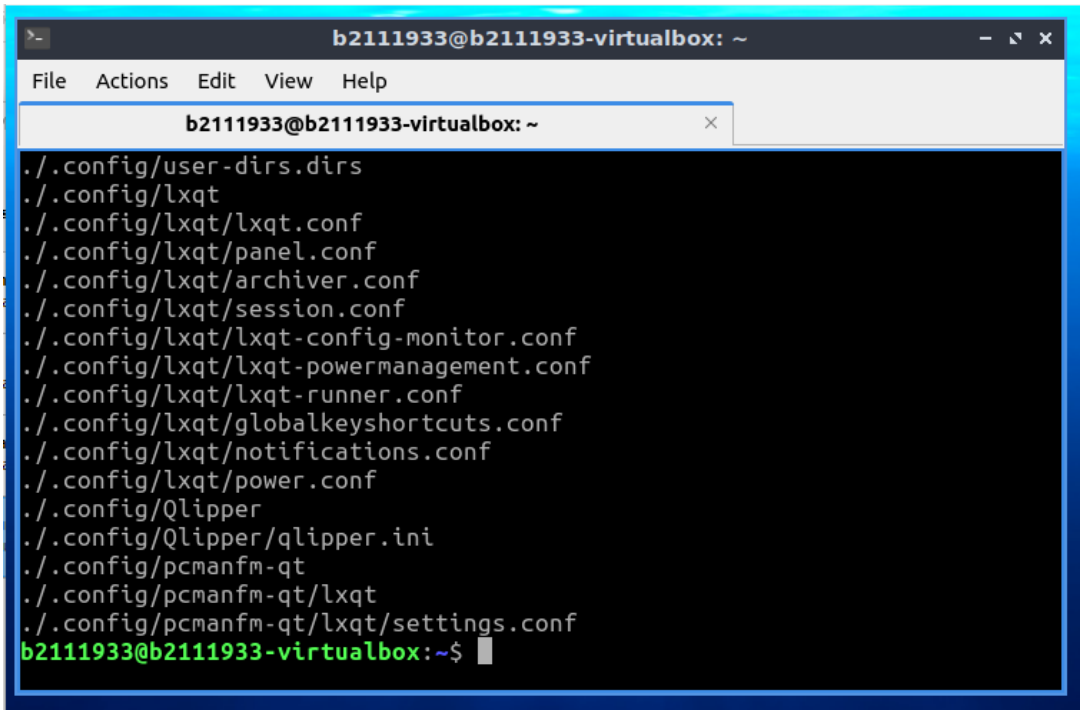


Searched for directories with the name **gcc**, in the **/usr** folder using **find** command

- Search for files in the current directory which were modified today.

```
$find ./ -mtime 0
```

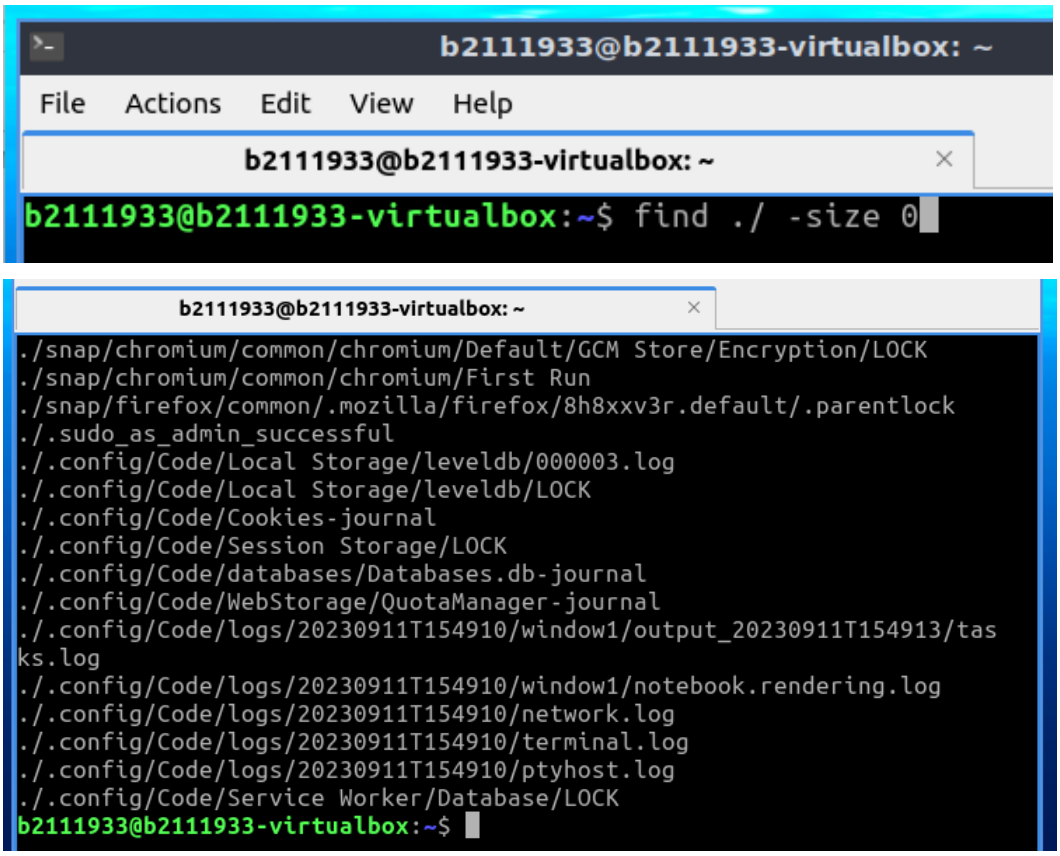




There are a lot of **files** in the **current directory** which were **modified today**

- Search for files with size **0 bytes**

```
$find ./ -size 0
```



There's many **files** with **size** that is around **0 bytes**

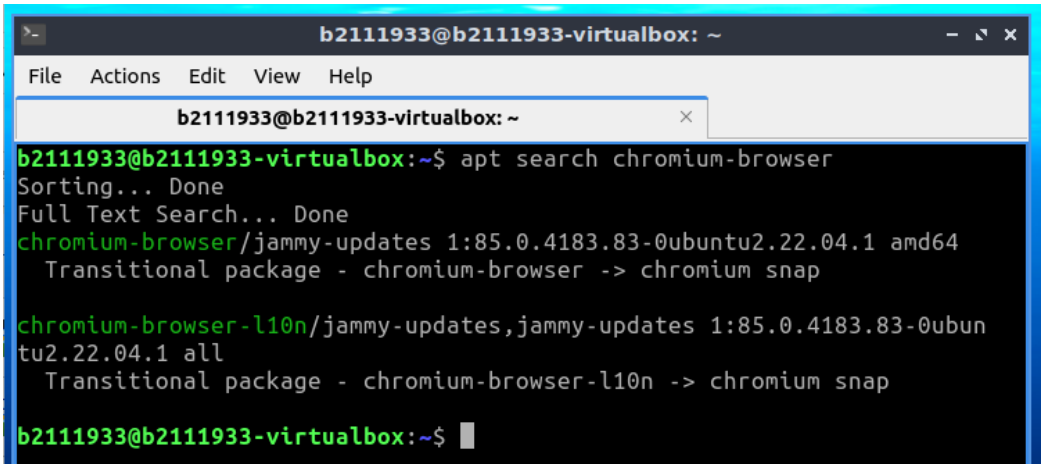
(take a screenshot of the console)

3. Installing and Removing software packages

3.1. Install from default repositories

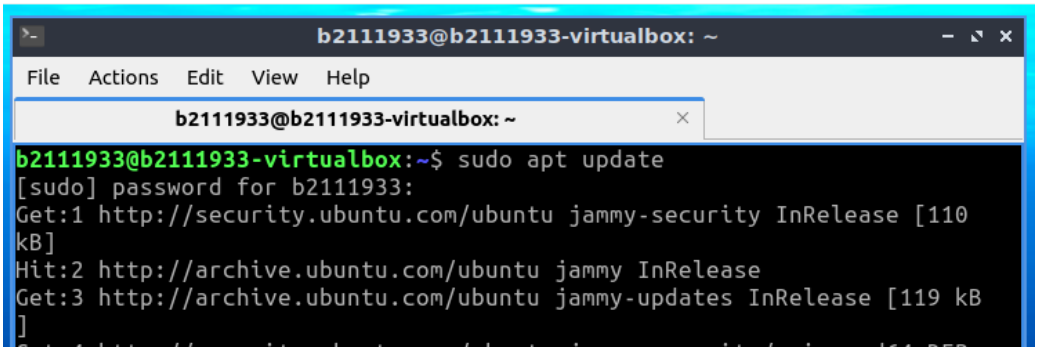
- Using apt tool, find the **chromium-browser** package.

```
$apt search chromium-browser
```

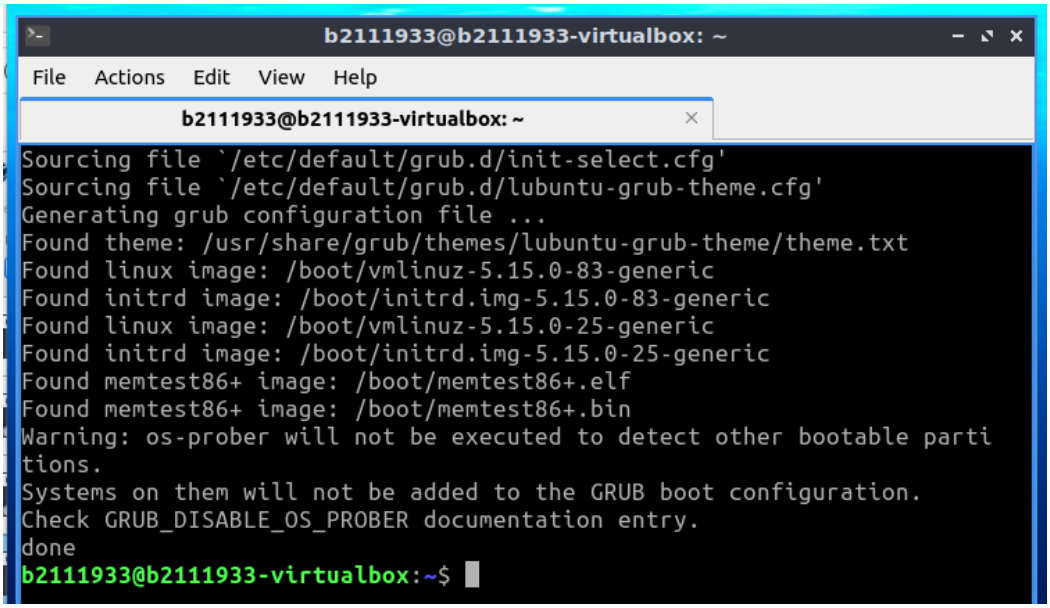


Found **chromium-browser** package

```
$sudo apt update
```

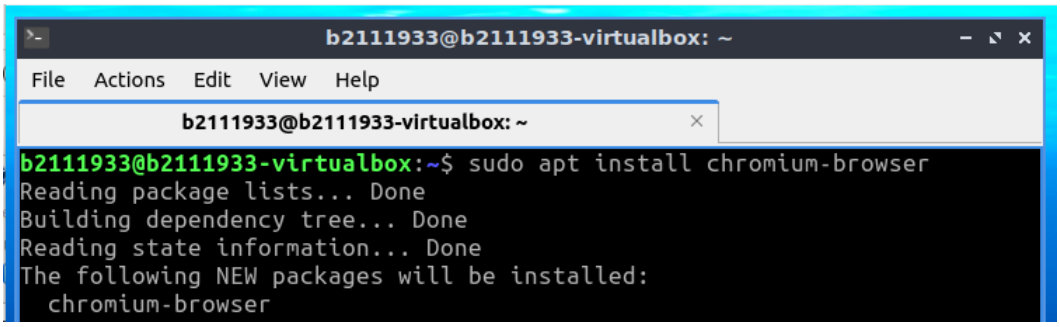


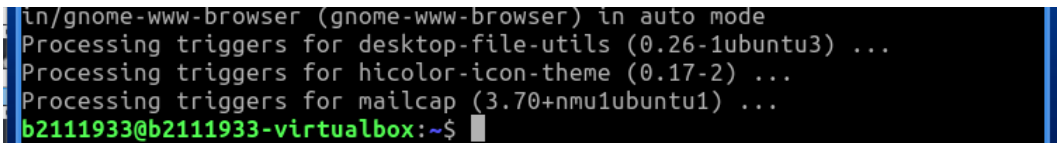

```
$sudo apt upgrade
```



Upgraded **Lubuntu** successfully

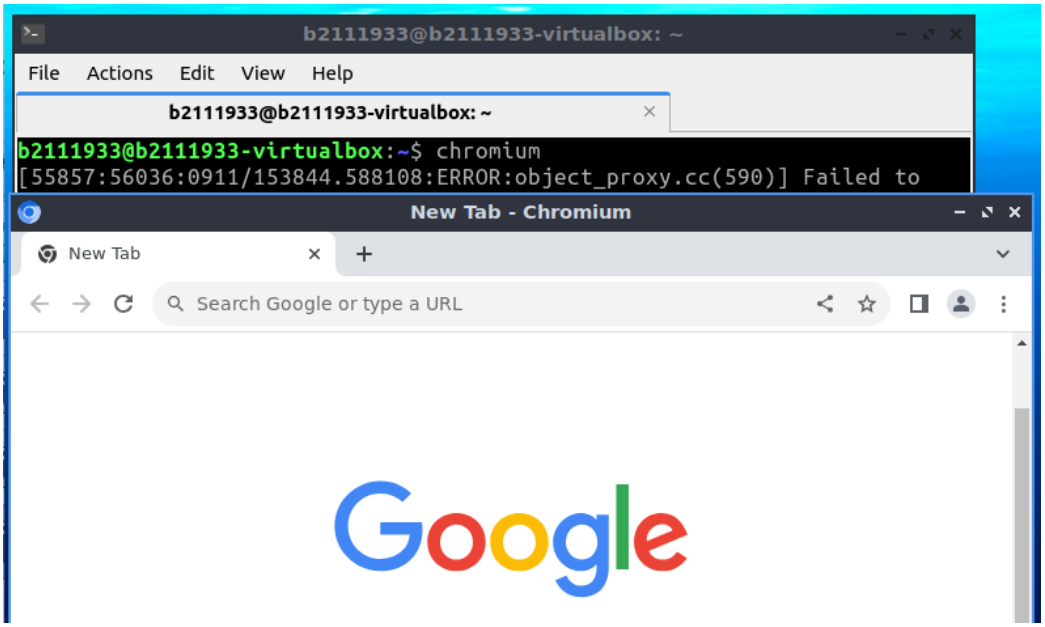
```
$sudo apt install chromium-browser
```





Installed **chromium** successfully

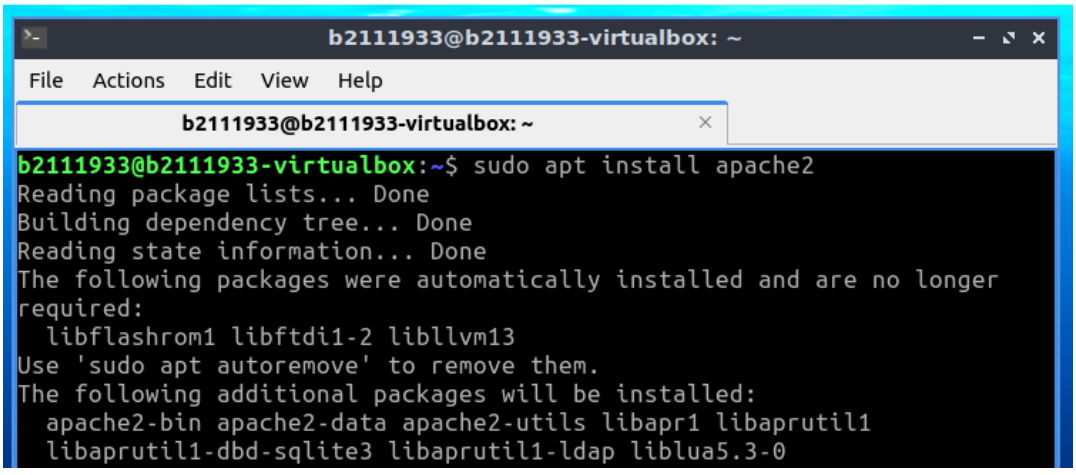
```
#
$sudo apt remove chromium-browser
$sudo apt install chromium-browser
```
(Skip those steps)

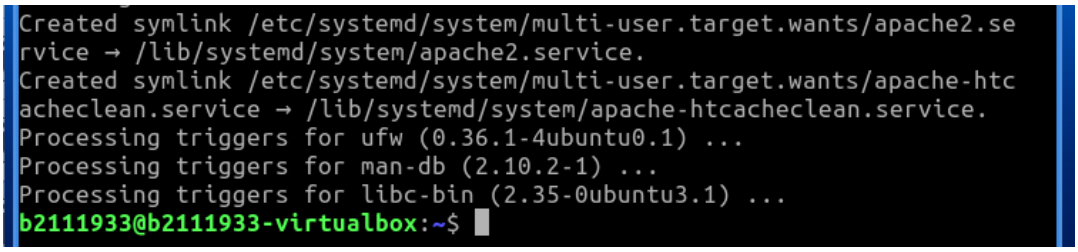


Result

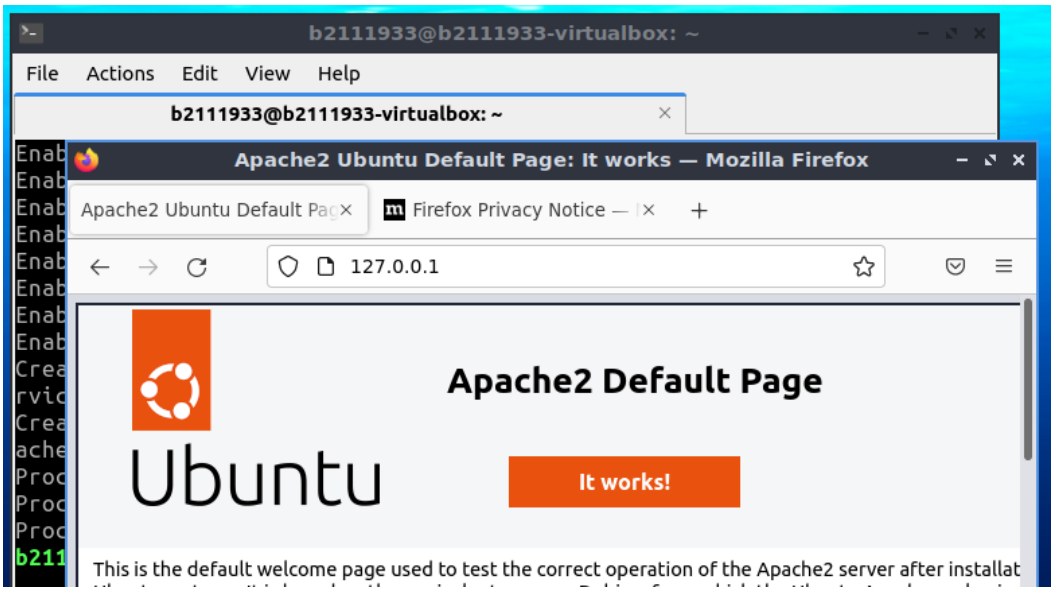
- Install **apache2** package using apt tool

```
$sudo apt update
$sudo apt install apache2
```





Installed **apache2** successfully

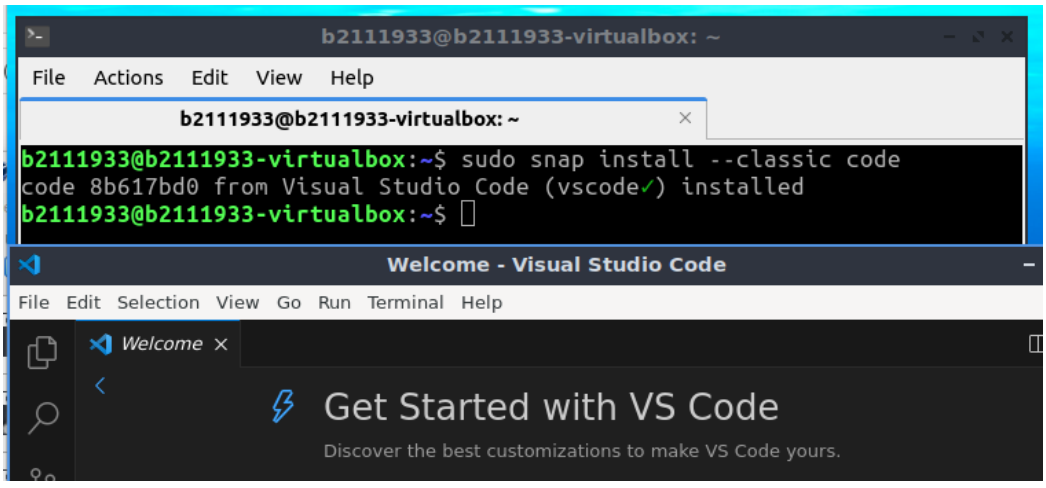


Result

3.2. Install using snap

- Install VScode using snap tool

```
$sudo snap install --classic code
```



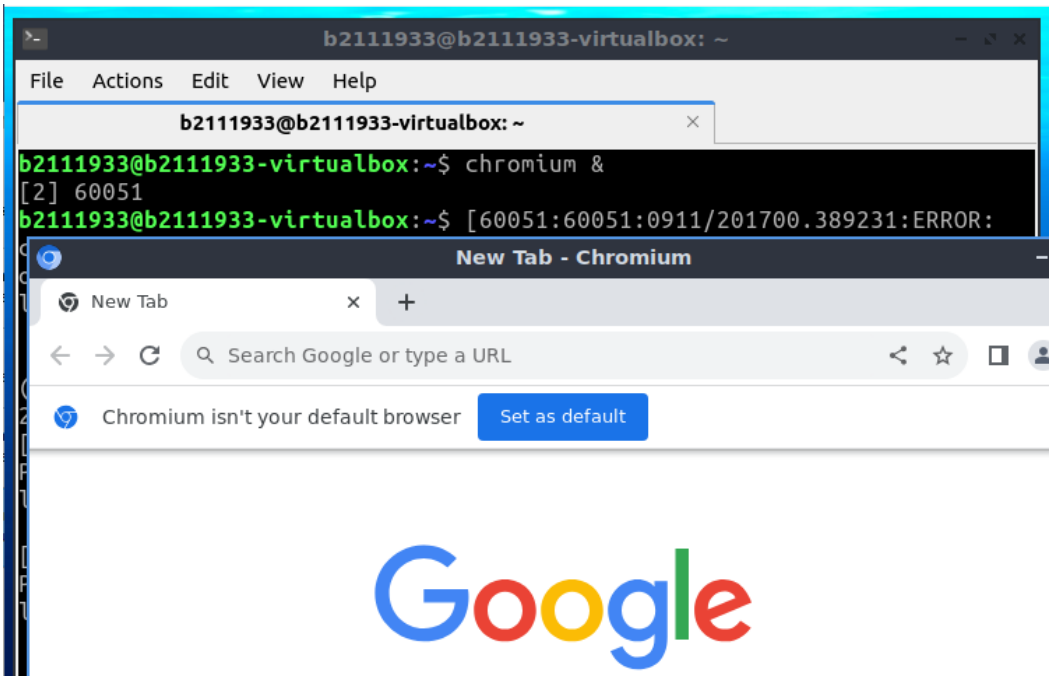
Installed **VScode** successfully

(take a screenshot of the console)

4. Process and daemons (services)

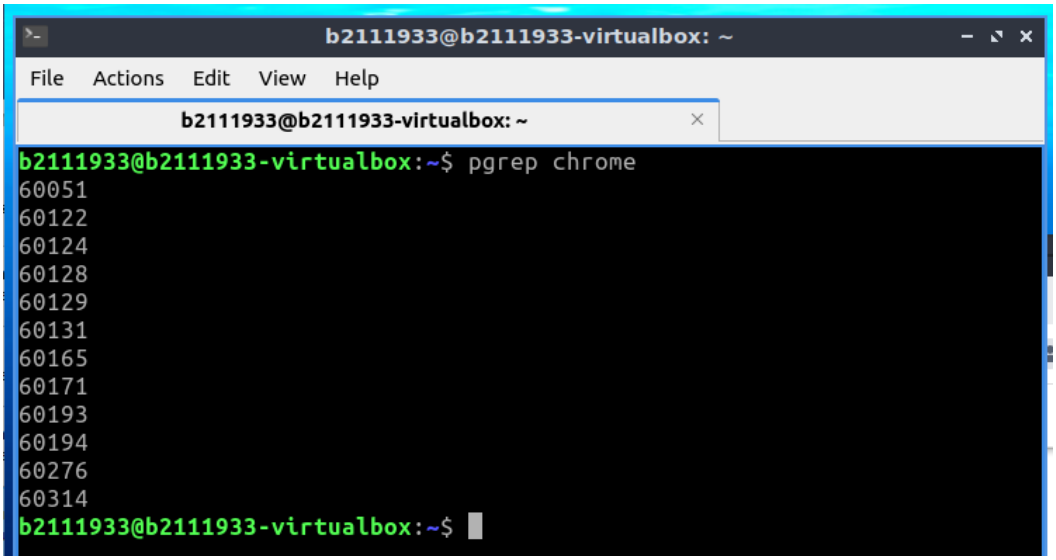
- Launch **chromium-browser** application; then find its PID using **pgrep** command

```
$chromium
```



Run **chromium-browser** application in background with "**&**"

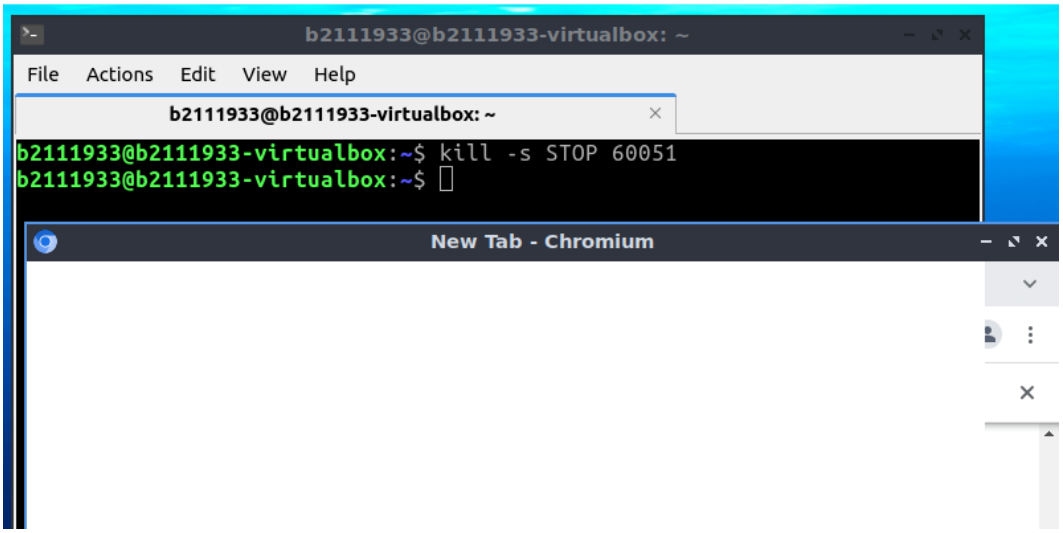
```
$pgrep chrome
```



Found its **PID** using **pgrep** command

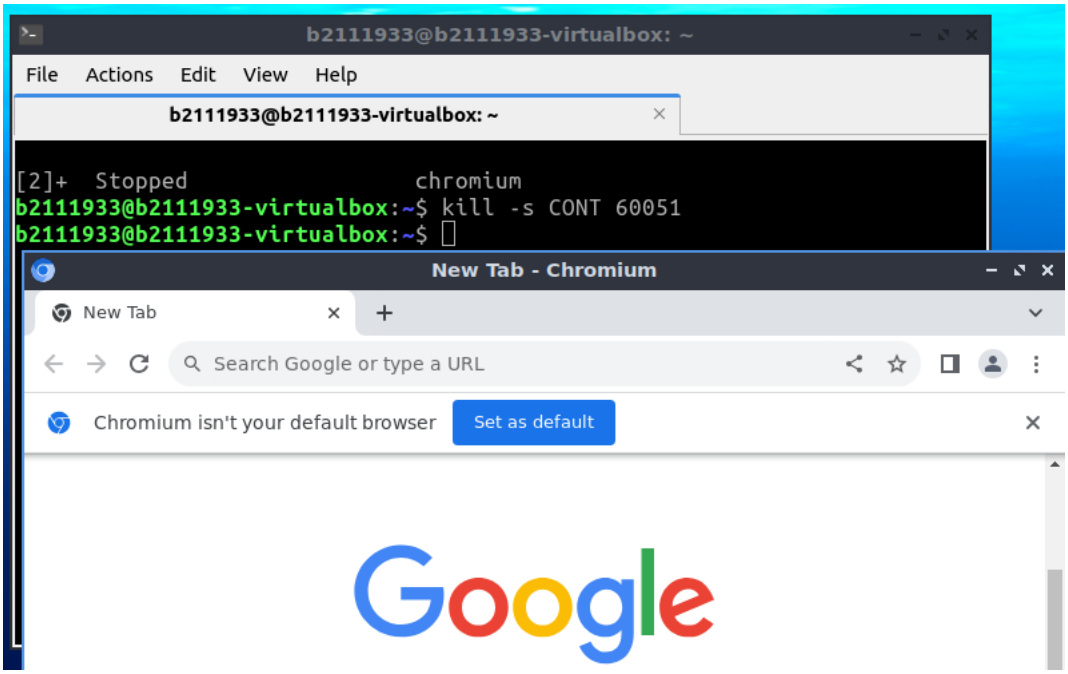
- Stop/continue/terminate **chromium-browser** application using **kill** command

```
$kill -s STOP <PID>
```



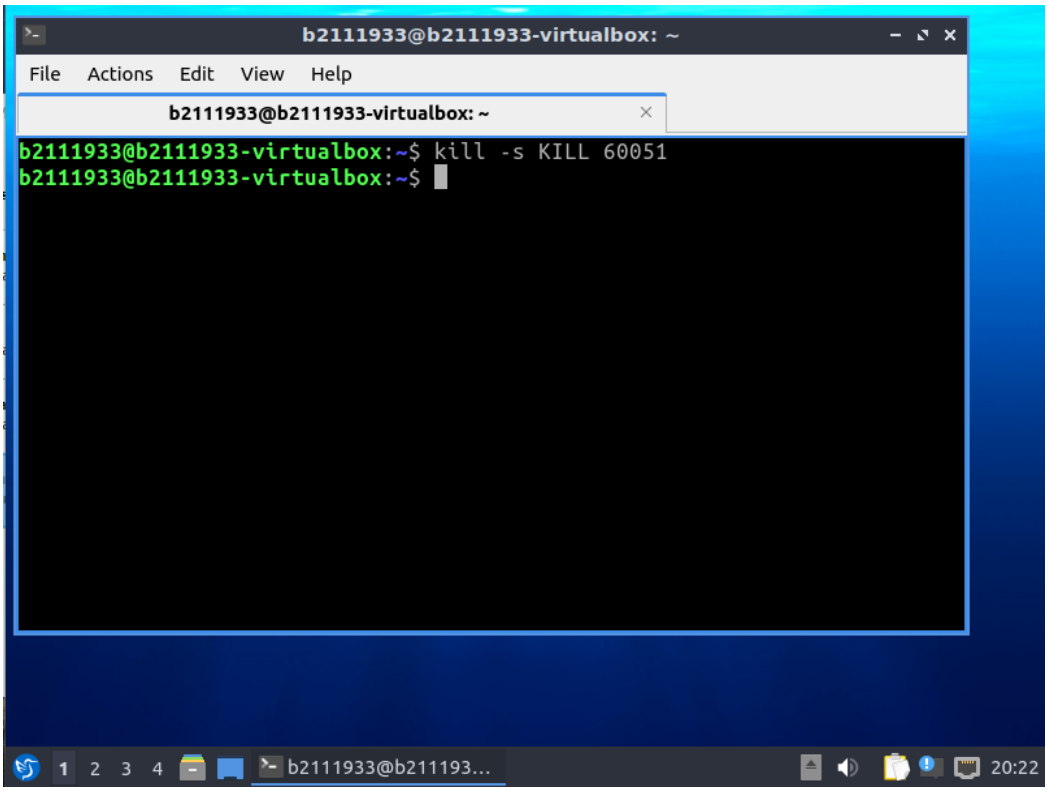
Chromium is stop working

```
$kill -s CONT <PID>
```



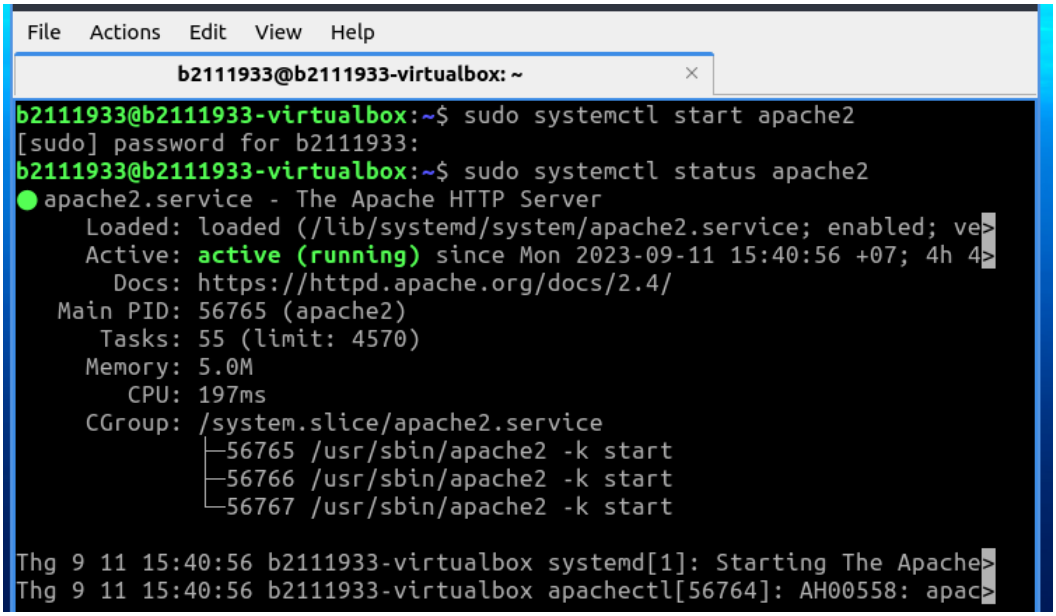
Now **Chromium** is working again

```
$kill -s KILL <PID>
```



- Start and then display the status of **apache2** web server using systemctl command

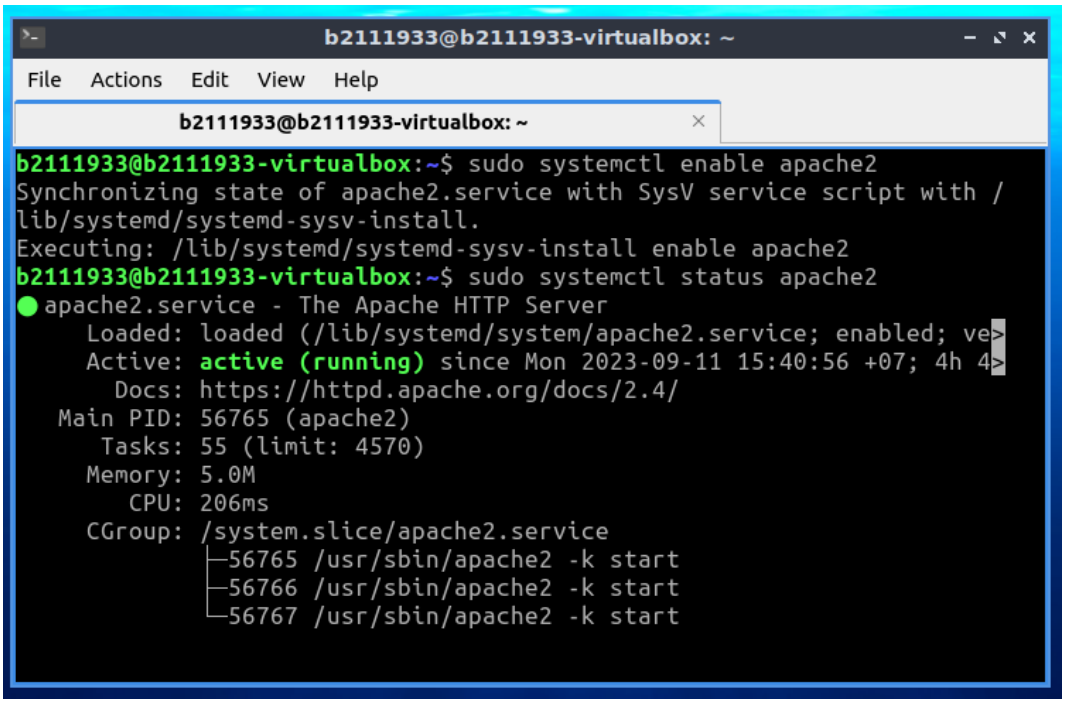
```
$sudo systemctl start apache2
$sudo systemctl status apache2
```



Start and display the status of **apache2** using **systemctl** command (started)

- Enable **apache2** web server starting on booting time

```
$sudo systemctl enable apache2
```



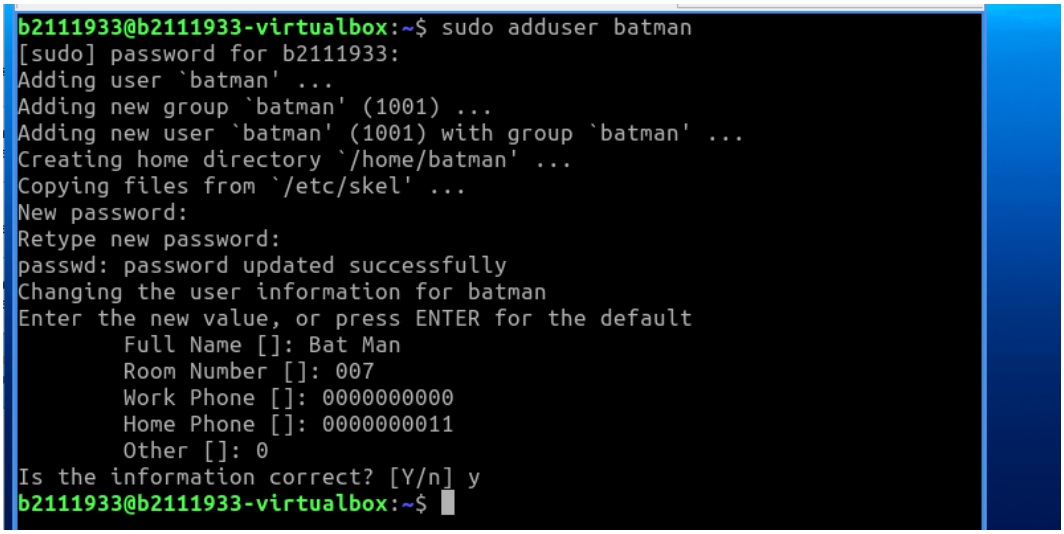
Enable **apache2** web server starting on booting time and display its status (take a screenshot of the console)

5. Local Security Principles

5.1. Create a new user and sudo

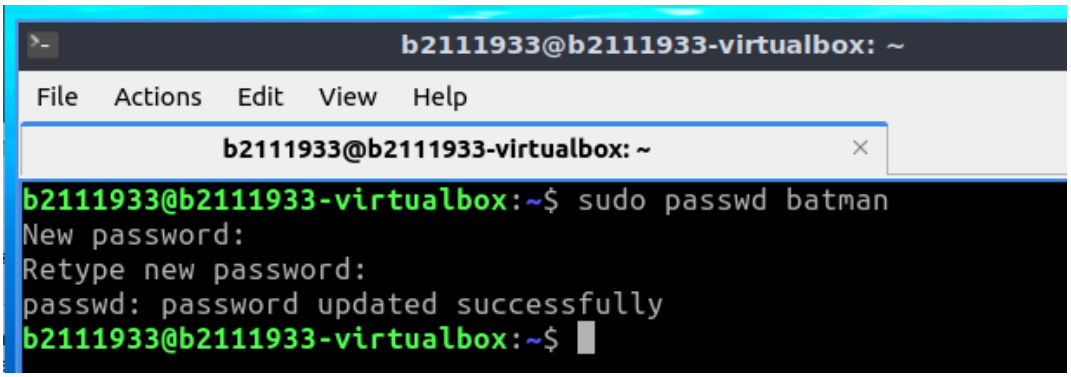
- Create a new user (batman), using **adduser**, and give the user an initial password with **passwd**.

```
$sudo adduser batman
```



Created a **new user** with username "**batman**" and filled the information

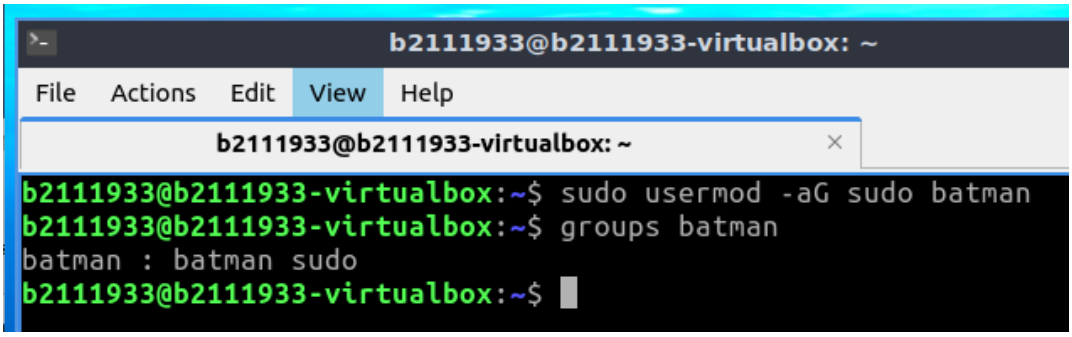
```
$sudo passwd batman
```



Updated **password** for user "**batman**"

- Configure this user to be able to use **sudo** command

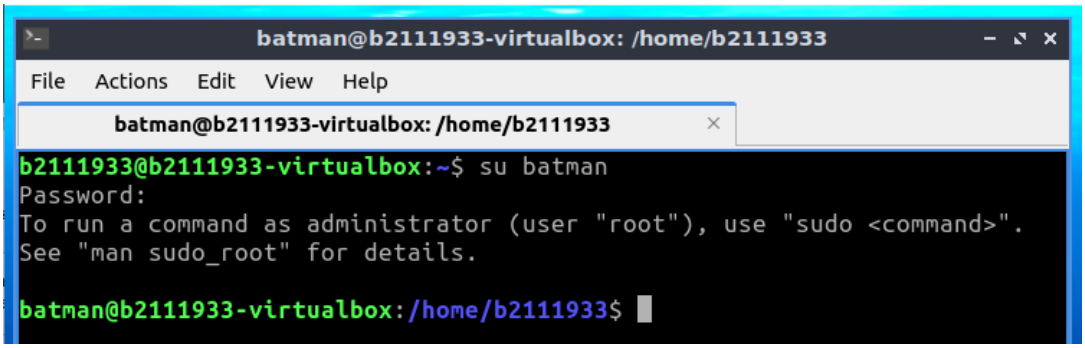
```
$sudo usermod -aG sudo batman
```



Added "**batman**" to "**sudo**" group to be able to use **sudo** command

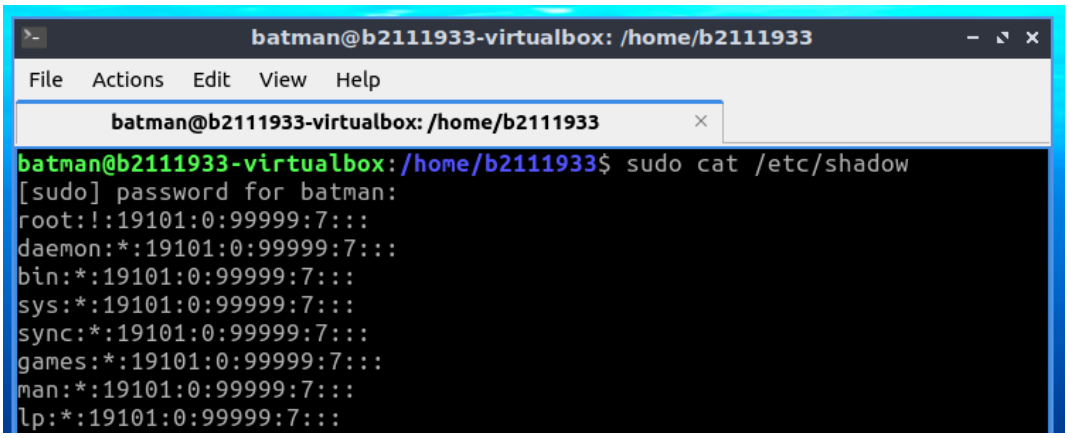
- Login as or switch to this new user and make sure you can execute a command (cat /etc/shadow) that requires root privilege.

```
$su batman
```



Switched to the new user "**batman**"

```
$sudo cat /etc/shadow
```

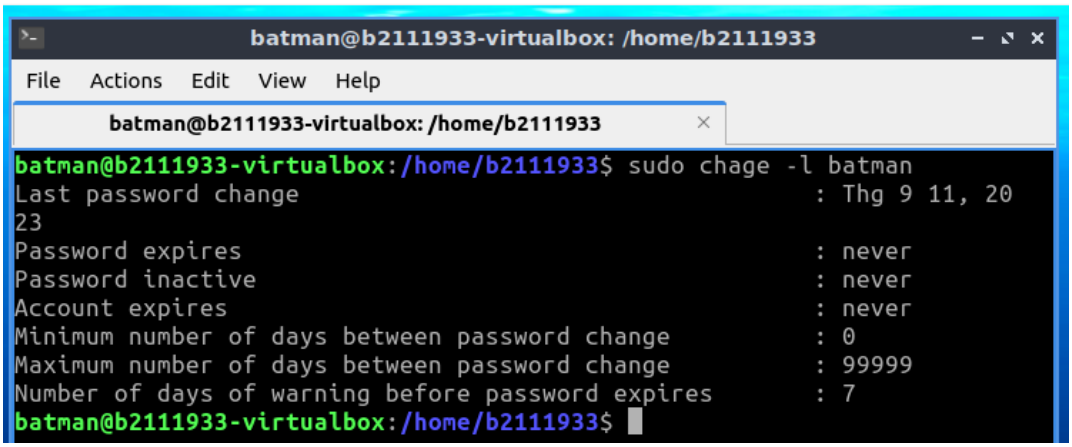


User "**batman**" can execute a command (cat /etc/shadow) that requires root privilege (take a screenshot of the console)

5.2. Password aging

- With the newly created user from 5.1, look at the password aging for the user.

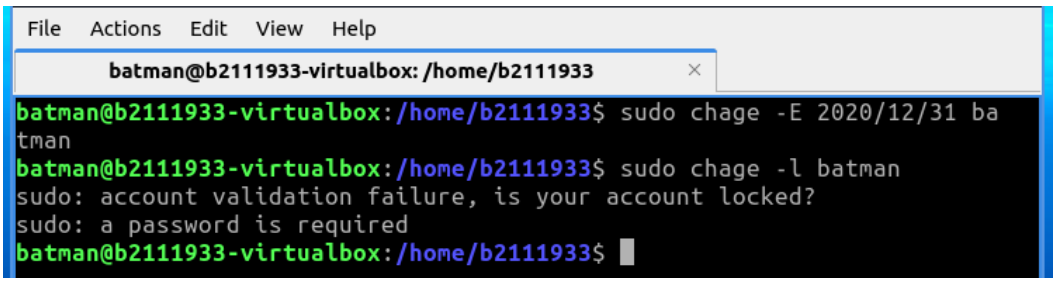
```
$sudo chage -l batman
```



Password aging for the user "**batman**"

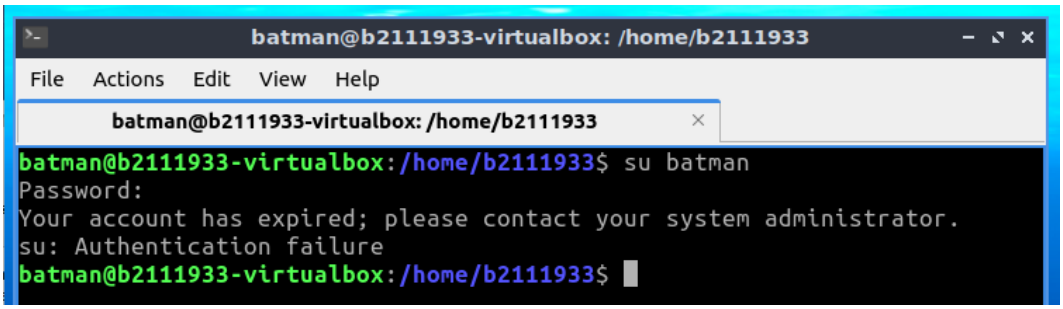
- Modify the expiration date for the user, setting it to be something that has passed, and check to see what has changed.

```
$sudo chage -E 2020/12/31 batman
```



Modified the expiration date for user "**batman**", now it is locked

```
$su batman
```



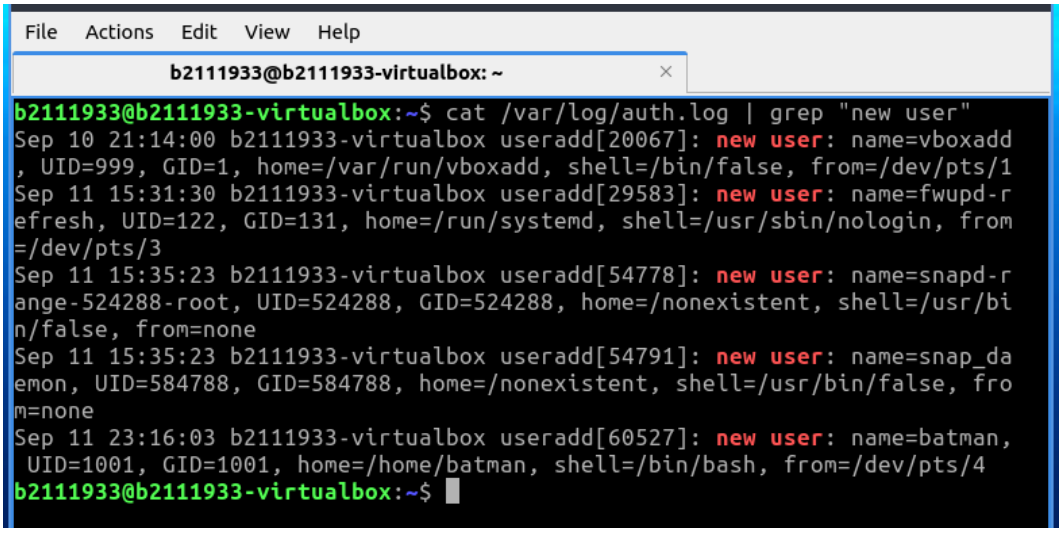
Account "**batman**" has expired as you can see

(take a screenshot of the console)

5.3. Log files

- Display the time when the user in 5.1 created

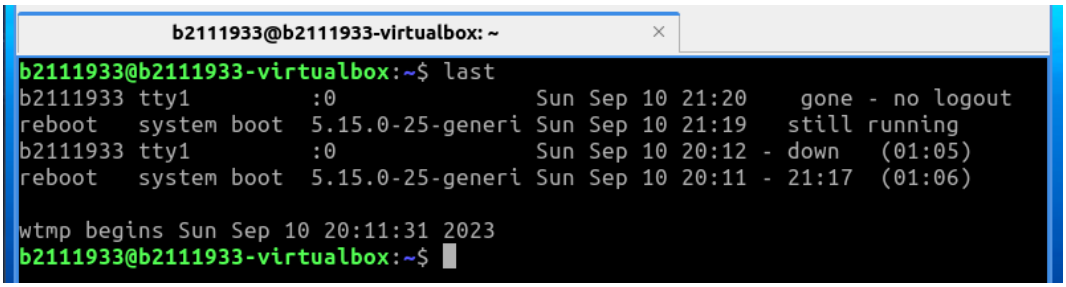
```
$cat /var/log/auth.log | grep "new user"
```



Check the **log file** that displays the time when **the user in 5.1** created

- Find the time of the last log in to the system

```
$last
```



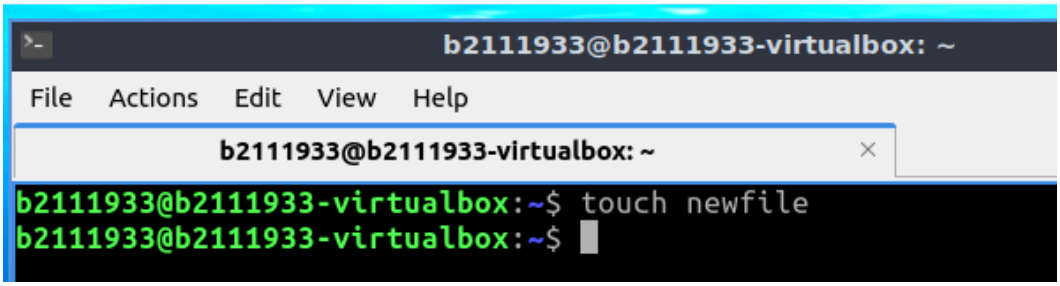
The **last log in** to the system was **21:20 10/9/2023**

(take a screenshot of the console)

6. File permissions

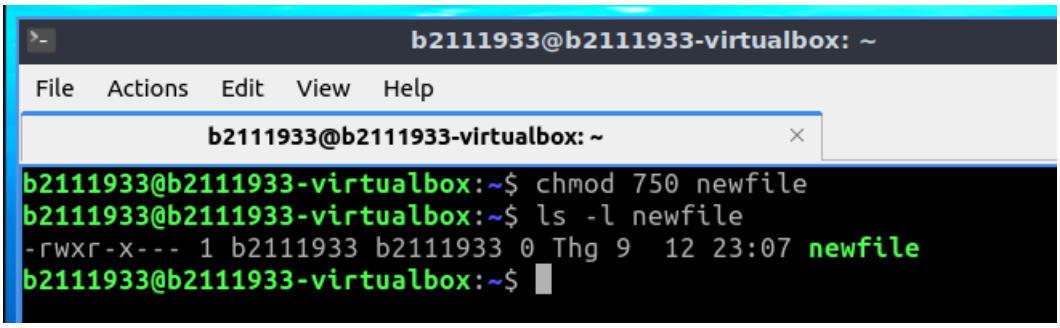
- Create a new file **newfile**; then change its permissions to **rwxr-x---**

```
$touch newfile
```



Created a new file named "**newfile**"

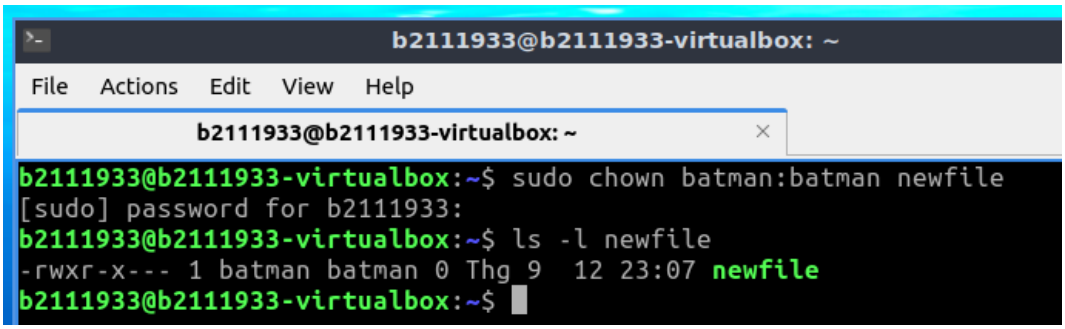
```
$chmod 750 newfile
```



Changed **newfile's permissions** to **rwxr-x---**

- Change the user and group ownership of **newfile** to the user 5.1

```
$sudo chown batman:batman newfile
```



Change the user and group ownership of **newfile** to the user "**batman**"

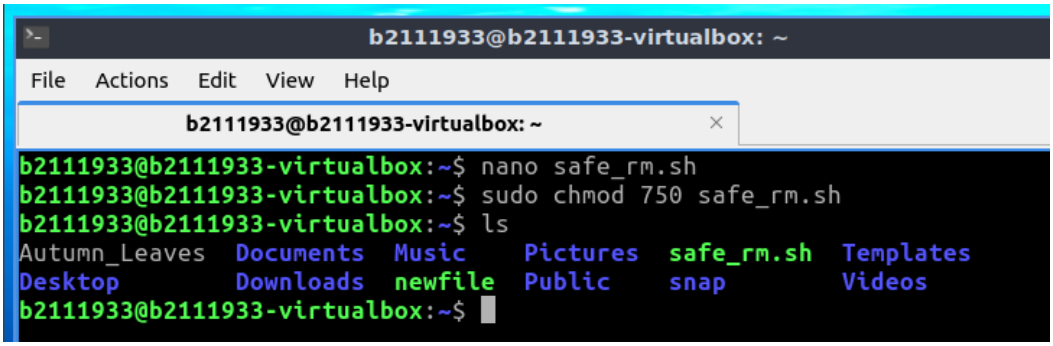
(take a screenshot of the console)

7. Shell scripting

The default rm command will not confirm before it deletes any regular files. Write a short script called *safe_rm.sh*, such that it will make a copy before deleting a single file (that is, we do not use wildcard expressions for this problem) by do the following:

- Take one and only one argument at the command line (hint: search for an expression representing the number of arguments in the shell scripts). Print out an error message if no argument or more than one argument are provided (hint: use echo).
- Create a directory “safe_rm_recycle” in the current one if it is not already created. Copy the file indicated by the first argument to this “safe_rm_recycle” folder. Remove this file in the current working directory.

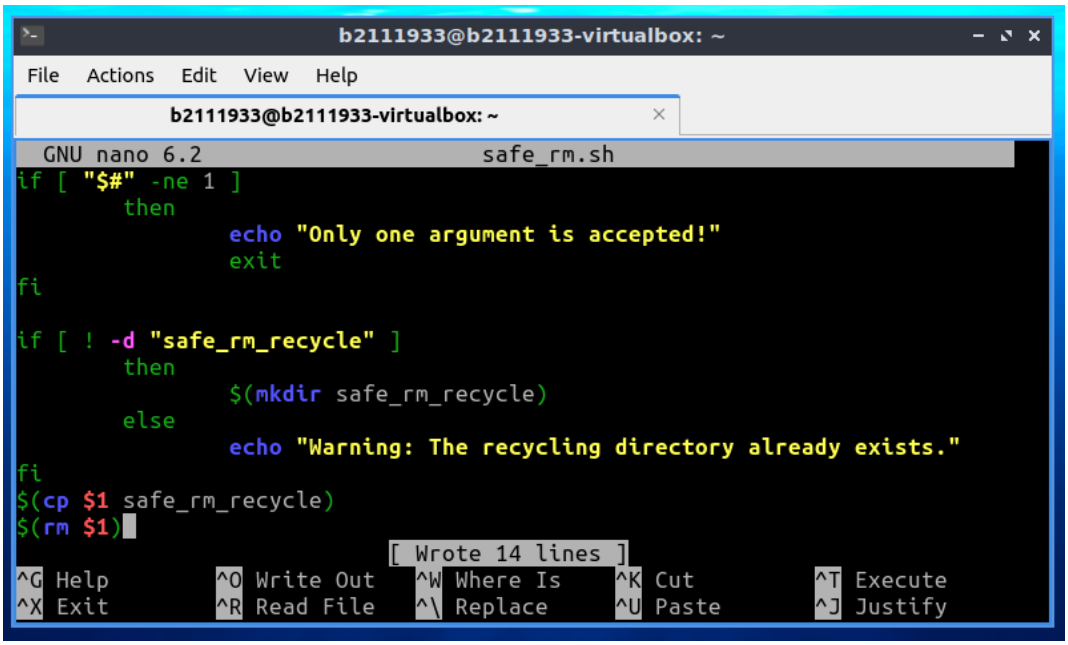
```
$nano safe_rm.sh
$sudo chmod 750 safe_rm.sh
```



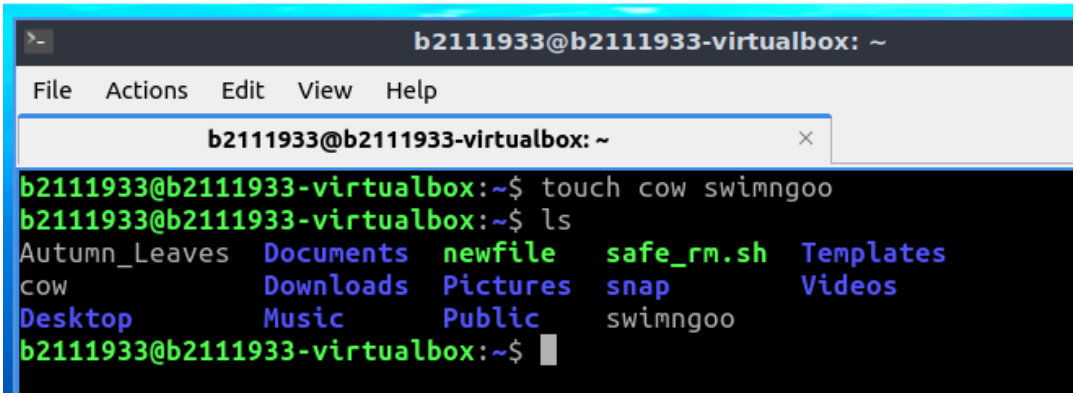
Created file **safe_rm.sh** with **nano** and changed its **permissions** to **-rwxr-x---**

safe_rm.sh

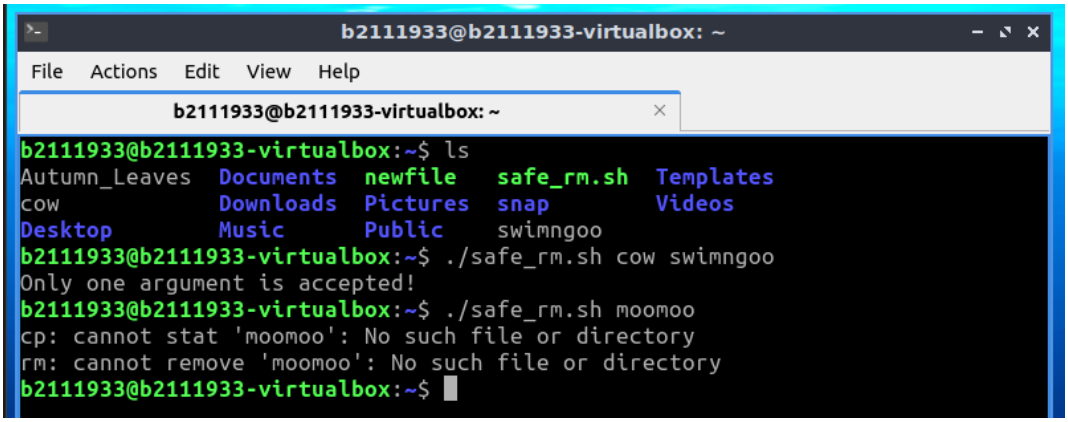
```
if [ “$#” -ne 1 ]
          then
                echo “Only one argument is accepted!”
                exit
fi

if [ ! -d “safe_rm_recycle” ]
          then
                $(mkdir safe_rm_recycle)
          else
               echo “Warning: The recycling
               directory already exists.”
fi
$(cp $1 safe_rm_recycle)
$(rm $1)
```



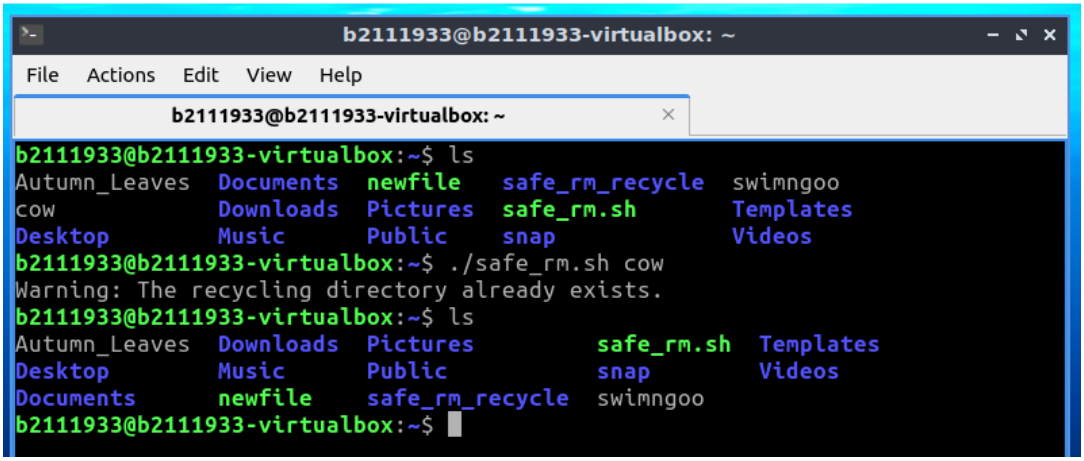
Contents of **safe_rm.sh**



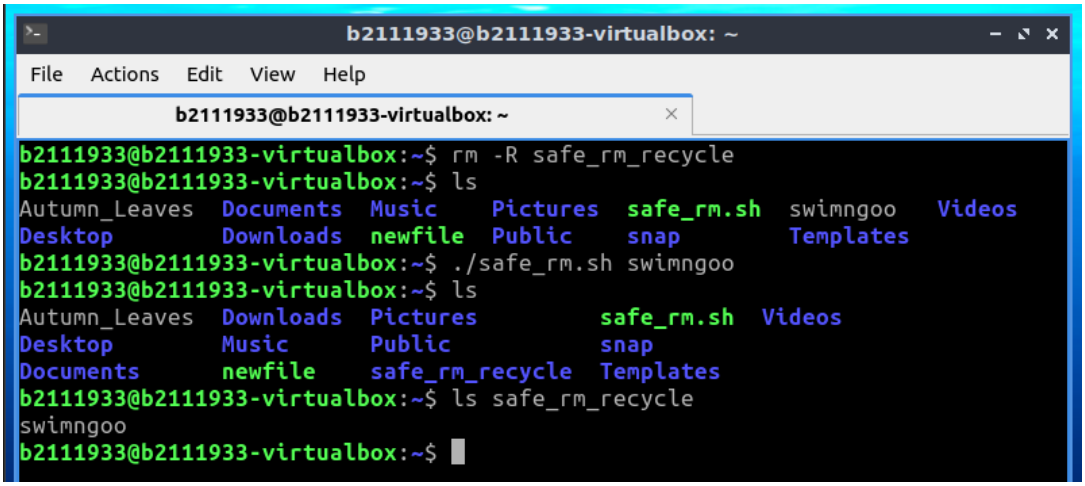
Created 2 files named “**cow**”& “**swimngoo**” for testing



Testing



Testing



Testing

Run the script, then take screenshots to show that you finish this exercise. Please also take screenshots of your code.

---END---