

# Individual Assessment Coversheet

To be attached to the front of the assessment.

**Campus:** Pretoria

**Faculty:** Information Technology

**Module Code:** ITLXA0-B44

**Group:** Group 1

**Lecturer's Name:** Mr. Kalonji. R.

**Student Full Name:** Keleboqile Nanikie Mathebula

**Student Number:** 1234567890

Indicate	Yes	No
Plagiarism report attached	x	

**Declaration:**

I declare that this assessment is my own original work except for source material explicitly acknowledged. I also declare that this assessment or any other of my original work related to it has not been previously, or is not being simultaneously, submitted for this or any other course. I am aware of the AI policy and acknowledge that I have not used any AI technology to generate or manipulate data, other than as permitted by the assessment instructions. I also declare that I am aware of the Institution's policy and regulations on honesty in academic work as set out in the Conditions of Enrolment, and of the disciplinary guidelines applicable to breaches of such policy and regulations.

<b>Signature:</b> KNP.	<b>Date:</b> 20/10/2025
------------------------	-------------------------

**Lecturer's Comments:**

<b>Marks Awarded:</b>	%
-----------------------	---

<b>Signature</b>	<b>Date</b>
------------------	-------------

---

## Table of Contents

---

Question 1.....	2
1.1. Install Oracle VM software and create two virtual machines using Oracle VirtualBox.....	2
1.2. Inside your home directory, create two-directories named Finance and HR. .....	12
1.3. Within the <b>Finance</b> directory, create a text file named <b>Expenses</b> . .....	13
1.4. Sort the items in <b>Expenses</b> alphabetically and, in a single command, save the sorted output to a file named <b>Budget</b> inside the <b>HR</b> directory. .....	15
1.5. While still in the <b>Finance</b> directory, display the number of: .....	16
Question 2.....	17
Question 3.....	17
3.1. Check if the /archive/logs/directory exists; create it if it does not.....	17
3.2. Accept an <b>optional parameter</b> for the number of days (default: 7 days).....	18
3.3. Move all .log files older than the specified number of days to /archive/logs/.....	19
3.4. Generate log_cleanup_report.txt in your home directory containing:.....	20
3.5. Include <b>comments</b> explaining the steps in your script. .....	20
3.6. Students must <b>take screenshots</b> of: .....	21
AI Declaration.....	22
References.....	23
Bibliography .....	23

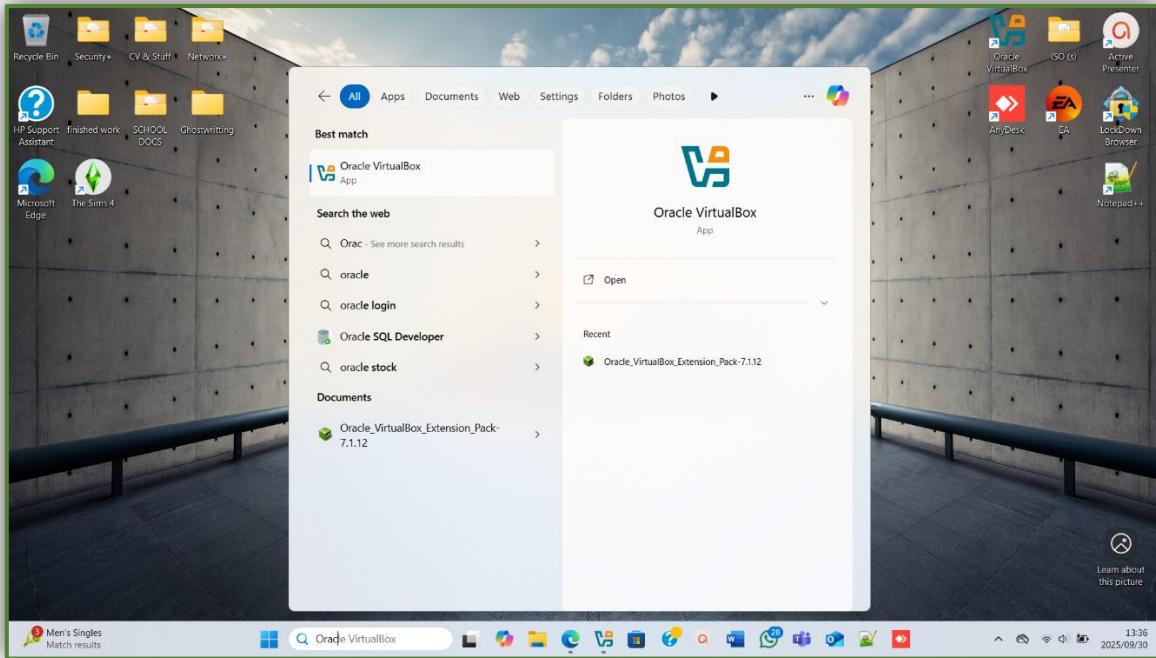
## Question 1

**NB: ZOOM FOR CLEAR VIEW!!**

### 1.1. Install Oracle VM software and create two virtual machines using Oracle VirtualBox.

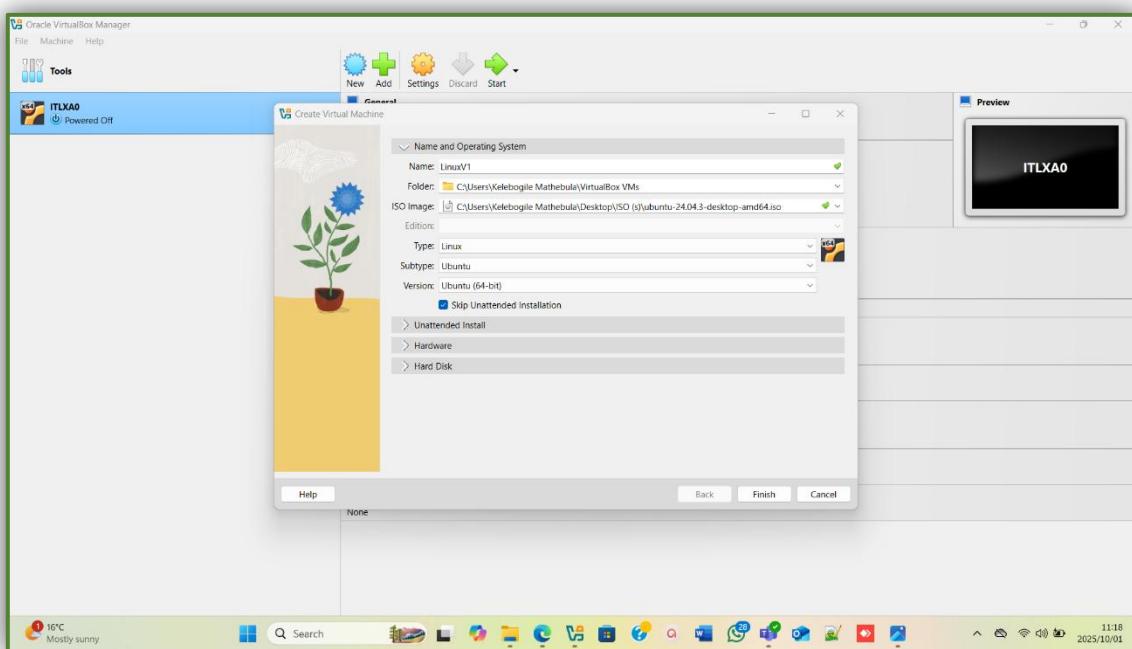
#### VM1:

The following image shows that Oracle VirtualBox was successfully installed.

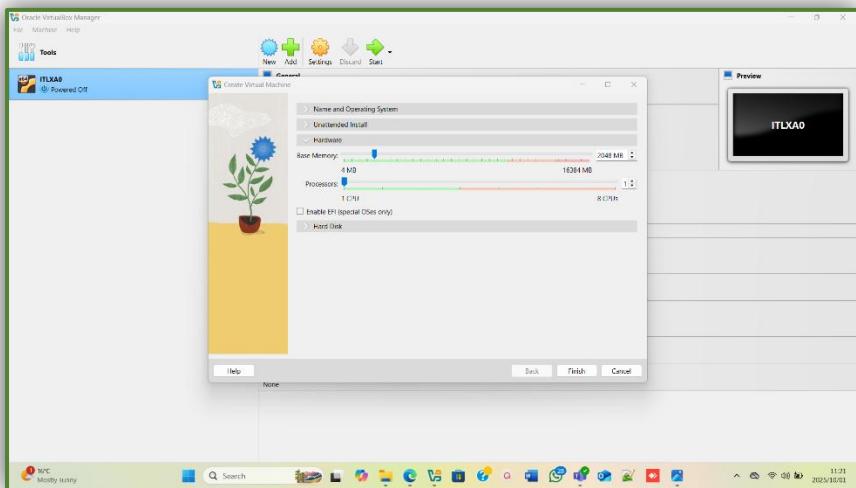


➤ Name: LinuxV1.

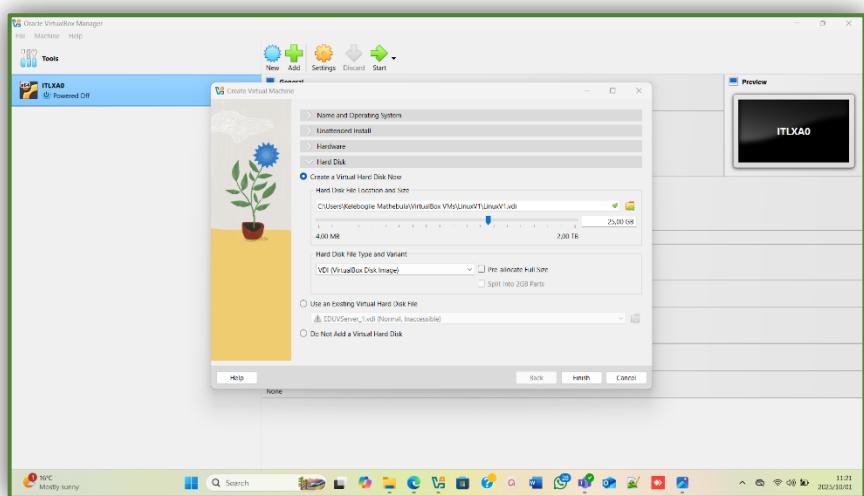
The following images show the creation of VM1 including what the specifications said I should name it as well as the disk size.



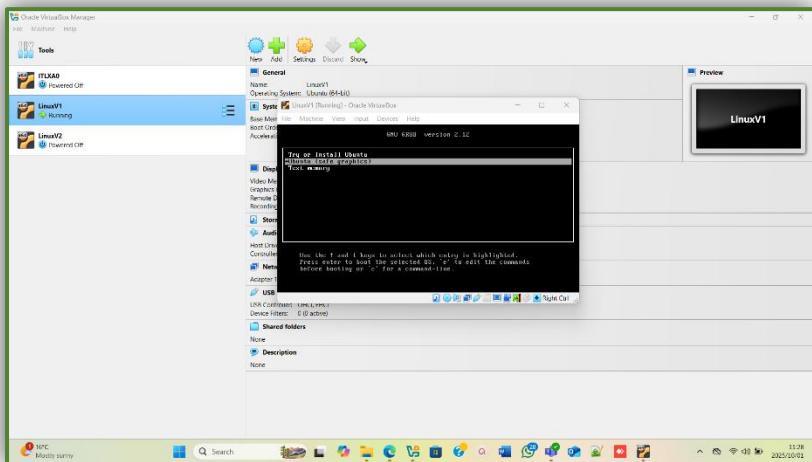
I left the memory as its default value.



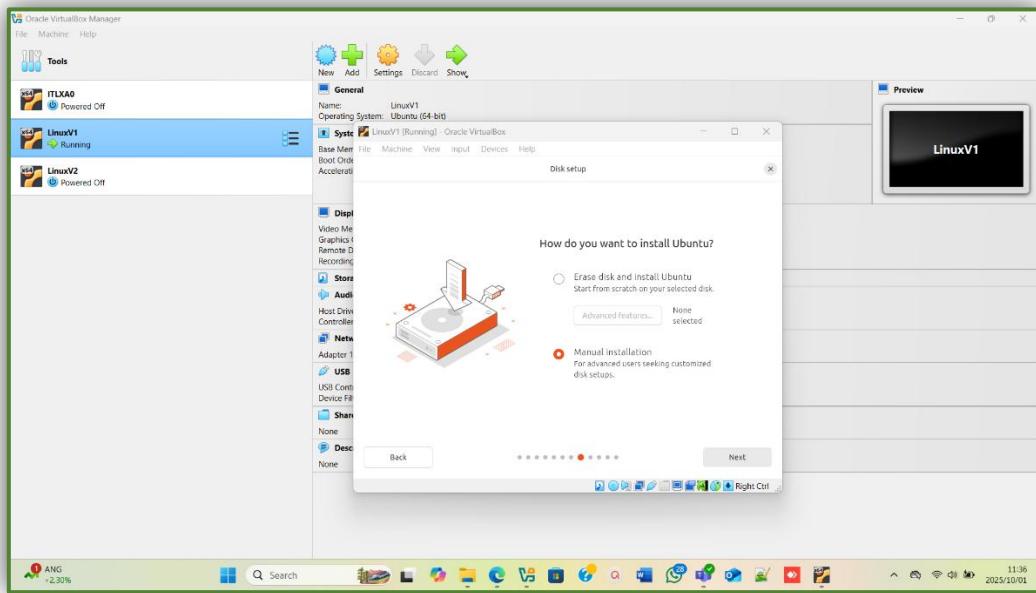
I then allocated the disk size which according to the specifications is 25GB.



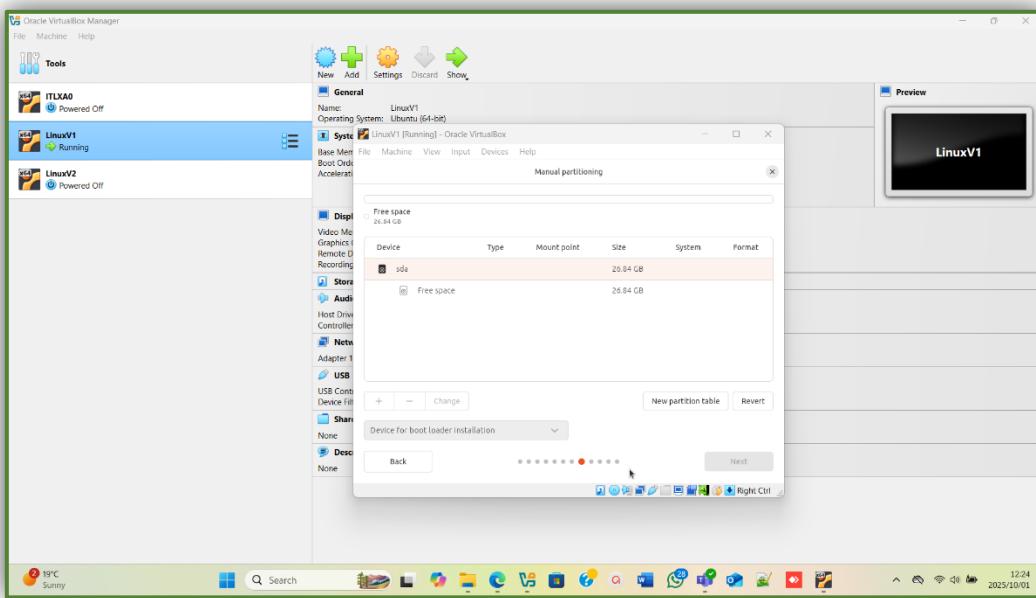
The following image shows the choices that appear once I started the VM. I choose “Ubuntu (safe graphics)” just so that I can bypass potential graphics card compatibility issues.



Once that I was done I then started the installation of the Ubuntu OS on the VM. I choose an Interactive installation; I also installed the recommended proprietary software. For the disk setup I went with a manual installation. The following images show the steps from the manual installation for the disk setup. (Programmer, 2024)

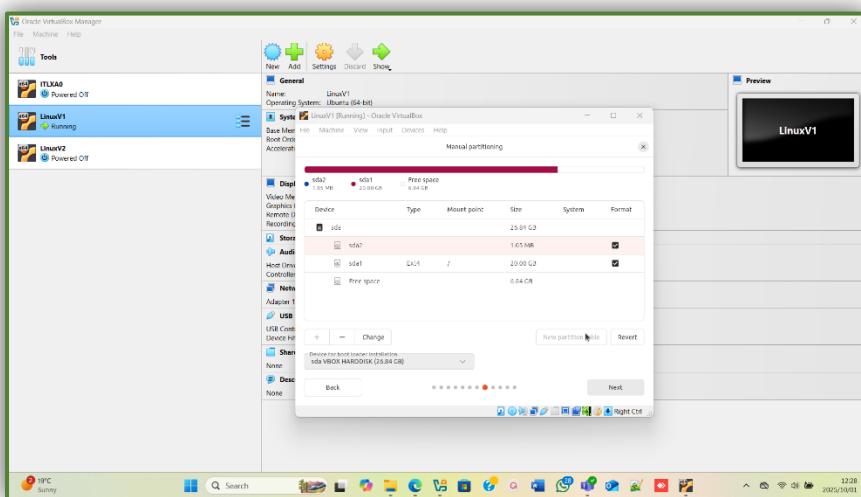
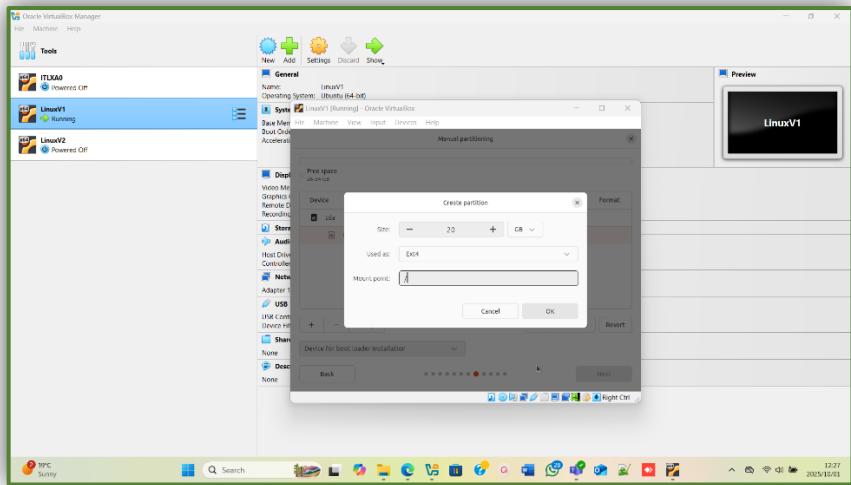


I then started the partition.



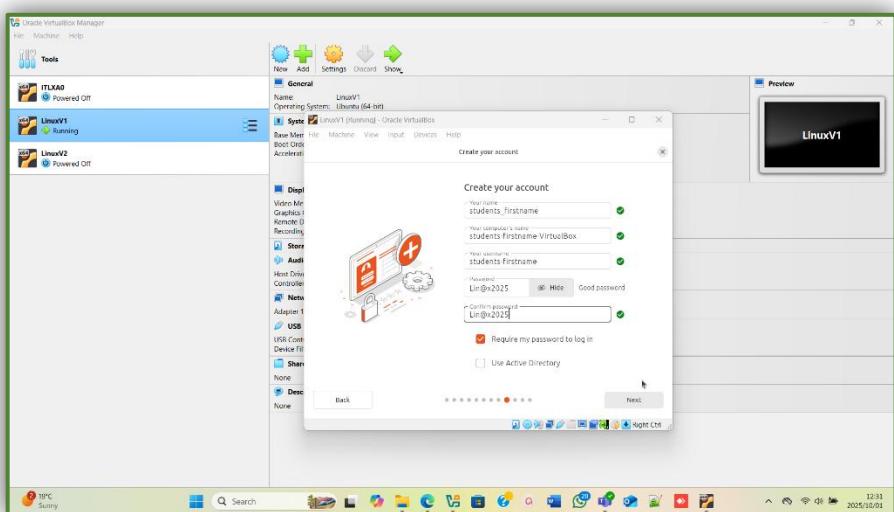
#### ➤ Disk: 25GB (20GB partition for OS).

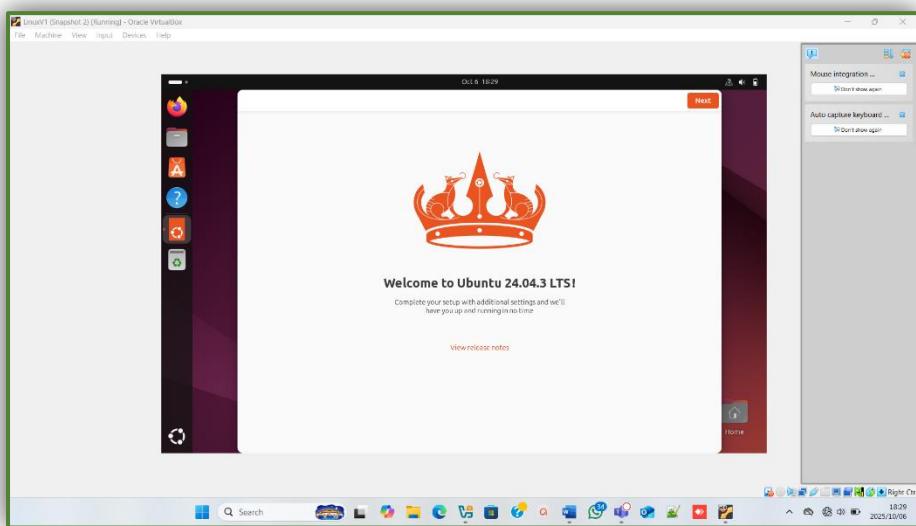
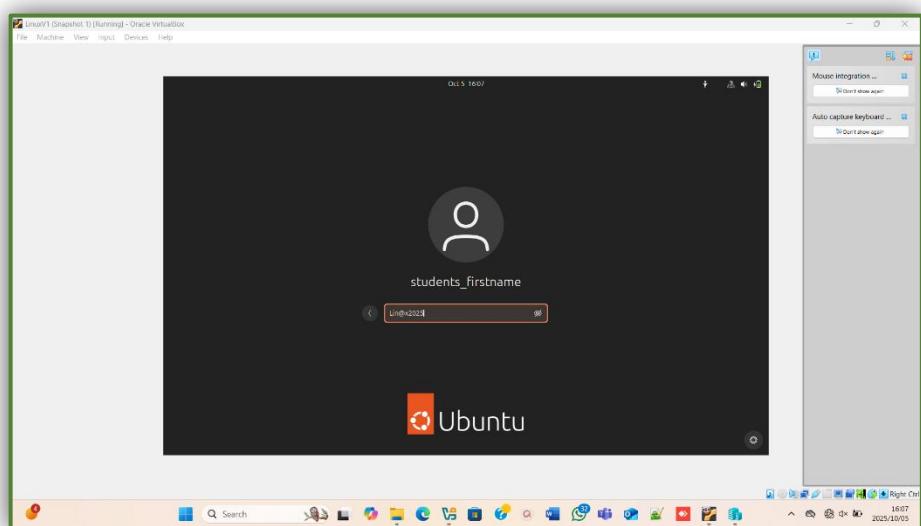
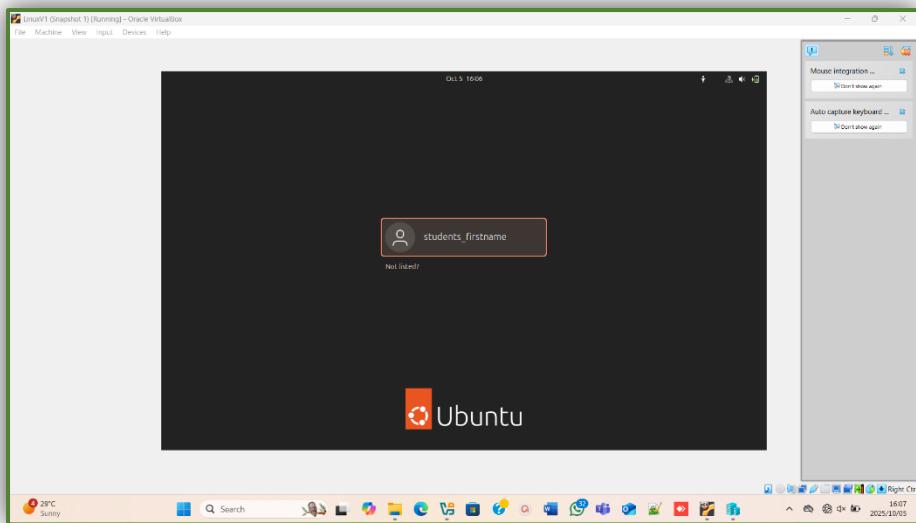
I choose the mount point to be root (/) since it is the central anchor of a Linux file system, acting as the topmost directory and the origin from which the entire directory tree branches. It controls all other directories and files for the operating system. It is also essential for the system to boot.



- Admin username: "students\_firstname".
- Password: Lin@x2025.

The following image shows the admin username as well as the password which concludes the installation. The next three images after the first image show that the Ubuntu OS was indeed installed.

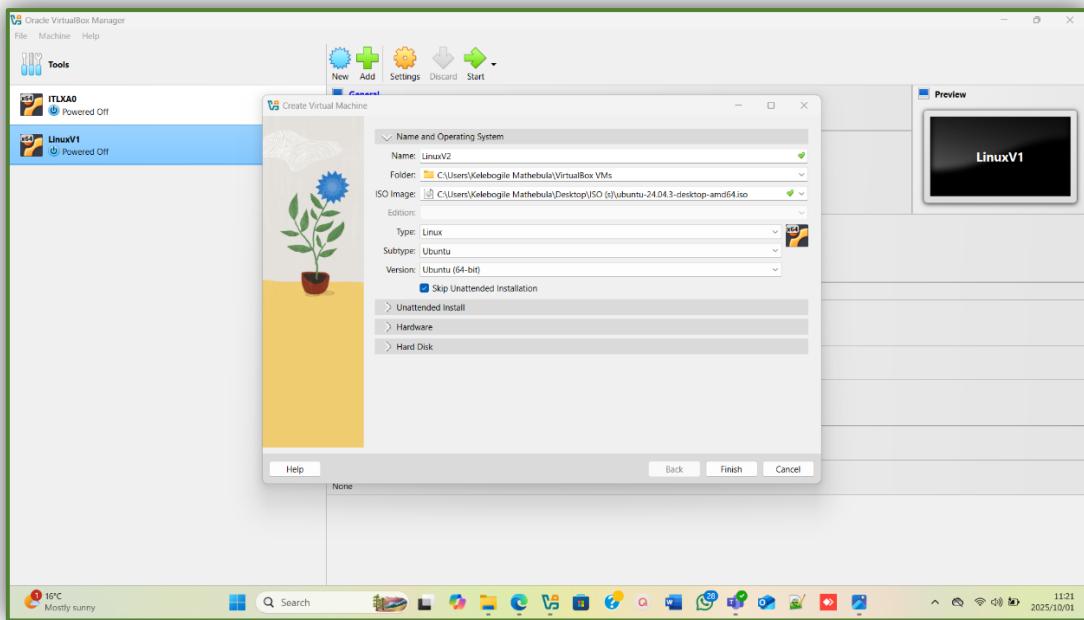




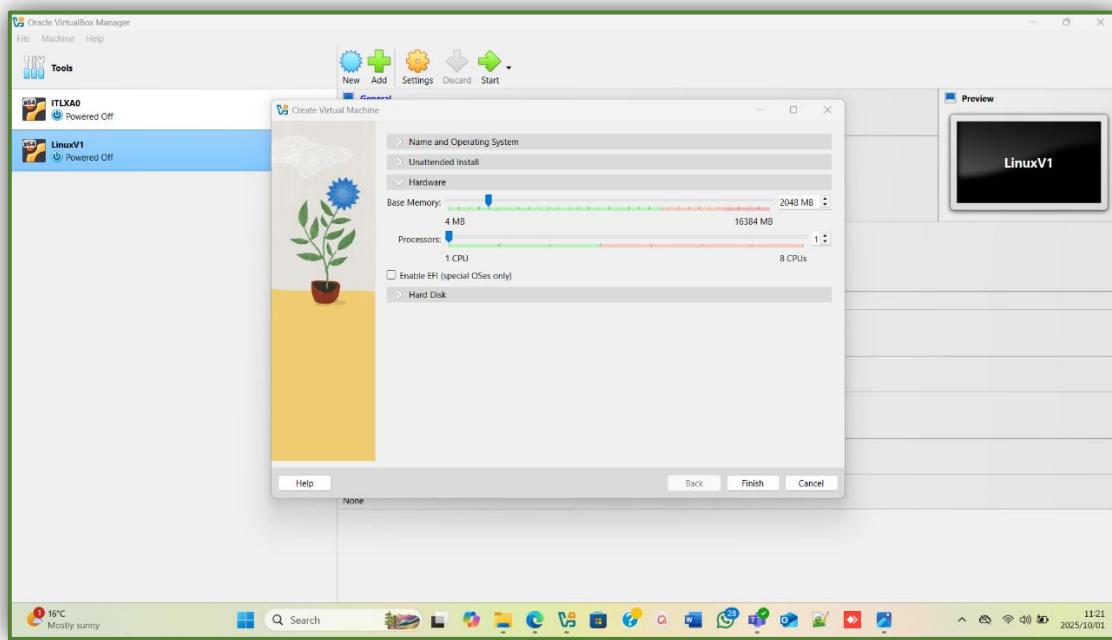
## VM2:

- Name: LinuxV2.

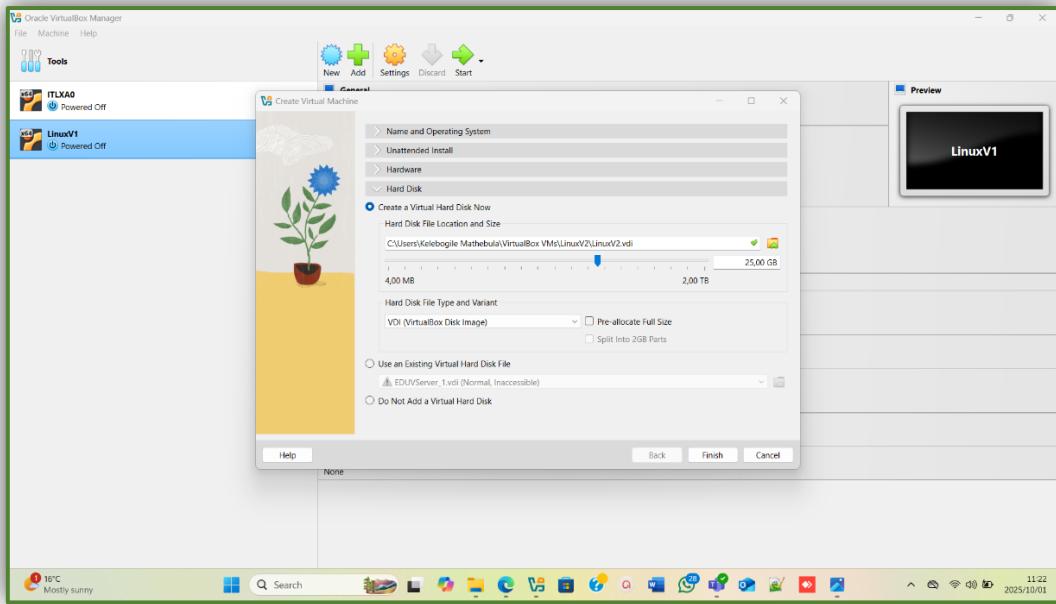
The following images show the creation of VM2 including what the specifications said I should name it as well as the disk size and memory.



I left the memory as the default value given.

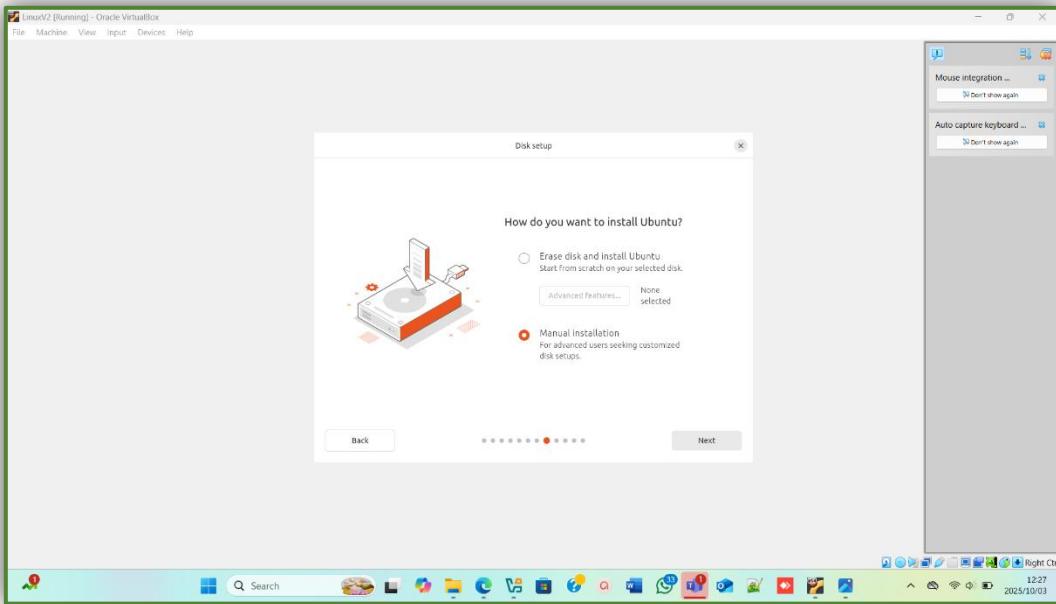


I then allocated the disk size according to the given specifications.

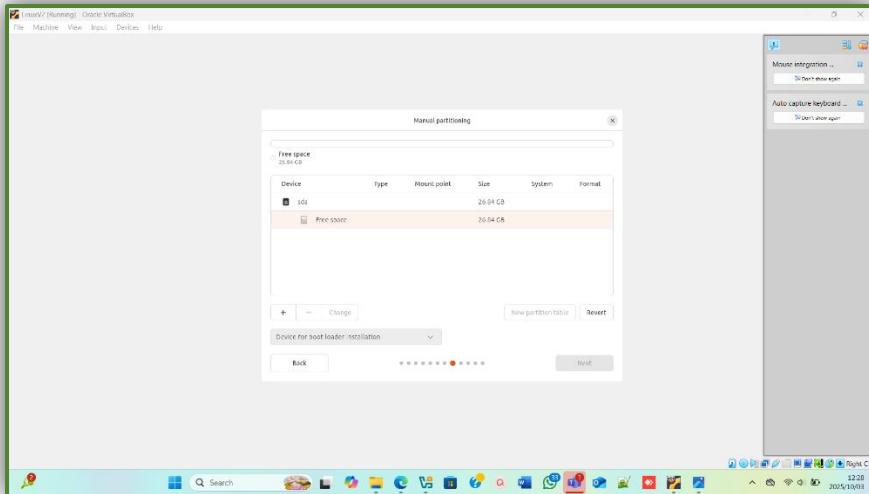


Once I started the VM there appeared a selection like I mentioned above where I choose “Ubuntu (safe graphics)” so that I am able to bypass potential graphics card compatibility issues as I already mentioned.

After I made the selection, I then went ahead and started the installation of the Ubuntu OS on the VM. I choose an Interactive installation; I also installed the recommended proprietary software. For the disk setup I went with a manual installation. The following images show the steps from the manual installation for the disk setup.

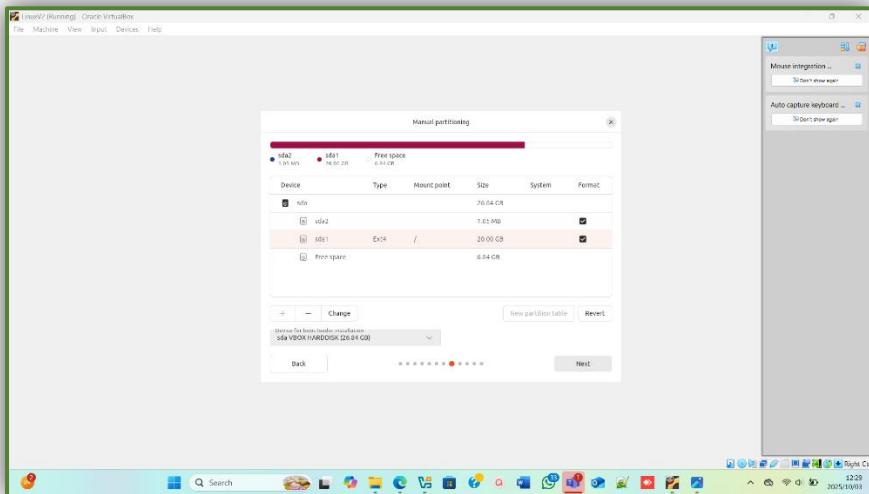
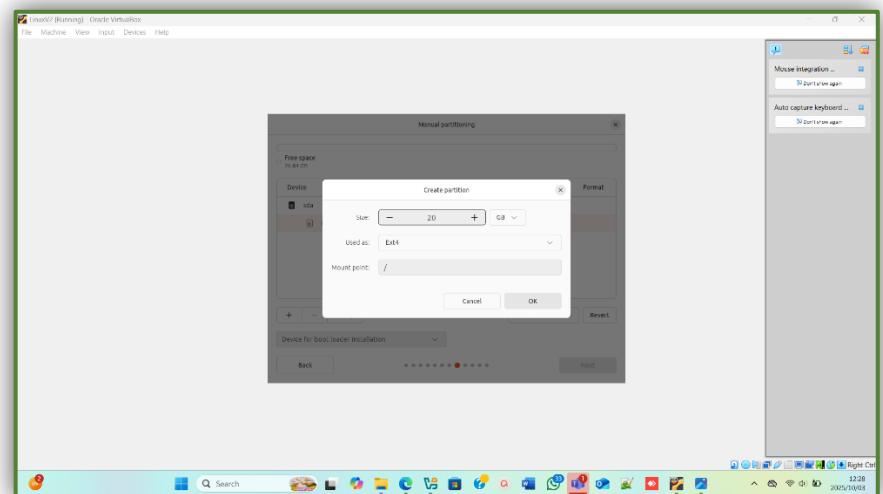


I then started the partition.

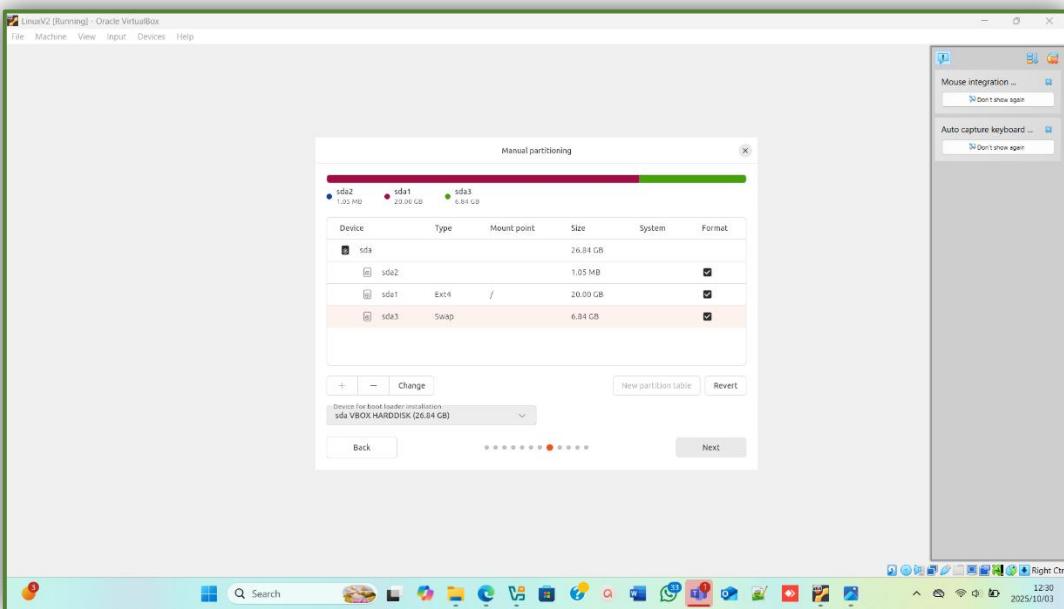
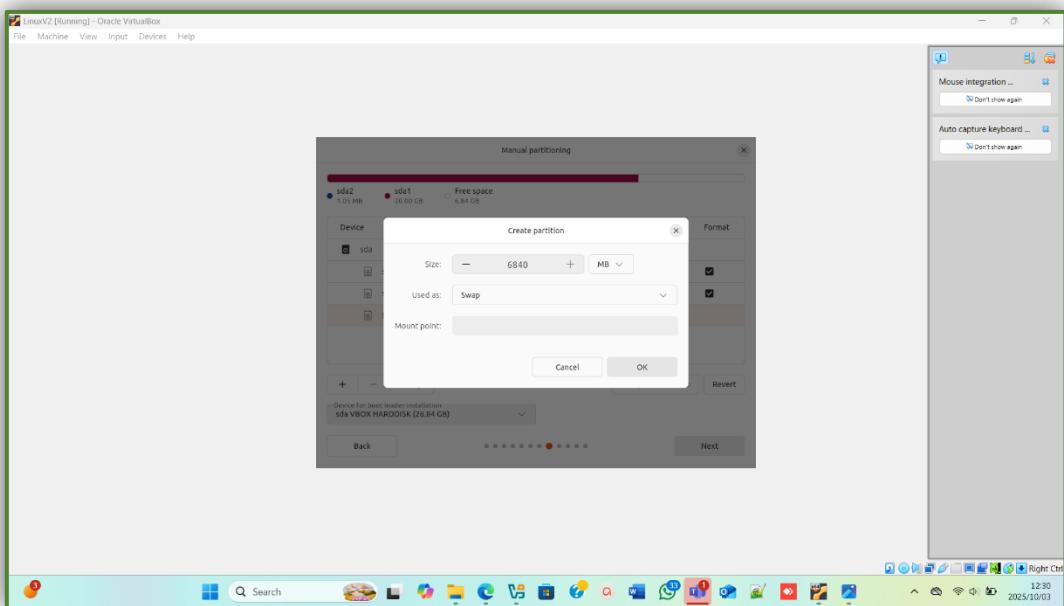


## ➤ Disk: 25GB (20GB partition for OS).

I did the same thing above with making the mount point for the 20GB root (/) and I explained above as to why I made the mount point root (/). For this VM I made the remaining space swap.

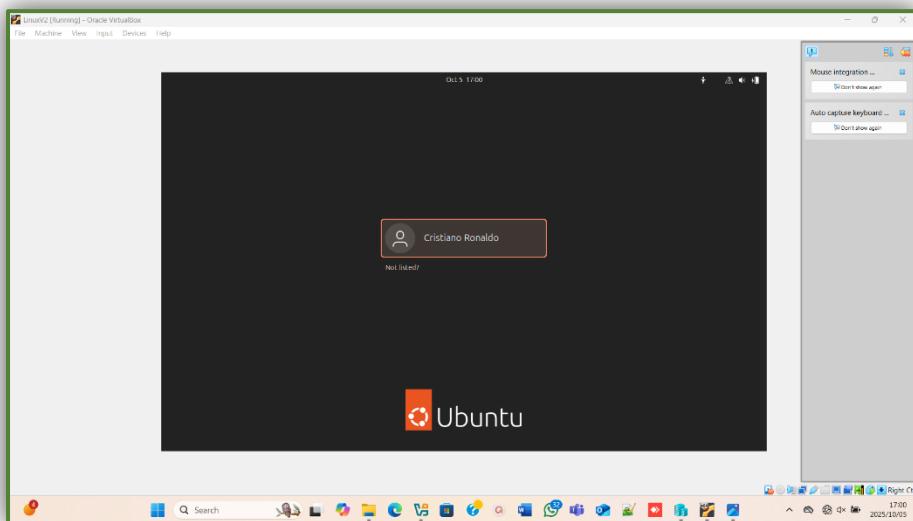
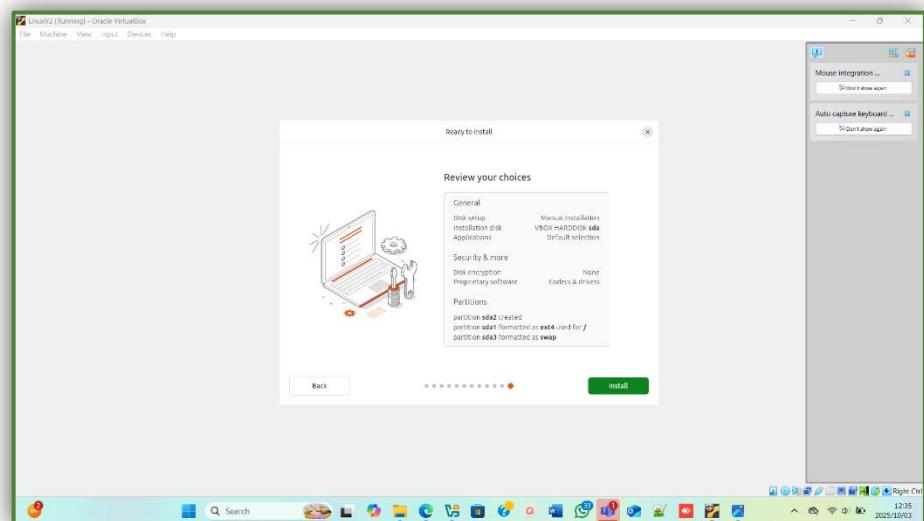
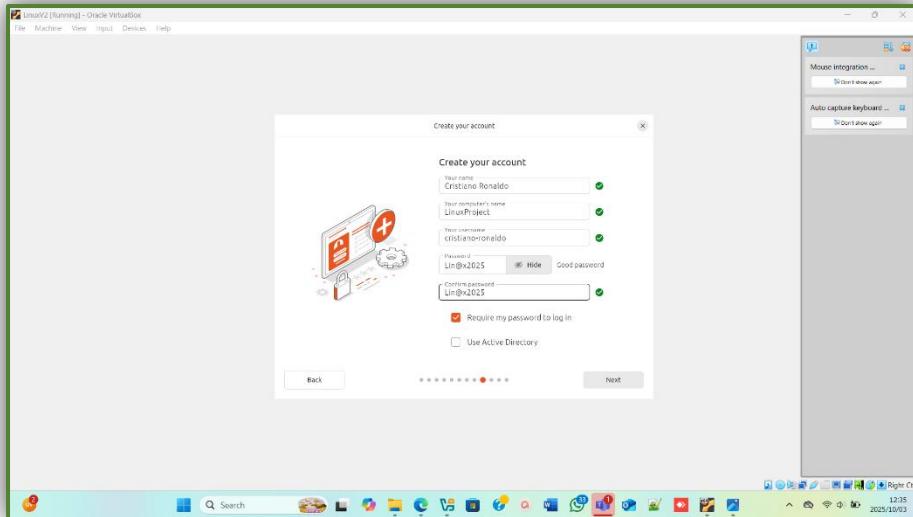


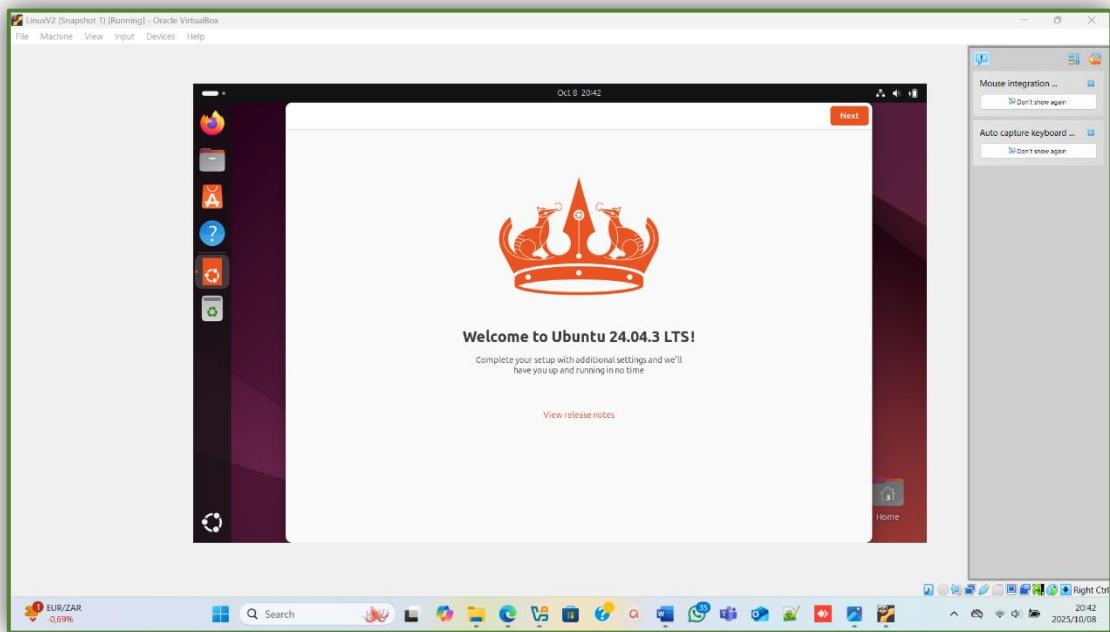
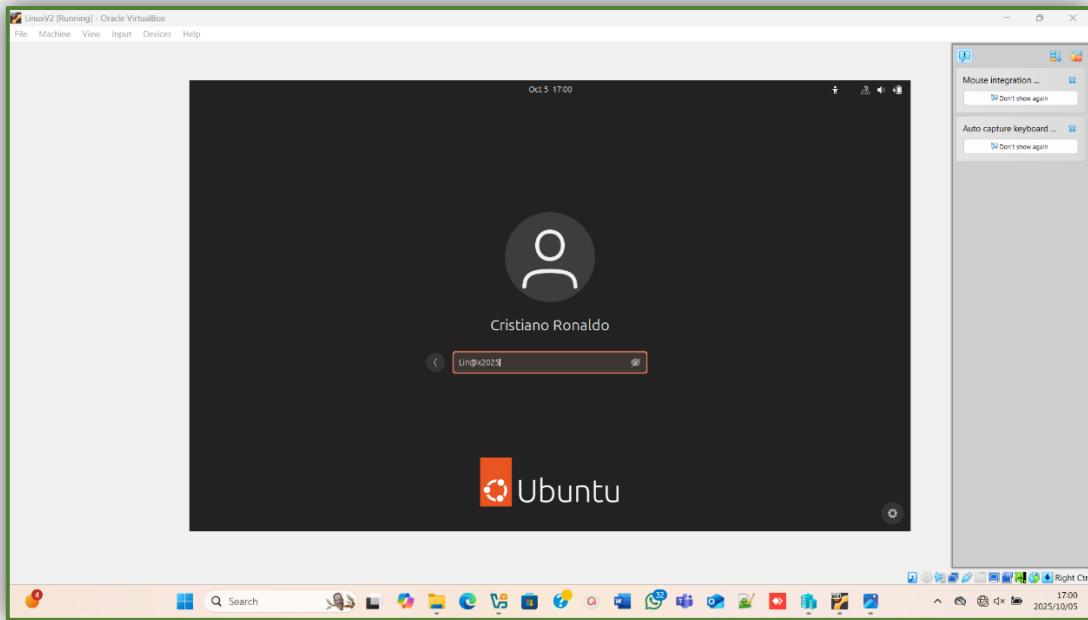
The following image shows the space I used as swap, the use or function of a swap partition is to provide virtual memory, acting as an extension of the physical RAM since it'll store inactive data on a hard drive. This helps the system to not run out of memory as well as crashing.



- Admin username: Cristiano Ronaldo.
- Computer name: LinuxProject.
- Password: Lin@x2025.

The following image shows the admin username, computer name, and the password which concludes the OS installation. The four images after the first image shows the summary as well as the login phase proving that the installation was a success.

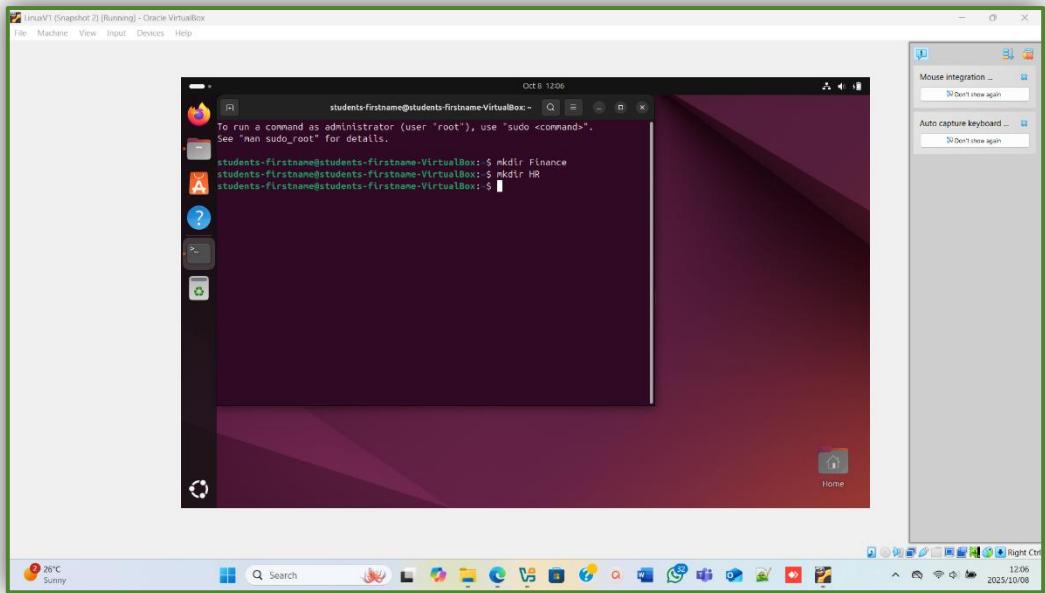




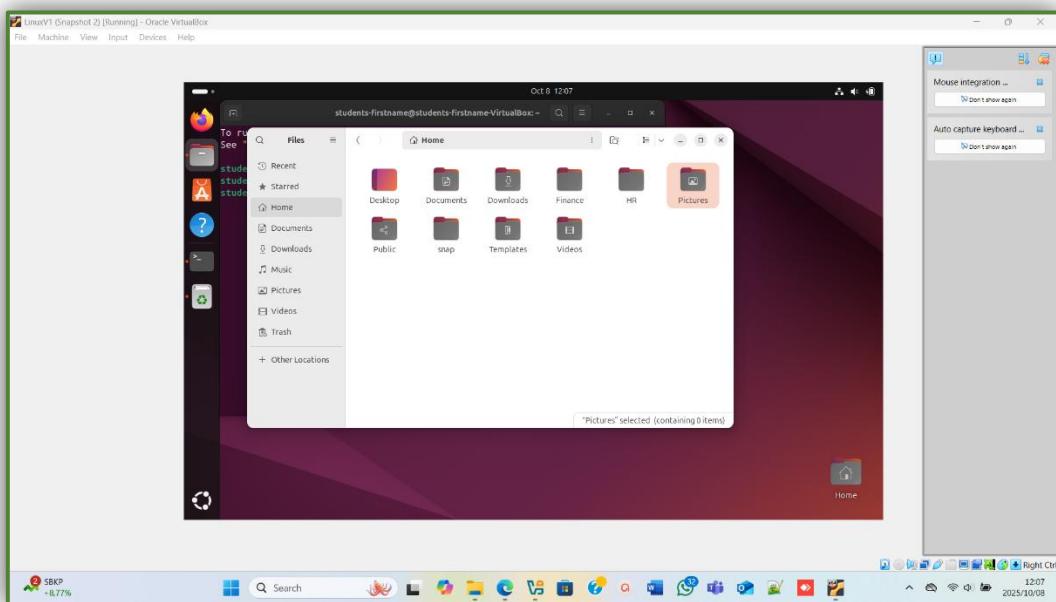
## 1.2. Inside your home directory, create two-directories named Finance and HR.

The home directory in file(s) acts as the parent directory for all individual user home directories which makes it the main directory. This setup creates a structured as well as organized file system where the home directory is a fundamental component that organizes user data separately from system files. (Tech, 2022)

Because "home" is the main directory, I did not create it in terminal, I started with "Finance" then "HR". I used "**mkdir**" to make the subdirectories. I used mkdir because its primary function is to create new and empty directories.



- Ensure that both directories are visible when you list the contents of your home directory.



### 1.3. Within the **Finance** directory, create a text file named **Expenses**.

I used the command “**cd**” to access the “Finance” folder, I basically moved from the home directory to the finance directory, the “**cd**” command allows you to navigate the file systems just so that you can work with files and directories in different locations.

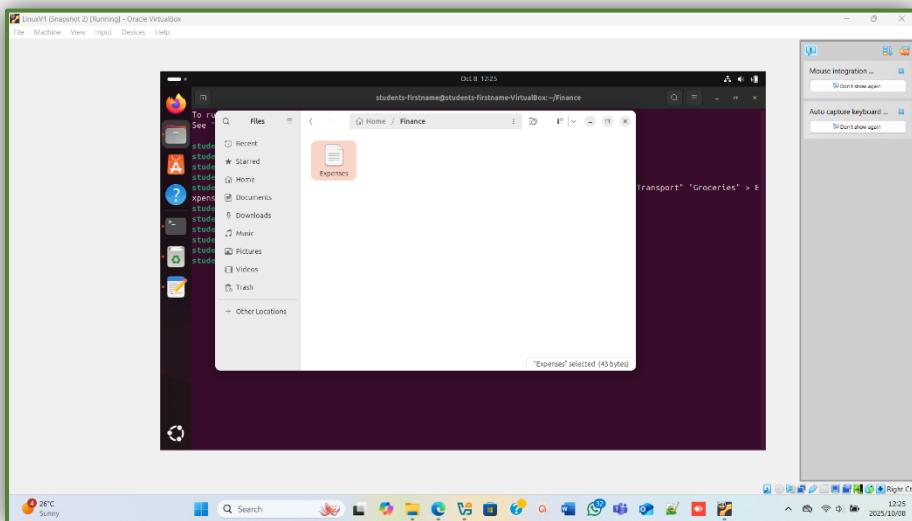
Once I accessed the Finance folder/directory, I made use of “**touch**” to create a file within the finance folder/directory. The “**touch**” command was then followed by the “**echo**” command. The “**echo**” command is used to write content to files, display variables, it can also be used for outputting messages.

The screenshot shows a Linux desktop environment. In the foreground, a terminal window titled 'Finance' is open, displaying a command-line session where the user creates a directory 'Finance', creates a file 'Expenses', and adds several expense items like 'MIFI', 'Insurance', 'Electricity', 'Water', and 'Groceries' to it. In the background, a file manager window titled 'Files' is open, showing a folder structure under 'Home / Finance'. The desktop bar at the bottom includes icons for various applications and a weather widget showing '26°C Sunny'.

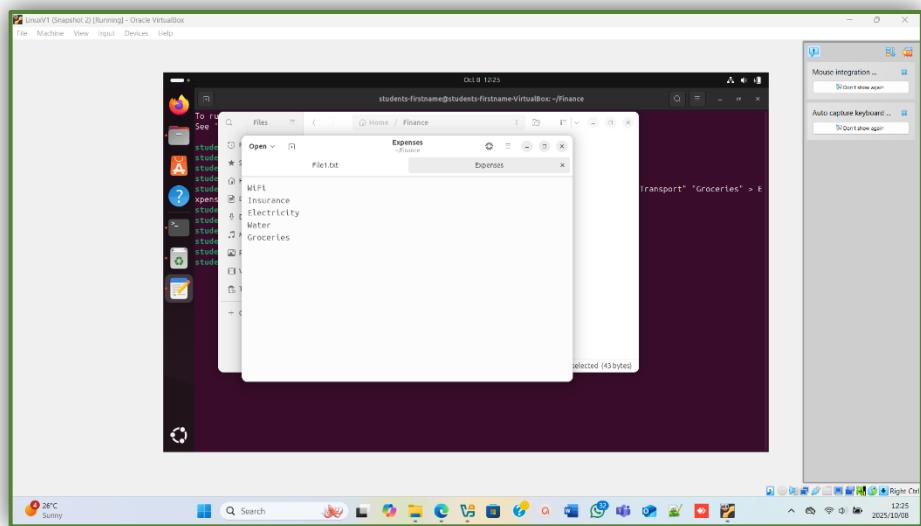
```

students-firstname@students-firstname-VirtualBox:~$ mkdir Finance
students-firstname@students-firstname-VirtualBox:~$ cd Finance
students-firstname@students-firstname-VirtualBox:~/Finance$ touch Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "MIFI" > Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "Insurance" >> Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "Electricity" >> Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "Water" >> Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "Groceries" >> Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ cat Expenses

```



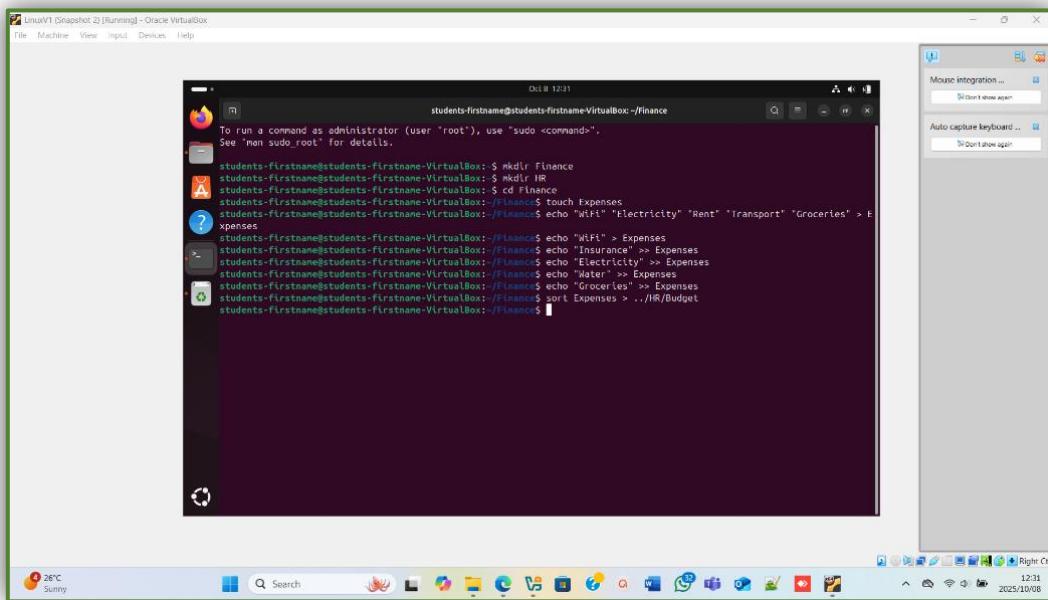
- Add at least five different expense items (e.g., rent, transport, groceries, etc.) to the file.



#### 1.4. Sort the items in Expenses alphabetically and, in a single command, save the sorted output to a file named **Budget** inside the **HR** directory.

I used the “**sort Expenses > ..//HR/Budget**” command. This command sorts the data in Expenses and then writes the sorted output into the file named budget, since the file was not there it created the file.

The following is how the command actually works: Shell will open “**..//HR/Budget**” for writing and that will create the file if it does not exist, the “**sort**” command reads the file Expenses and outputs the sorted lines; because of the greater than sign (**>**), that is then written to the Budget file. The Expenses file is not changed. (TV, 2024)

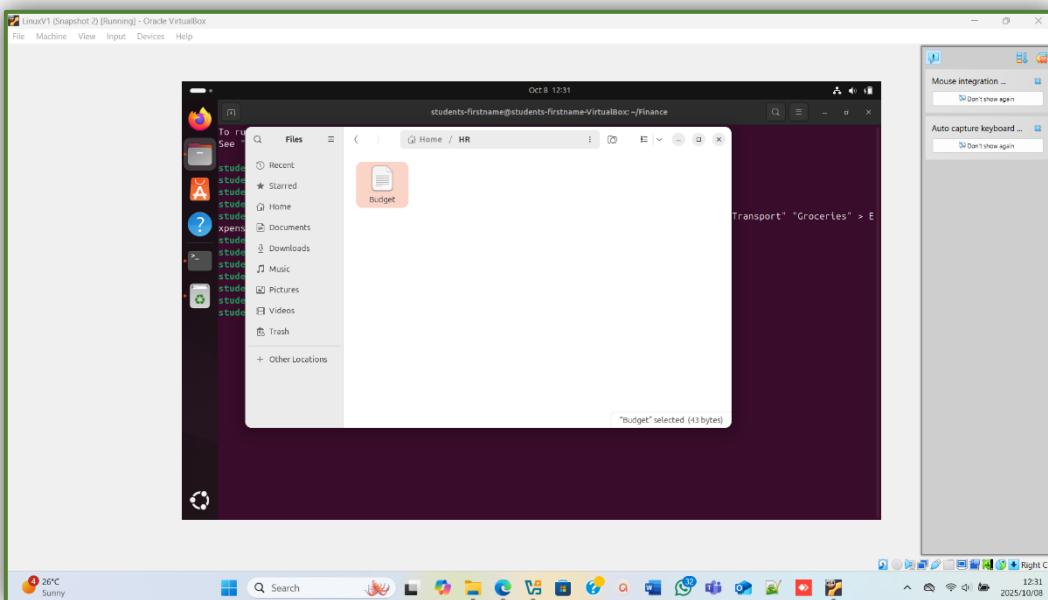


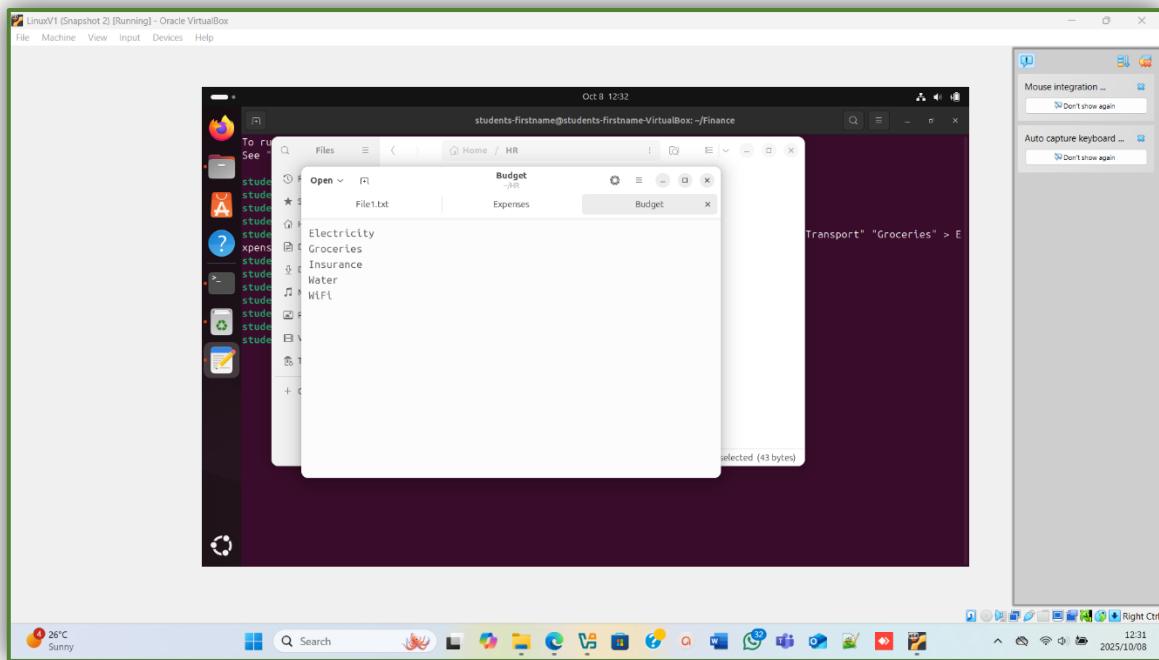
A screenshot of a Linux terminal window titled "students-firstname@students-firstname-VirtualBox: ~ /Finance". The terminal shows the following command sequence:

```
students-firstname@students-firstname-VirtualBox: ~ /Finance
To run a command as administrator (user 'root'), use "sudo <command>".
See 'man sudo_root' for details.

students-firstname@students-firstname-VirtualBox: $ mkdir Finance
students-firstname@students-firstname-VirtualBox: $ mkdir HR
students-firstname@students-firstname-VirtualBox: $ cd Finance
students-firstname@students-firstname-VirtualBox: ~/Finance$ touch Expenses
students-firstname@students-firstname-VirtualBox: ~/Finance$ echo "WiFi" "Electricity" "Rent" "Transport" "Groceries" > Expenses
students-firstname@students-firstname-VirtualBox: ~/Finance$ echo "Water" >> Expenses
students-firstname@students-firstname-VirtualBox: ~/Finance$ echo "Insurance" >> Expenses
students-firstname@students-firstname-VirtualBox: ~/Finance$ echo "Electricity" >> Expenses
students-firstname@students-firstname-VirtualBox: ~/Finance$ sort Expenses > ..//HR/Budget
students-firstname@students-firstname-VirtualBox: ~/Finance$
```

- Verify that the **Budget** file exists and contains the sorted list.





### 1.5. While still in the **Finance** directory, display the number of:

- Lines.
- Words.
- Characters contained in the **Expenses** file.

I made use of the command “**wc -l**” because it counts how many lines are in the file, “**wc**” means **word count**, as well as the command “**wc -w**” because it counts the words that are in a file. Lastly the command “**wc -m**” since it counts characters that are in the file. The **-l** stands for lines, the **-w** stands words and lastly **-m** stands for characters (the “**m**” is for **multibyte characters**).

```
LinuxV1 (Snapshot 2) [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Oct 8 12:36
students-firstname@students-firstname-VirtualBox:~/Finance
See 'man sudo_root' for details.

students-firstname@students-firstname-VirtualBox: $ mkdir Finance
students-firstname@students-firstname-VirtualBox: $ mkdir HR
students-firstname@students-firstname-VirtualBox: $ cd Finance
students-firstname@students-firstname-VirtualBox:~/Finance$ touch Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "WiFi" "Electricity" "Rent" "Transport" "Groceries" > Expenses
xpenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "WiFi" > Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "Insurance" >> Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "Electricity" >> Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ echo "Water" >> Expenses
students-firstname@students-firstname-VirtualBox:~/Finance$ sort Expenses > ../HR/Budget
students-firstname@students-firstname-VirtualBox:~/Finance$ wc -l Expenses
5 Expenses
students-firstname@students-Firstname-VirtualBox:~/Finance$ wc -w Expenses
5 Expenses
students-firstname@students-Firstname-VirtualBox:~/Finance$ wc -m Expenses
43 Expenses
students-firstname@students-Firstname-VirtualBox:~/Finance$
```

## Question 2

For question two I was given a scenario whereby I had to recorded a 100MB video based on the scenario explaining how I would check the network interfaces on a Linux workstation. I described the command I would use (ip a or ifconfig) and how I would identify whether the workstation has a valid IP address. I then explained how I would test network connectivity using ping to the local loopback (127.0.0.1), the gateway/router, and an external IP. I described what different results would mean, for example, if the external IP works but domain names fail or if the gateway cannot be reached. I included the steps involved in examining and potentially rebuilding the initramfs how configuration in Linux.

I also had to explain how I would check the status of the network service using systemctl status NetworkManager and how I would restart it if necessary. I described why the service is important for network connectivity. Then, I explained how I would test DNS resolution using nslookup and checking the current DNS settings using cat/etc/resolv.conf. I described what I would do if the DNS is not functioning correctly.

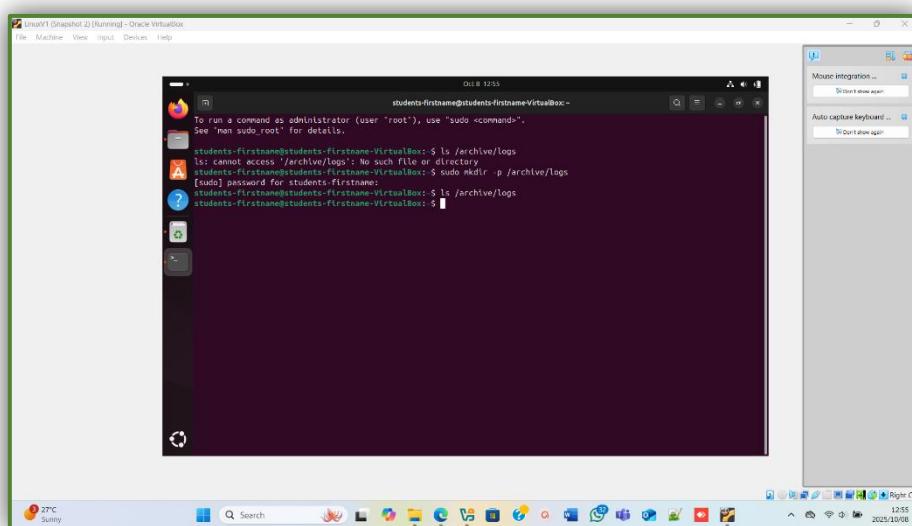
Lastly, I explained the steps I would take to fix a DNS problem, either by editing Netplan configuration files or using the NetworkManager GUI/command-line. I went on to describe how I would confirm that the DNS issue is resolved (e.g., successfully pinging a domain like google.com). I also summarized all the troubleshooting steps I would take to restore internet connectivity on the workstation.

## Question 3

### 3.1. Check if the /archive/logs/directory exists; create it if it does not.

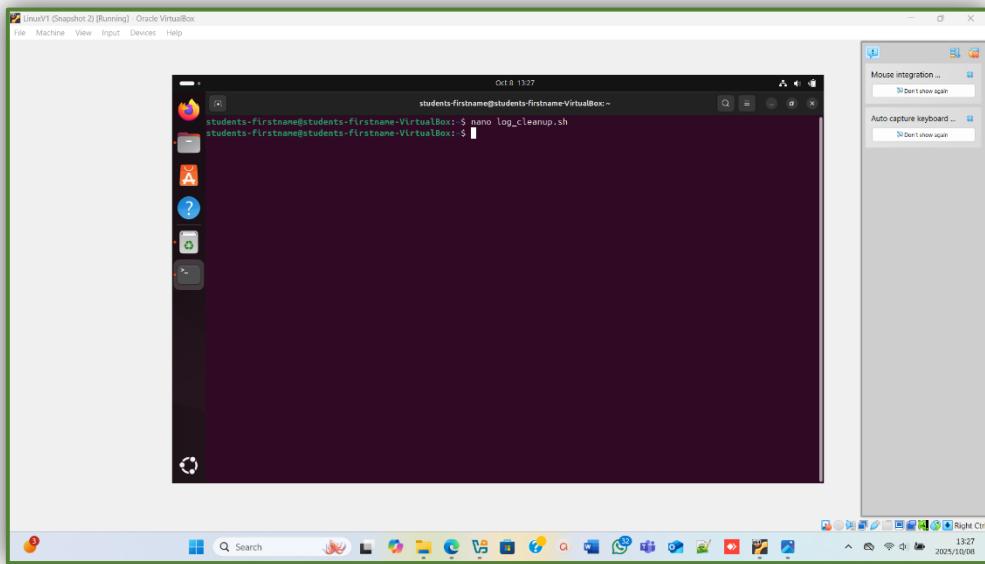
I made use of the “ls” command since it is used to list the files and folders in a directory, I also used the command “sudo mkdir -p”, mkdir is used to create a directory/folder while “sudo” means **superuser do** and I made use of sudo since I required system-level permission; archive is a folder directly under the root (/) and inside it there’s another file named logs. The **slash “/”** represents an absolute path which means it starts from the root directory of the entire filesystem. Lastly, I made use of the command “-p”, the -p command simply means create parent directories if they do not exist.

I then re used the ls command to see if the directory exists which it did after the sudo mkdir -p command.

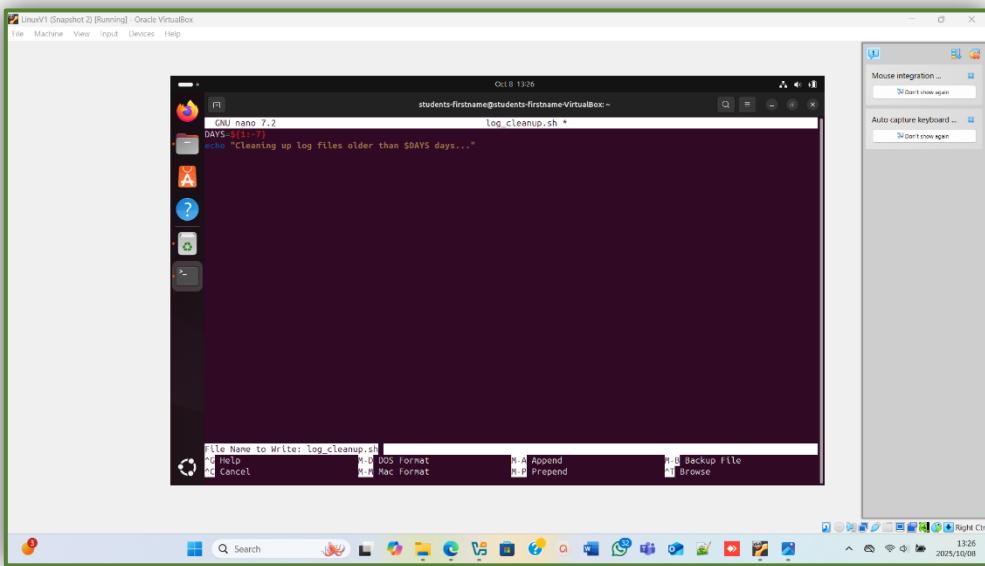


### 3.2. Accept an optional parameter for the number of days (default: 7 days).

The following code is used by Ubuntu to open a small text editor inside the terminal window. It opens the script, “**nano**” is a text editor and “**log\_cleanup.sh**” is the name of the file I’m editing.

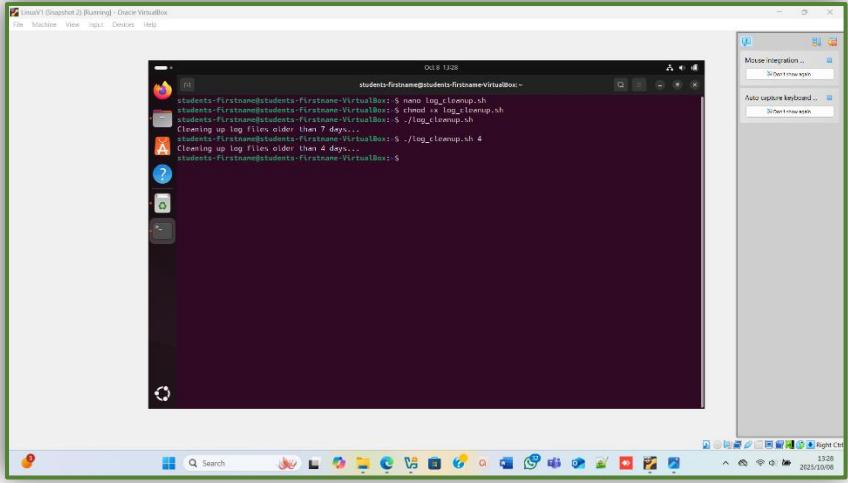


The following just sets the variable DAYS to whatever number I type when I run the script, but if I don’t type anything it then uses 7. (*Kindly visit question 3.5 for explanation*)



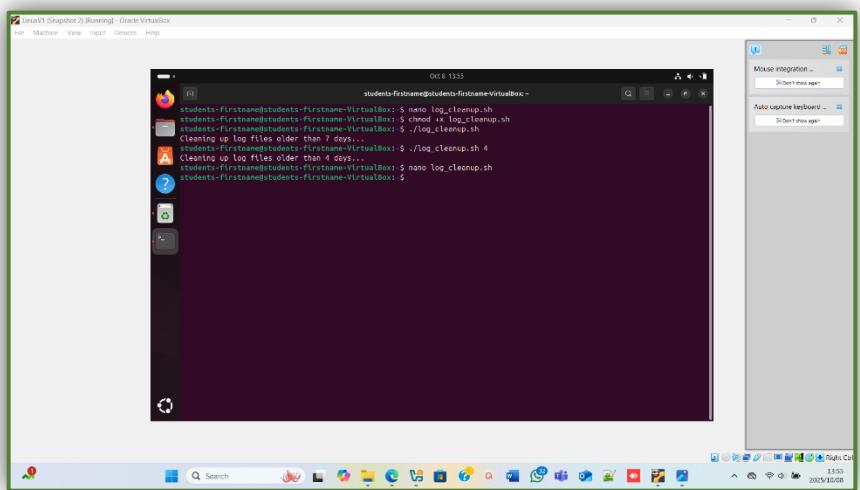
After I wrote what I wrote in the script I ran the code “**chmod +x log\_cleanup.sh**”, **chmod** basically changes the mode meaning it changes the permissions of a file and the **+x** adds an executable permission basically making it runnable. Lastly, the **log\_cleanup.sh** is the script that I want to make executable. (ProgrammingKnowledge2, 2023)

The code I ran after basically runs my script, specifically the **“./”** means run the file named **log\_cleanup.sh** in the current directory.

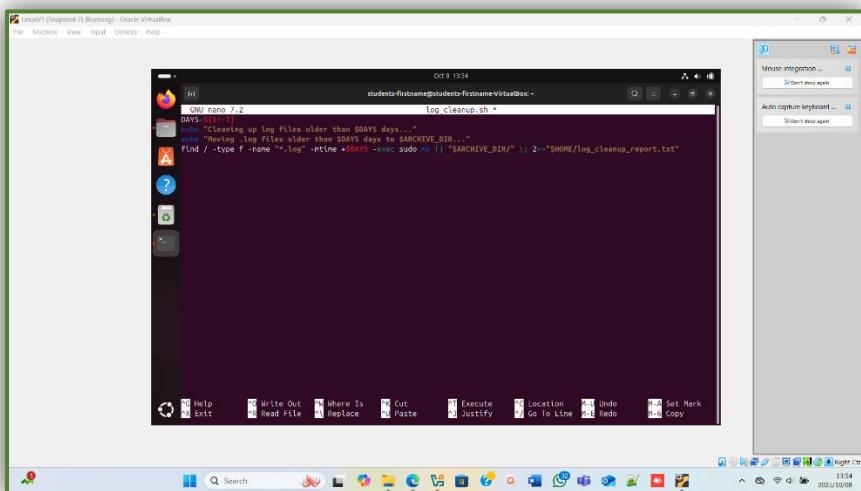


### 3.3. Move all .log files older than the specified number of days to /archive/logs/.

I first opened the script after running the code “./log\_cleanup.sh 4”.



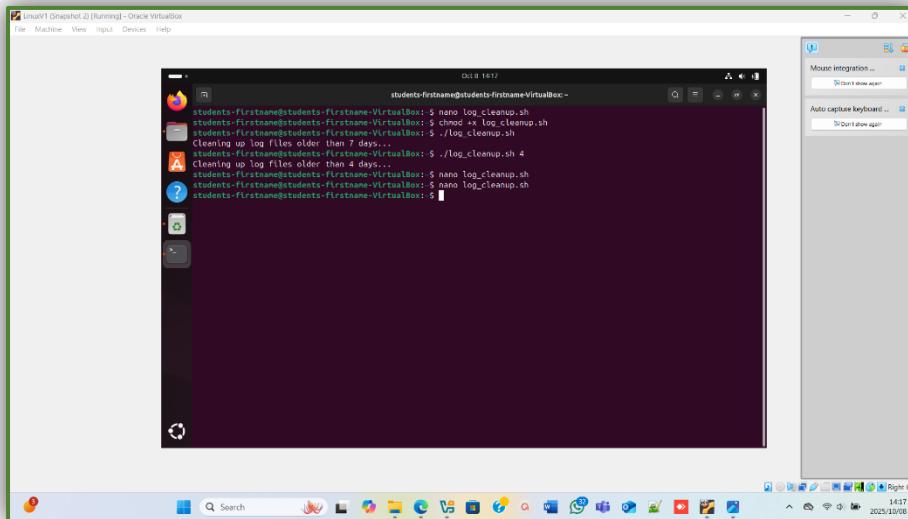
The following is what I then wrote in my script. I explain what I wrote in the script in question 3.5 of this project. (*Kindly visit question 3.5 for explanation*)



### 3.4. Generate log\_cleanup\_report.txt in your home directory containing:

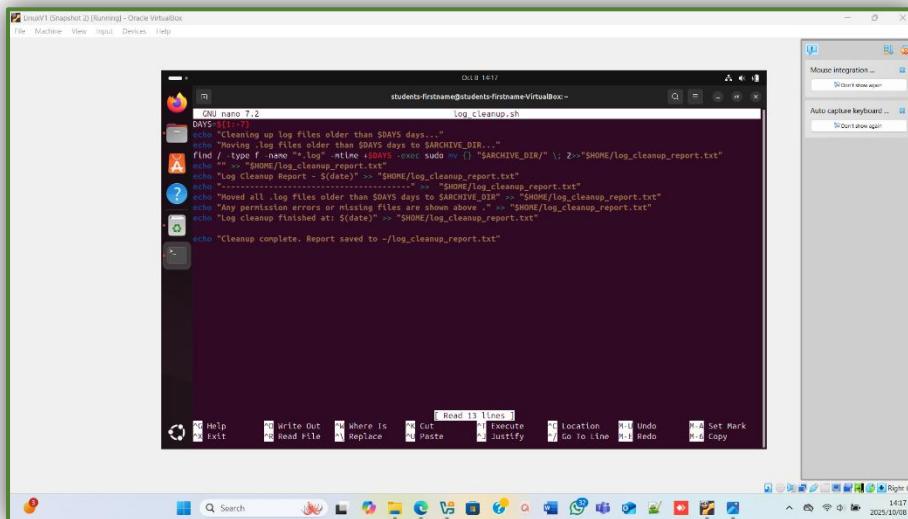
- Current date and time.
- List of files moved.
- Any errors or files that could not be moved.

I first opened the script.



```
DLB 1677
studentsfirstname@students-VirtualBox: ~
$ nano log_cleanup.sh
studentsfirstname@students-VirtualBox: ~
$ chmod +x log_cleanup.sh
studentsfirstname@students-VirtualBox: ~
$ ./log_cleanup.sh
Cleaning up log files older than 7 days...
A Cleaning up log files older than 4 days...
studentsfirstname@students-VirtualBox: ~
$ ./log_cleanup.sh 4
studentsfirstname@students-VirtualBox: ~
$ nano log_cleanup.sh
studentsfirstname@students-VirtualBox: ~
$ nano log_cleanup.sh
studentsfirstname@students-VirtualBox: ~
```

Like I said in question 3.3, kindly look at question 3.5 for explanations of what I wrote in my script.



```
DLB 1677
studentsfirstname@students-VirtualBox: ~
$ nano 7.2
studentsfirstname@students-VirtualBox: ~
$ log_cleanup.sh
DAYS=7
echo "Cleaning up log files older than $DAYS days..."
echo "Moving log files older than $DAYS days to $ARCHIVE_DIR..."
find . -type f -name ".log" -mtime +$DAYS -exec sudo mv {} "$ARCHIVE_DIR/" \; >> "$HOME/log_cleanup_report.txt"
A
echo "Log Cleanup Report at $date" >> "$HOME/log_cleanup_report.txt"
echo "-----" >> "$HOME/log_cleanup_report.txt"
echo "Moved all .log files older than $DAYS days to $ARCHIVE_DIR" >> "$HOME/log_cleanup_report.txt"
echo "Any permission errors or missing files are shown above." >> "$HOME/log_cleanup_report.txt"
echo "Log cleanup finished at $date" >> "$HOME/log_cleanup_report.txt"
echo "Cleanup complete. Report saved to ~/log_cleanup_report.txt"
```

### 3.5. Include **comments** explaining the steps in your script.

For line 1, which is on question 3.2, the line sets a variable called DAYS. It basically decides how long the log files should be before moving them and according to the question I used the default being 7 days.

For line 2, which is also on question 3.2, this basically prints a message on the screen to tell the user that the cleanup process has started and it will deal with files older than the chosen number of days.

For line 3, which is on question 3.3, this displays another message which says that the script will now move those old log files to the archive directory (/archive/logs/).

For line 4, which is on question 3.3, this line finds all old .log files and moves them to the archive folder, while saving any problems to the report.

For line 5, which is on question 3.4, it just adds an empty line to the report file to make it look neat and readable.

For line 6, which is on question 3.4, it adds a timestamp (current date and time) to the top of the report so you know when the cleanup happened.

For line 7, which is on question 3.4, it adds a divider line in the report to separate sections, it makes the text look organized.

For line 8, which is on question 3.4, the line is a summary statement into the report which explains what files were moved and where.

For line 9, which is on question 3.4, after the “echo” there is a note, that note in the report reminds you that if any files couldn’t be moved which might be because of the permissions, those errors are already listed above in the report.

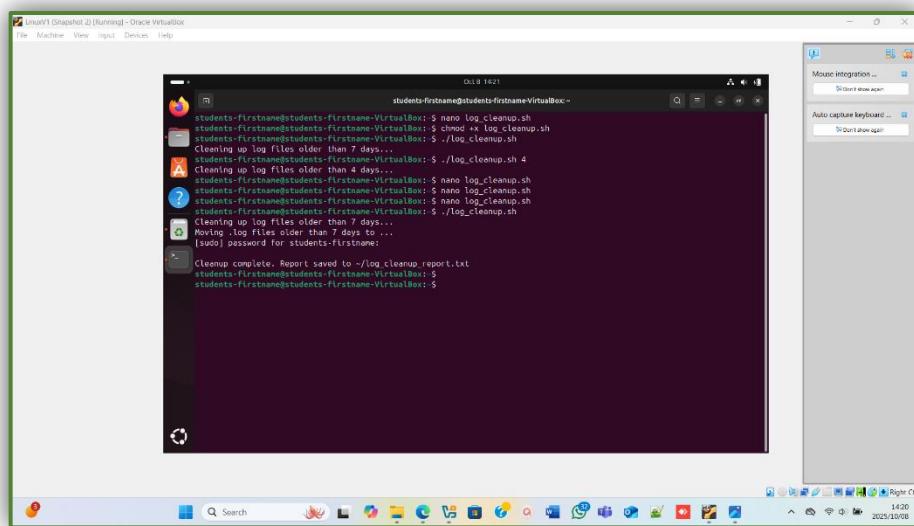
For line 10, which is on question 3.4, the line records the time the cleanup finished in the report, like the timestamp at the end.

For line 11, which is on question 3.4, the text that is after “echo” appears in the terminal (not the report). It basically tells the user that everything is done and where the report has been saved.

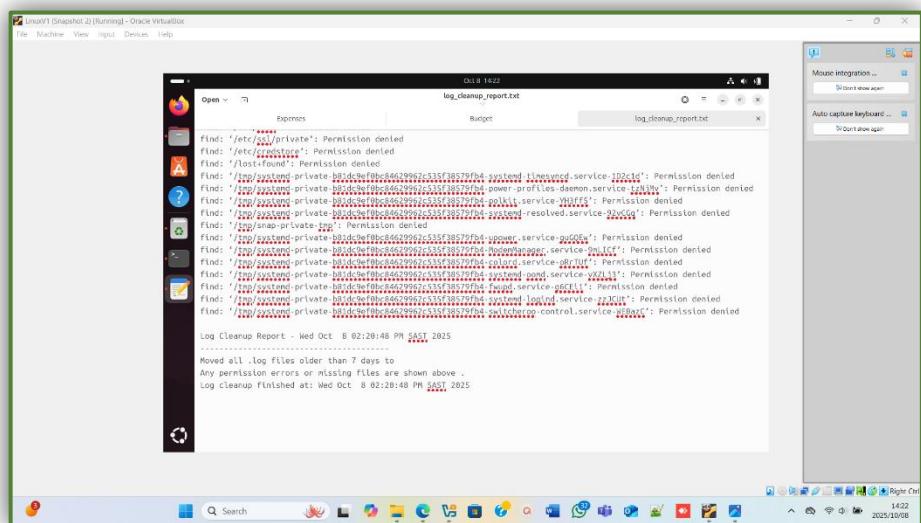
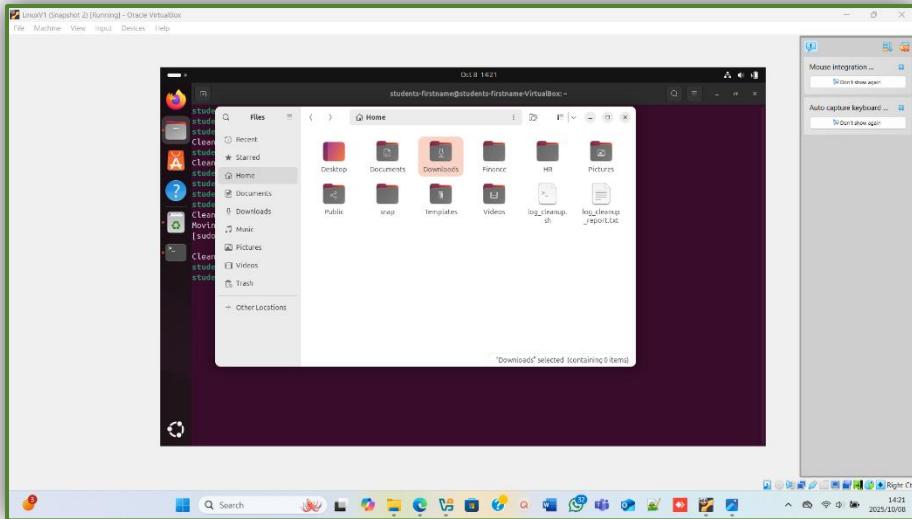
### 3.6. Students must take screenshots of:

- The script running in the terminal.
- Output messages (files moved, errors).

Like I have mentioned above the “**./log\_cleanup.sh**” command is used to run the script, specifically the “**./**” symbol which means run the file named **log\_cleanup.sh** in the current directory.



- The generated **log\_cleanup\_report.txt**.



## AI Declaration

I carefully read the assignment instructions, and the extent to which AI may be used for the assignment.

I used the following AI system(s)/tool(s):

I did not use any AI tool.

I used it for the following:

I did not make use of any AI tool.

If I quoted or paraphrased an AI output, I have referenced the relevant tool, version, and the date I used the tool.

I still consider this work my own (i.e., I have not outsourced the final product, or significant portions of it, to AI tools/systems)

If required, I can defend my argument/perspective, explain my choices and approach, and can show that I am knowledgeable about the details of my work.

## References

---

### Bibliography

---

- Programmer, T. (2024, August 13). *How to install Ubuntu 24.04 LTS in VirtualBox 2024*. Retrieved from Youtube: <https://youtu.be/Hva8lsV2nTk?si=X-Jy7Ue1im60npmN>
- ProgrammingKnowledge2. (2023, February 6). *How to Create And Run A Script In Ubuntu*. Retrieved from Youtube: [https://youtu.be/hKCX6nkA\\_oA?si=gvcXU0J2Janxrhi\\_](https://youtu.be/hKCX6nkA_oA?si=gvcXU0J2Janxrhi_)
- Tech, O. (2022, September 27). *How to create folder and file in linux using terminal | Ubuntu | 2022*. Retrieved from Youtube: [https://youtu.be/1Qyjqs-N4RQ?si=Vlu8\\_rY8hpH\\_qVKm](https://youtu.be/1Qyjqs-N4RQ?si=Vlu8_rY8hpH_qVKm)
- TV, L. L. (2024, October 29). *Linux Basics: How to Use the sort Command to Organize Data*. Retrieved from Youtube: <https://youtu.be/2B16aYGzIEQ?si=n9pr6MT1V4gRQeXV>