

Week 5 – Operating Systems

Student number: 578438

Assignment 5.1: Unix-like

- a) Find out what the difference is between UNIX and unix-like operating systems?

UNIX originally refers to the operating system created at AT&T Bell Labs in the 1970s, and today the name “UNIX” is a protected trademark that can only be used by systems that pass formal certification (the Single UNIX Specification/POSIX). These certified systems, such as AIX, HP-UX or some versions of macOS, are considered “true” UNIX and directly inherit the original design and standards.

Systems that are "Unix-like," on the other hand, are operating systems that act a lot like UNIX and follow many of the same rules and interfaces. However, they aren't built from the original code or officially certified. Think of this group as including popular Linux versions like Ubuntu, plus FreeBSD, OpenBSD, and NetBSD. Basically, from a user's or developer's perspective, Unix-like systems feel super similar to UNIX, even though they're legally and historically separate things.

- b) Study the image above named UNIX timeline. Find out who Ken Thompson, Dennis Ritchie, Bill Joy, Richard Stallman, and Linus Torvalds are and what they have contributed to the development of UNIX or unix-like systems and to IT in general. **TIP!** English-language sources often contain more detailed information about these individuals.

- Ken Thompson
 - He was Computer scientist at Bell Labs.
 - He was Co-creator of UNIX and the B programming language, significant contributions to Plan 9 and other operating systems.
 - He helped define the fundamental design of modern operating systems.
- Dennis Ritchie
 - Also affiliated with Bell Labs.
 - Co-creator of UNIX alongside Ken Thompson.
 - Developed the C programming language, which became the standard language for OS and systems programming.
- Bill Joy
 - He was Co-founder of Sun Microsystems.
 - Lead developer of BSD Unix (4BSD) at the University of California, Berkeley.
 - Wrote important tools such as the vi editor and the C shell (csh).
 - His BSD work strongly influenced later Unix and Unix-like systems.
- Richard Stallman
 - He was Founder of the GNU Project (1983) and the Free Software Foundation (FSF).
 - Initiated many key free-software components: GNU Emacs, GCC compiler, and the concept of copyleft via the GNU General Public License (GPL).

- His work provided most of the user-space tools used with the Linux kernel.
- Linus Torvalds
 - Created the Linux kernel in 1991 as a free, Unix-like kernel.
 - Still leads Linux kernel development.
 - Also created Git, the distributed version-control system.

c) What is the philosophy of the GNU movement?

The core of the GNU movement, and the Free Software Foundation (FSF), is the belief in software freedom. This is defined by four fundamental rights for every user:

1. **Run the program:** The freedom to use the software for any purpose.
2. **Study and modify:** The freedom to examine how the program works and make changes to it (which requires access to the source code).
3. **Redistribute exact copies:** The freedom to share precise copies of the software.
4. **Distribute modified versions:** The freedom to share custom versions so the entire community can benefit from improvements.

The movement views these freedoms as an ethical and social issue, not just a technical one. To protect them, GNU promotes copyleft licenses (like GPL) that require derived works to remain free.

d) Does Ubuntu as a Linux operating system conform to the philosophy of the GNU movement?
Please explain your answer.

Ubuntu is technically built from predominantly GNU software and the Linux kernel, licensed under free-software licenses like the GPL, granting users substantial freedoms. However, it is not fully conformant with the strict standards of the GNU philosophy and the Free Software Foundation (FSF) because it includes and promotes non-free components such as proprietary drivers, firmware, and optional non-free codecs or applications via third-party repositories and Snaps. Fully conformant distributions (like Trisquel or Parabola) strictly exclude all proprietary software.

e) Find out what is the Windows Subsystem for Linux?

The Windows Subsystem for Linux (WSL) constitutes a compatibility layer within the Windows operating system, enabling users to execute a Linux environment directly. This functionality facilitates the installation of authentic Linux distributions sourced from the Microsoft Store and permits the simultaneous utilization of Linux command-line utilities alongside Windows applications.

WSL exists in two iterations:

1. WSL 1, which functions by translating Linux system calls into Windows kernel calls.
2. WSL 2, which leverages a lightweight virtual machine incorporating a complete Linux kernel to achieve enhanced compatibility and performance.

f) Find out, which operating system family belongs to Android, iOS and ChromeOS?

Android

- **Basis:** Utilizes the Linux kernel in conjunction with proprietary Google libraries and frameworks.
- **Family:** Linux / Unix-like.

iOS

- **Basis:** Constructed upon Darwin, which is derived from BSD Unix and NeXTSTEP (both of which are Unix-like systems).
- **Family:** BSD / Unix-like (Shares the same foundational family as macOS).

ChromeOS

- **Basis:** Relies on the Linux kernel and a user space derived from Gentoo Linux.
- **Family:** Linux / Unix-like.

Assignment 5.2: Supercomputers and gameconsoles

- a) Research on this site what supercomputers are used for and write a short summary of it:
<https://www.computerhistory.org/timeline/search/?q=Supercomputer>

Supercomputers are some of the most powerful computers in existence, handling problems that require incredible processing power. They tackle challenges that would take regular computers years to solve. Here are some ways supercomputers make a difference:

In Engineering and industry, they are used for designing aircraft, cars and materials, testing them virtually instead of with many physical prototypes. They are used for Nuclear and weapons research simulating nuclear explosions and ensuring the safety of nuclear stockpiles without real tests. In Scientific research simulating climate and weather, modelling the universe, black holes, galaxies, and other physical processes. In Medicine and biology, they are used for protein-folding, gene analysis, drug discovery, and simulating how diseases spread. And in Cryptography and security they are used for breaking or testing encryption and securing communications.

They are built to perform a massive number of operations in parallel and are usually formed from thousands of connected processors.

- b) IBM is a company that has already built a number of supercomputers. One of them is IBM's Roadrunner. The CPU developed for this supercomputer was further developed at a later stage as the CPU for the PlayStation 3 console. Find out what a **PlayStation 3 cluster** is and what it was used for?

A PlayStation 3 cluster is basically a group of PlayStation 3 consoles that are connected and used like a small supercomputer.

The key idea is that the PS3 uses the Cell Broadband Engine processor, which is related to the CPU used in IBM's Roadrunner supercomputer. For a while, Sony allowed users to install Linux on the PS3 ("OtherOS" feature). By installing Linux on many PS3s and connecting them over a network, researchers could build a cheap high-performance computing cluster.

What it was used for:

Scientific research – Universities built PS3 clusters to run physics and math simulations. For example, some groups used them for astrophysics, like simulating black holes and gravitational waves, because the Cell processor was very good at number-crunching.,

Military and government work – The U.S. Air Force Research Laboratory built a large PS3 cluster often called the Condor Cluster with over a thousand PS3s. It was used for things like: Processing and analysing high-resolution images and video,

Pattern recognition and AI-related tasks,
Other workloads that need a lot of parallel computation,

General high-performance computing experiments Labs like Los Alamos used PS3 clusters as a low-cost testbed for Cell-based supercomputing, related to systems like Roadrunner.,

In short, a PlayStation 3 cluster is a bunch of PS3s turned into a parallel computer and used for serious scientific and technical computing, because they offered a lot of computing power for relatively little money.

- c) You can build a supercomputer by putting a few computers together in a cluster. Here's what Oracle did with a collection of Raspberry Pi's, for example:

<https://blogs.oracle.com/developers/post/building-the-worlds-largest-raspberry-pi-cluster>

What specific operating system is running on this cluster?

Oracle's Raspberry Pi supercomputer runs Oracle Linux, the ARM version of Oracle Linux 7.

- d) Does Oracle's Raspberry Pi supercomputer appear in the list of the 500 fastest supercomputers in the world? Make a logical decision for this, without going through the entire list.

<https://www.top500.org/lists/top500/list/2023/06/>

No, Oracle's Raspberry Pi "supercomputer" will not appear in the TOP500 list.

The TOP500 only includes the 500 most powerful supercomputers in the world, measured in floating-point performance Rmax on the Linpack benchmark, which is currently in the petaflops range (10^{15} operations per second) and above.,

A Raspberry Pi is a low-power single-board computer designed for education, hobby projects and light workloads, not high-performance computing.,

Even if you connect hundreds or a few thousand Raspberry Pis, their combined performance is still only in the gigaflops to low teraflops range (10^9 – 10^{12} operations per second), which is orders of magnitude below modern TOP500 systems.,

Oracle's Raspberry Pi cluster is clearly presented as a demonstration/experimental cluster, not as a machine aimed at TOP500-class performance.,

So, logically, it is far too weak to qualify and therefore will not be on the list of the 500 fastest supercomputers.

- e) What CPU architecture is used for the PlayStation 5 and Xbox Series X?

What operating systems run on these consoles?

What conclusion can you draw from the answer to the previous question?

1. CPU architecture used in PlayStation 5 and Xbox Series X

PlayStation 5: Custom AMD Zen 2, 8-core, 64-bit x86-64 (AMD64) CPU.

Xbox Series X: Custom AMD Zen 2, 8-core, 64-bit x86-64 (AMD64) CPU.,

So, both consoles use very similar PC-style 64-bit x86 processors from AMD.

Operating systems of Ps5 and Xbox:

PlayStation 5: Runs a proprietary Sony operating system that is based on FreeBSD, a Unix-like OS like what the PS4 used, often called Orbis OS.

Xbox Series X: Runs a customized version of Microsoft Windows, built on the Windows NT kernel essentially a specialized Windows 10/11-family OS for Xbox.

Conclusion from this,

Both the PS5 and Xbox Series X:

Use standard PC-style x86-64 CPUs, and,

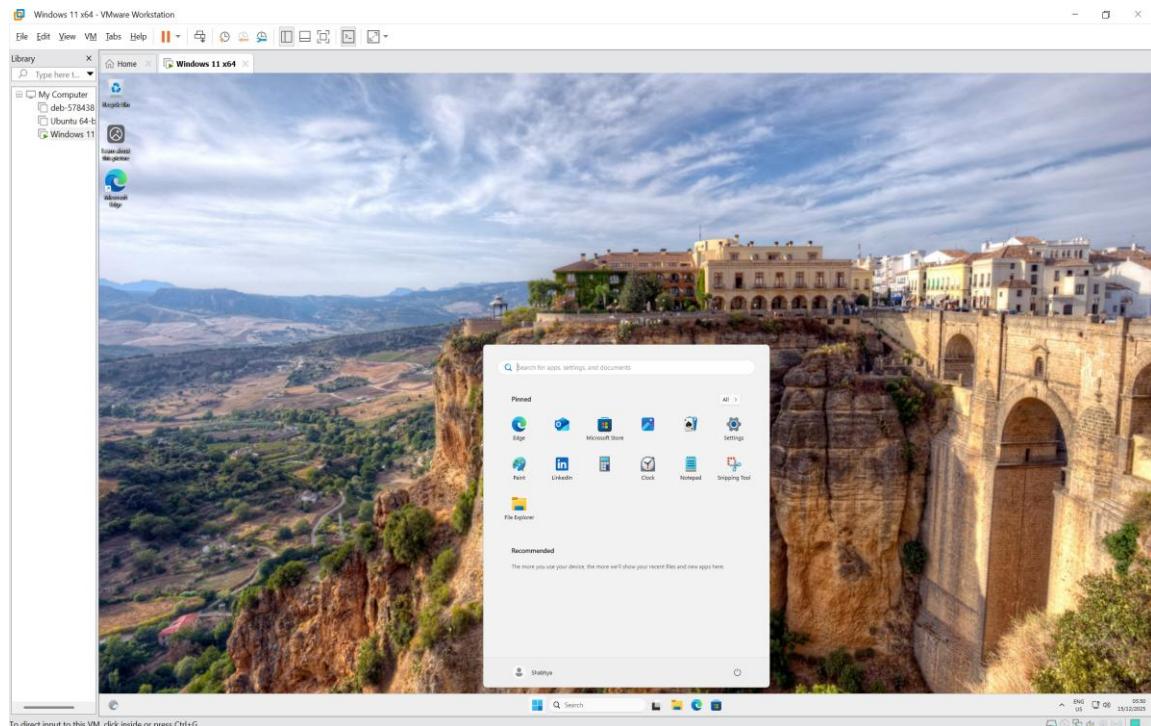
Run operating systems that are closely related to mainstream desktop OS families Unix-like/BSD for PS5, Windows NT for Xbox.

The main conclusion is that modern game consoles are, under the hood, very similar to locked-down, specialized PCs, which makes it much easier to develop and port games between console and PC.

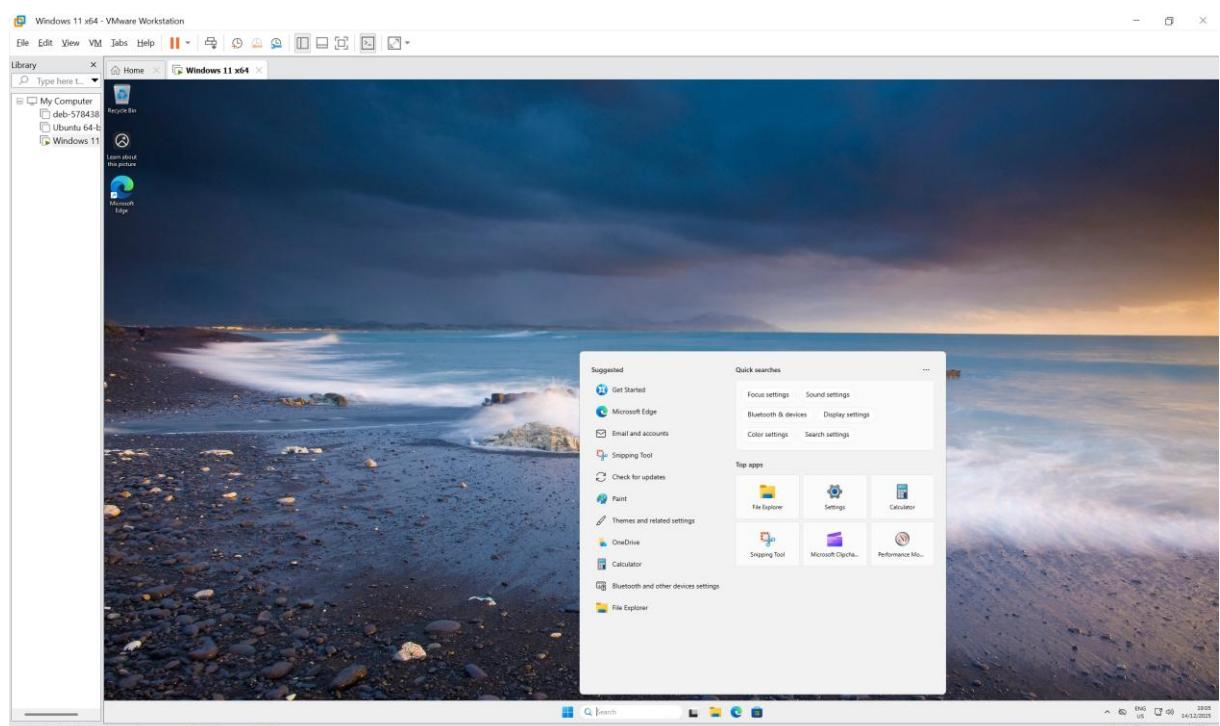
Assignment 5.3: Working with Windows

Take relevant screenshots of the assignments below

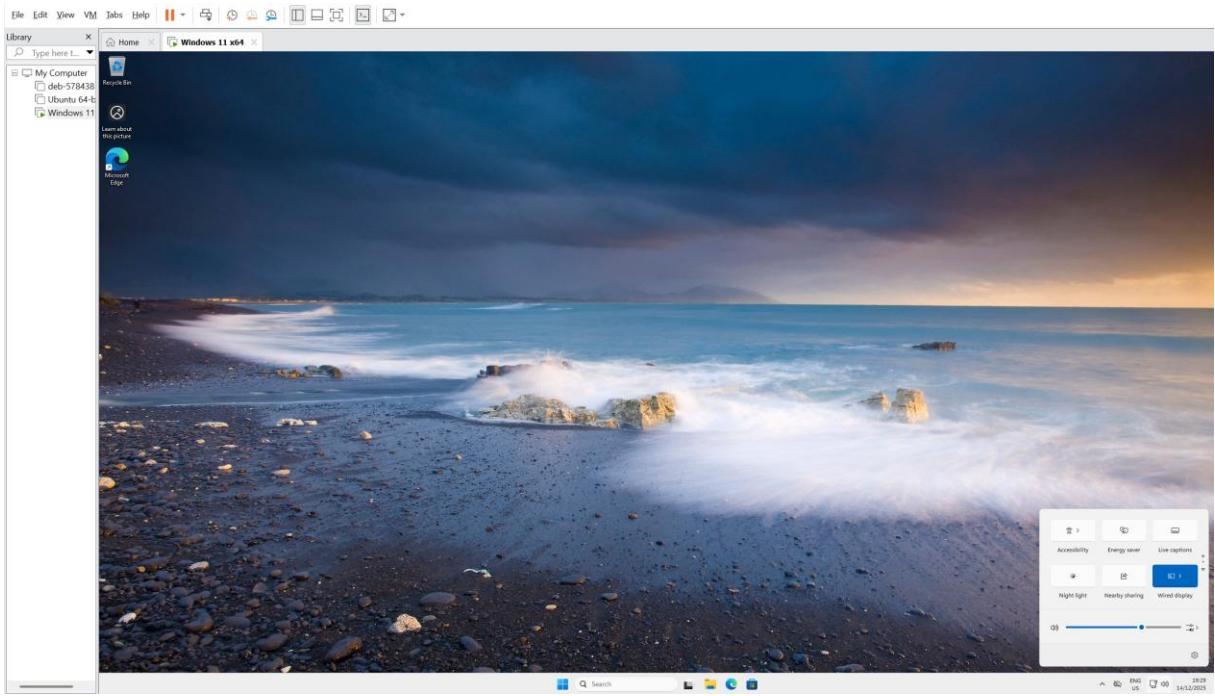
- a) Practice for about 10 minutes with the  keyboard shortcuts combinations, skip the general shortcuts in this exercise. Take a look at which screens are opened.



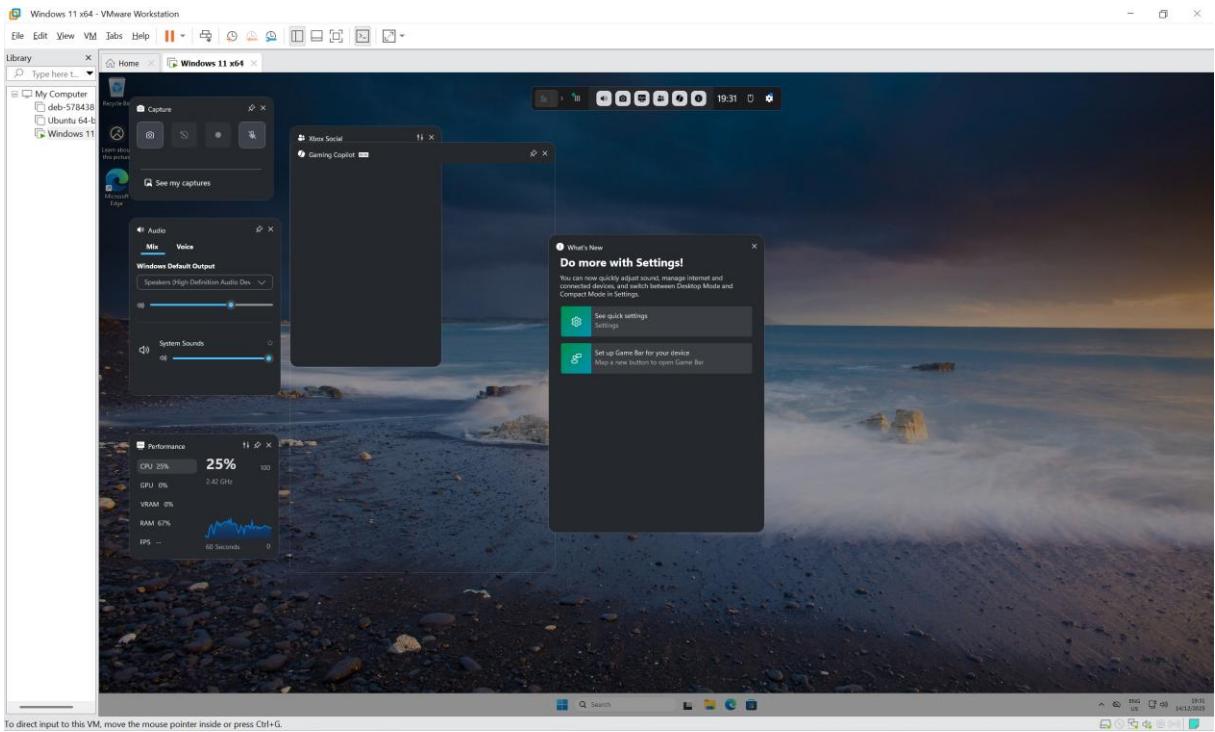
Windows key – Start Menu



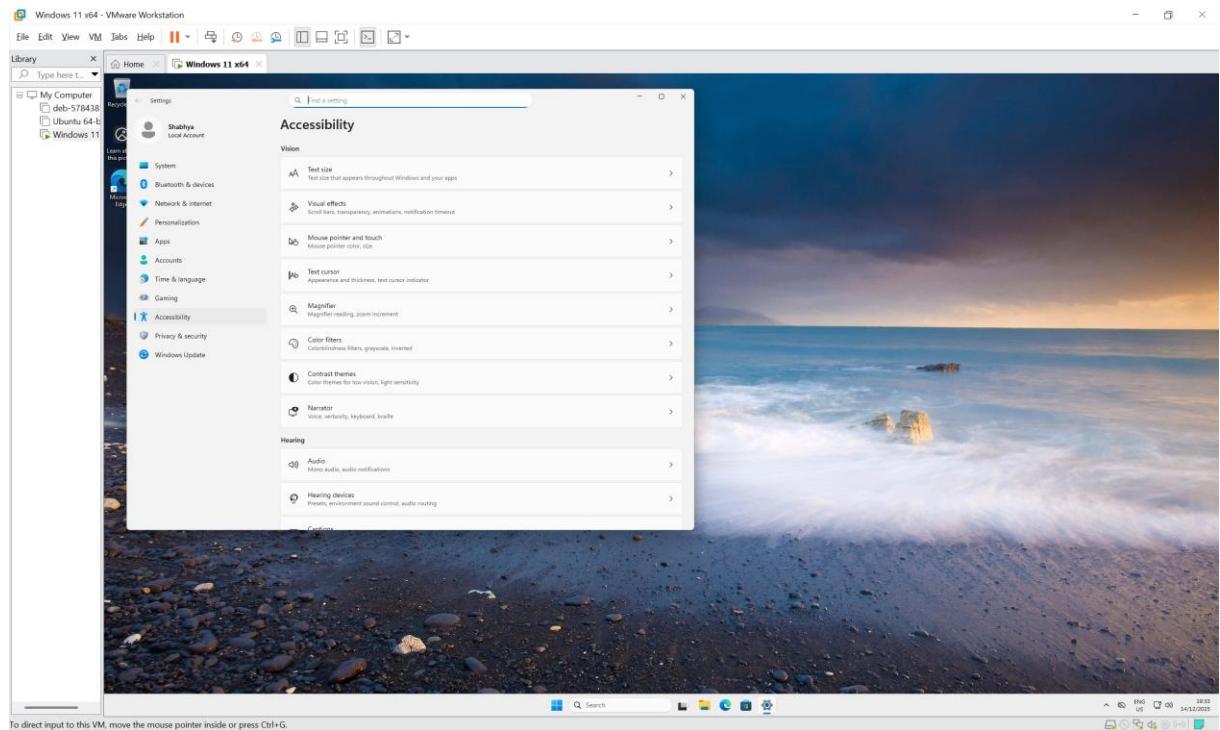
Windows + S “Search”



Windows + A “Action Screen”

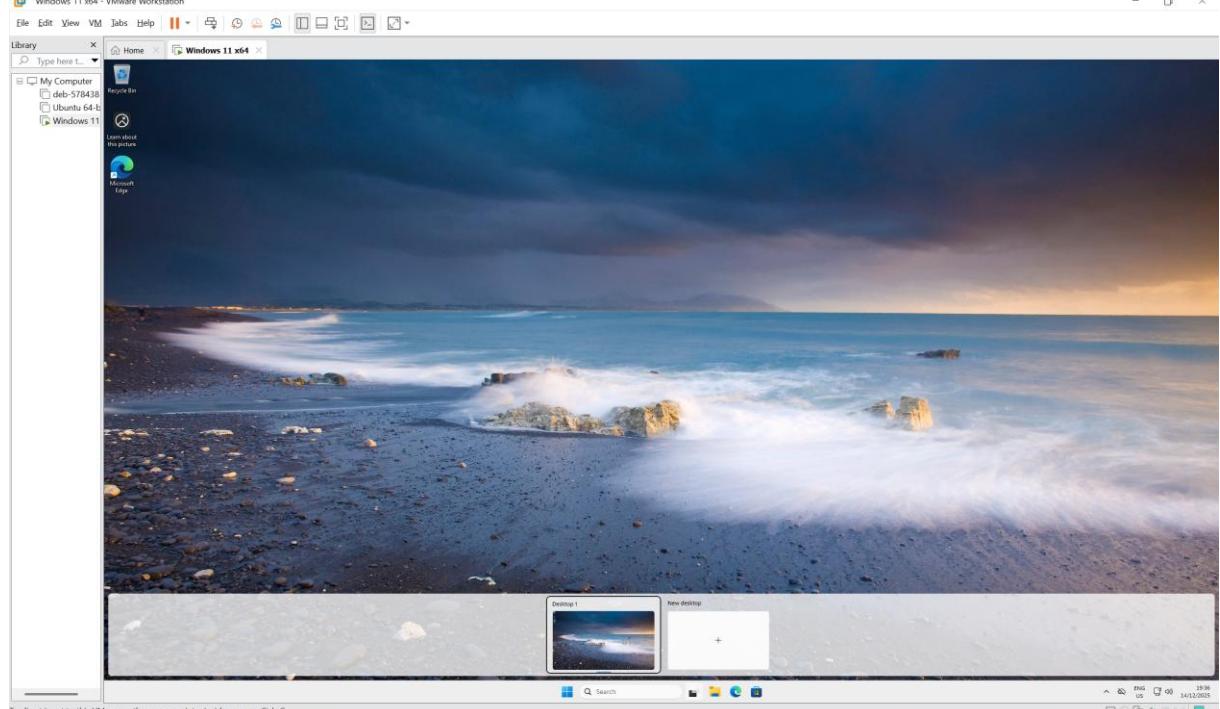


Windows G “Game Bar”



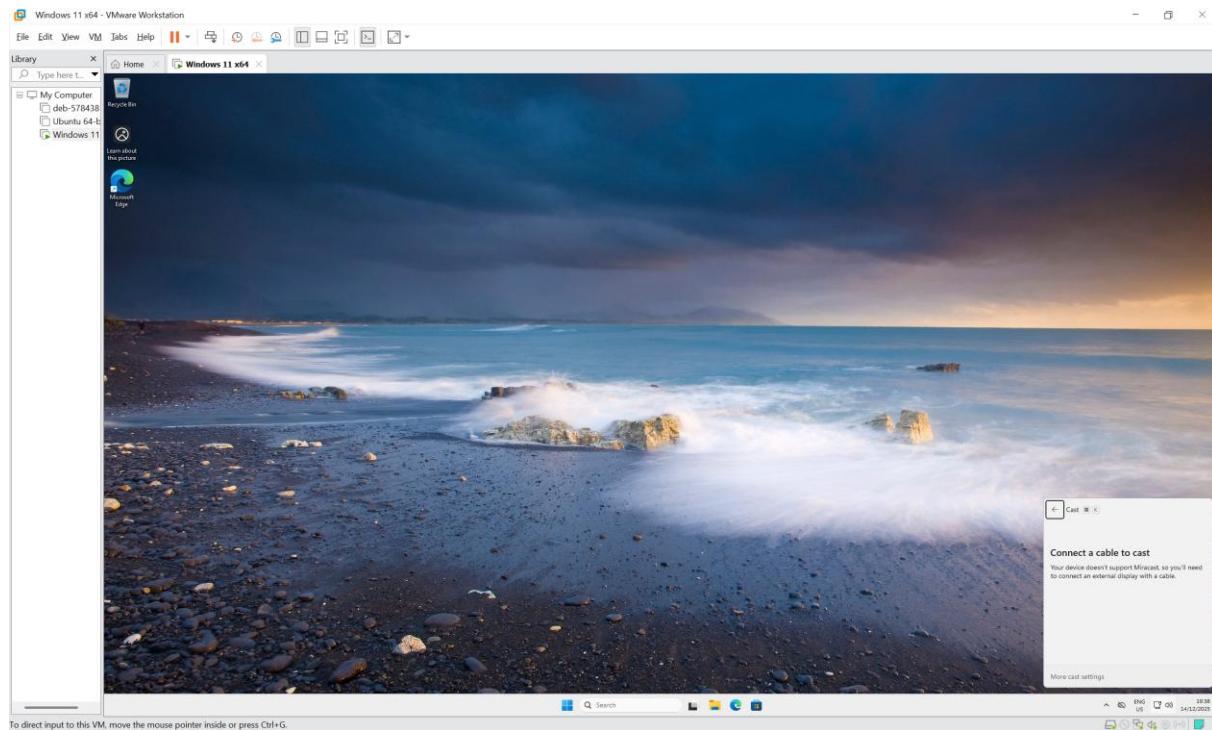
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Windows + U “Connect”



To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Windows + Tab “Task view”

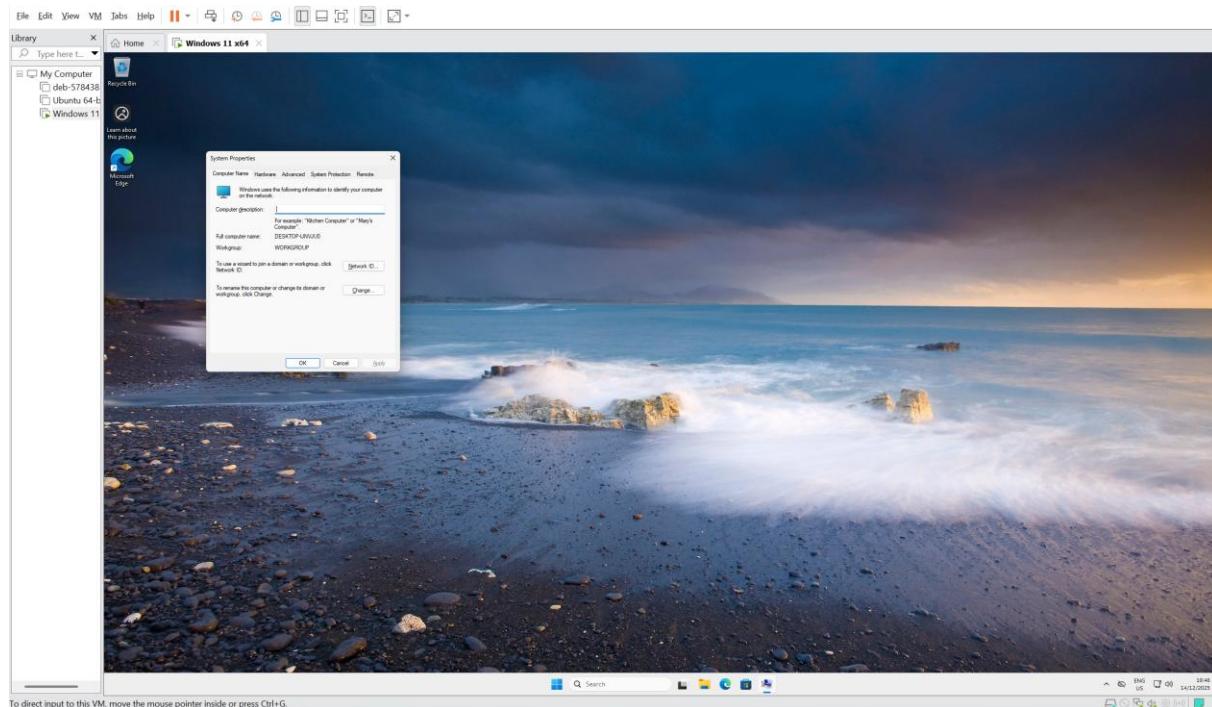


Windows + K “Connect”

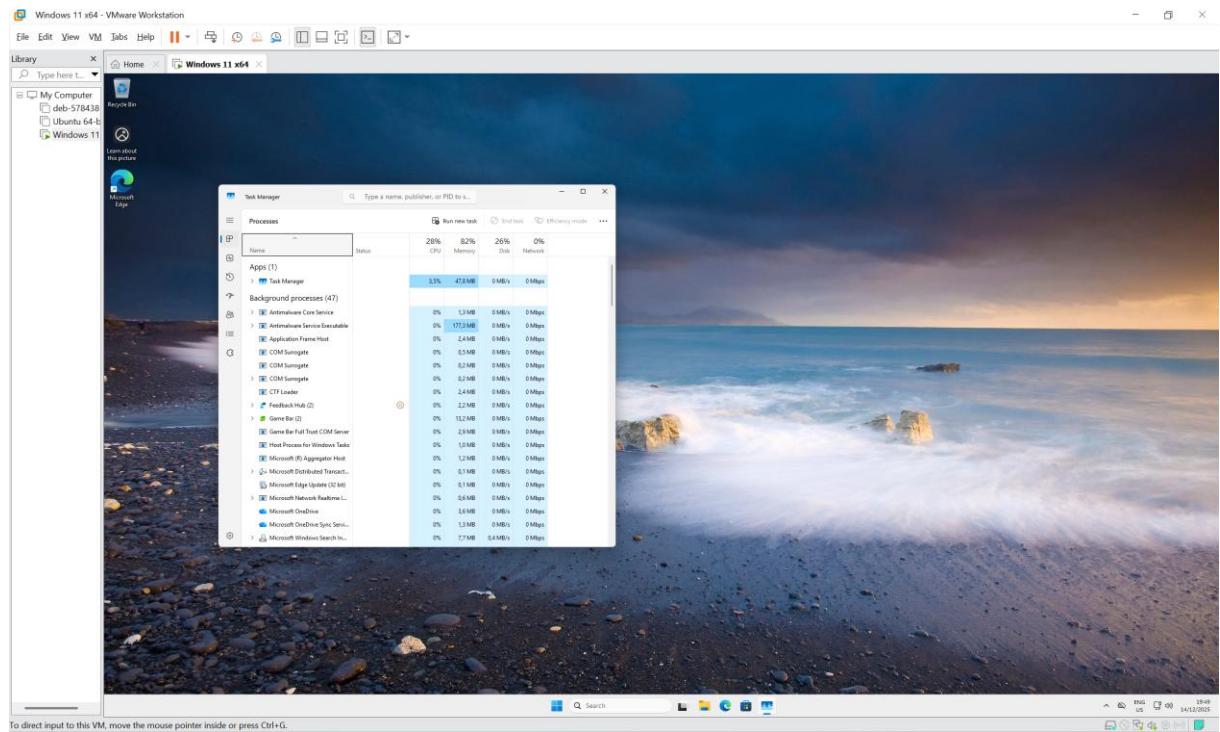
- b) The file explorer can be opened with **Windows + E**, which key combination could you also use?

We can also use key combination, Windows + X then E to open file explorer.

- c) Open the system properties with a **Windows** key combination, take a screenshot of the open screen. Paste this screenshot into this template.

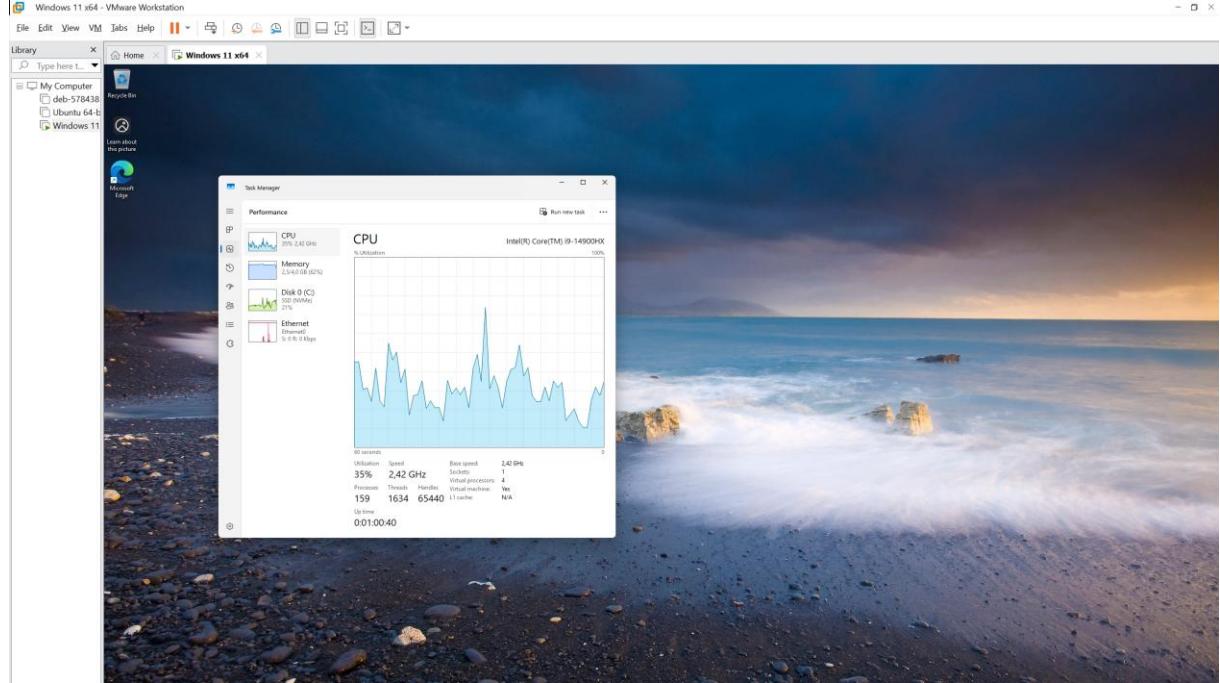


- d) Open task manager with a key combination. Take screenshots of the tabs: processes (shows active processes), performance, and users. Place these three screenshots in this template.

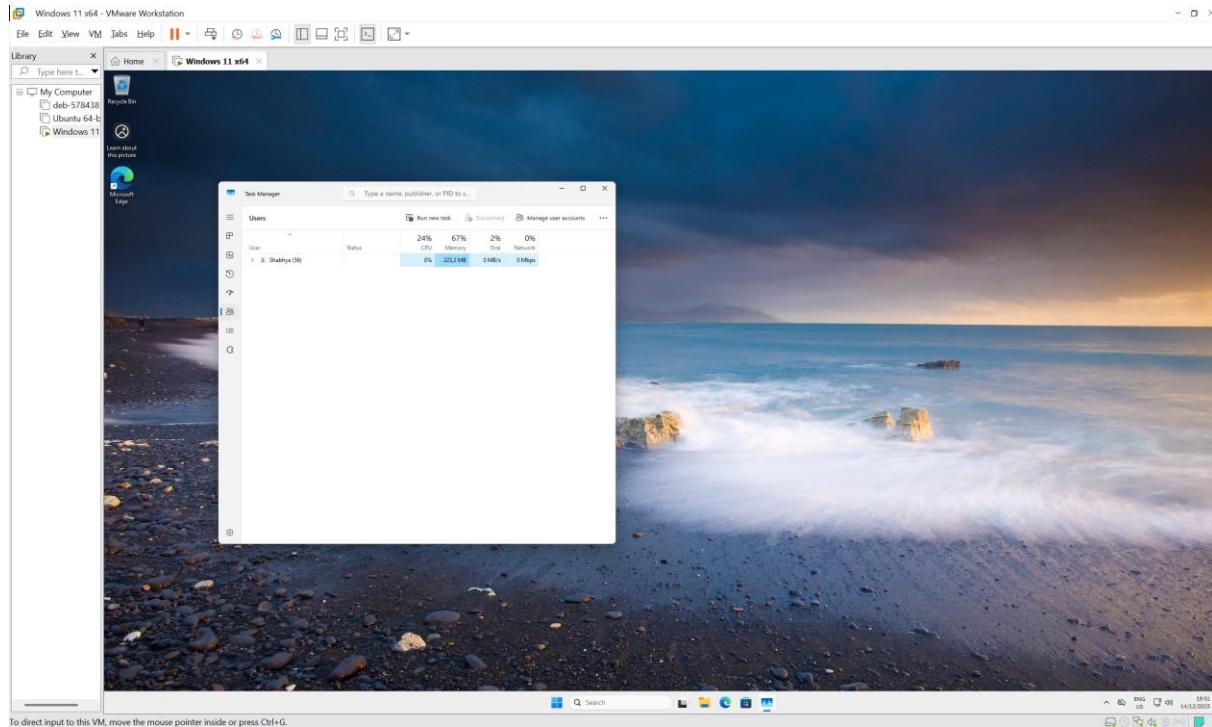


To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Processes

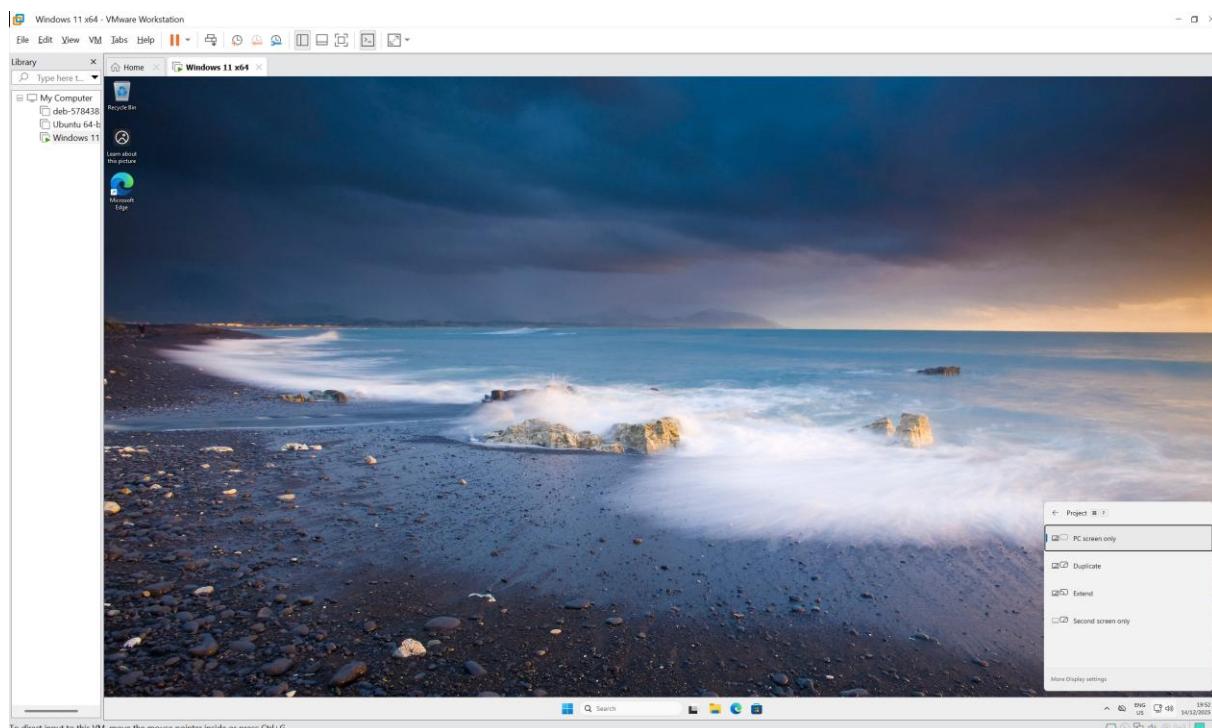


Performance



User

- e) If you're giving a PowerPoint presentation and you connect your laptop to a projector, Windows can use the projector as a second screen. For example, you may have Outlook open on your first screen that you don't show over the projector, while the PowerPoint presentation is displayed on the projector, or the second screen. Which key combination should you use for this?

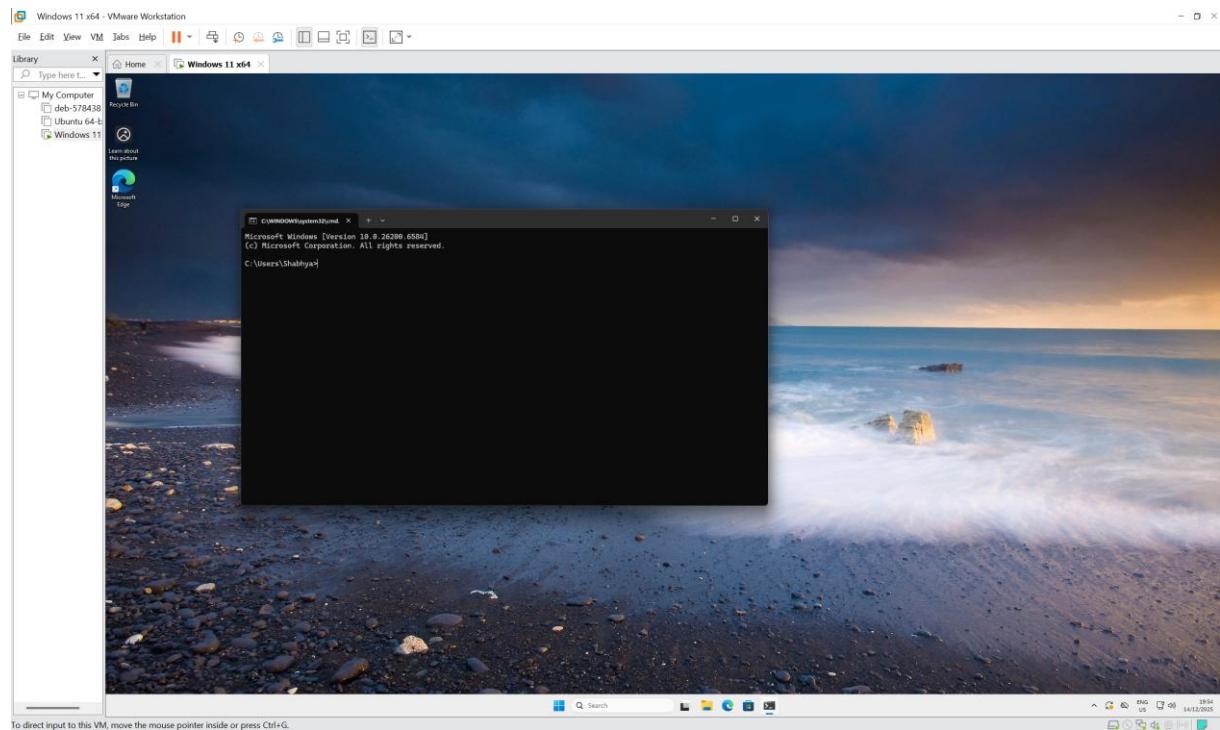


Windows + P

- f) If you leave the classroom for a while and you leave your laptop behind, it is wise to lock the screen. Your Apps will continue to run in the background. So, for example, if you're waiting for a download that takes a while, lock the screen and get a cup of coffee. Which key combination do you use for this?

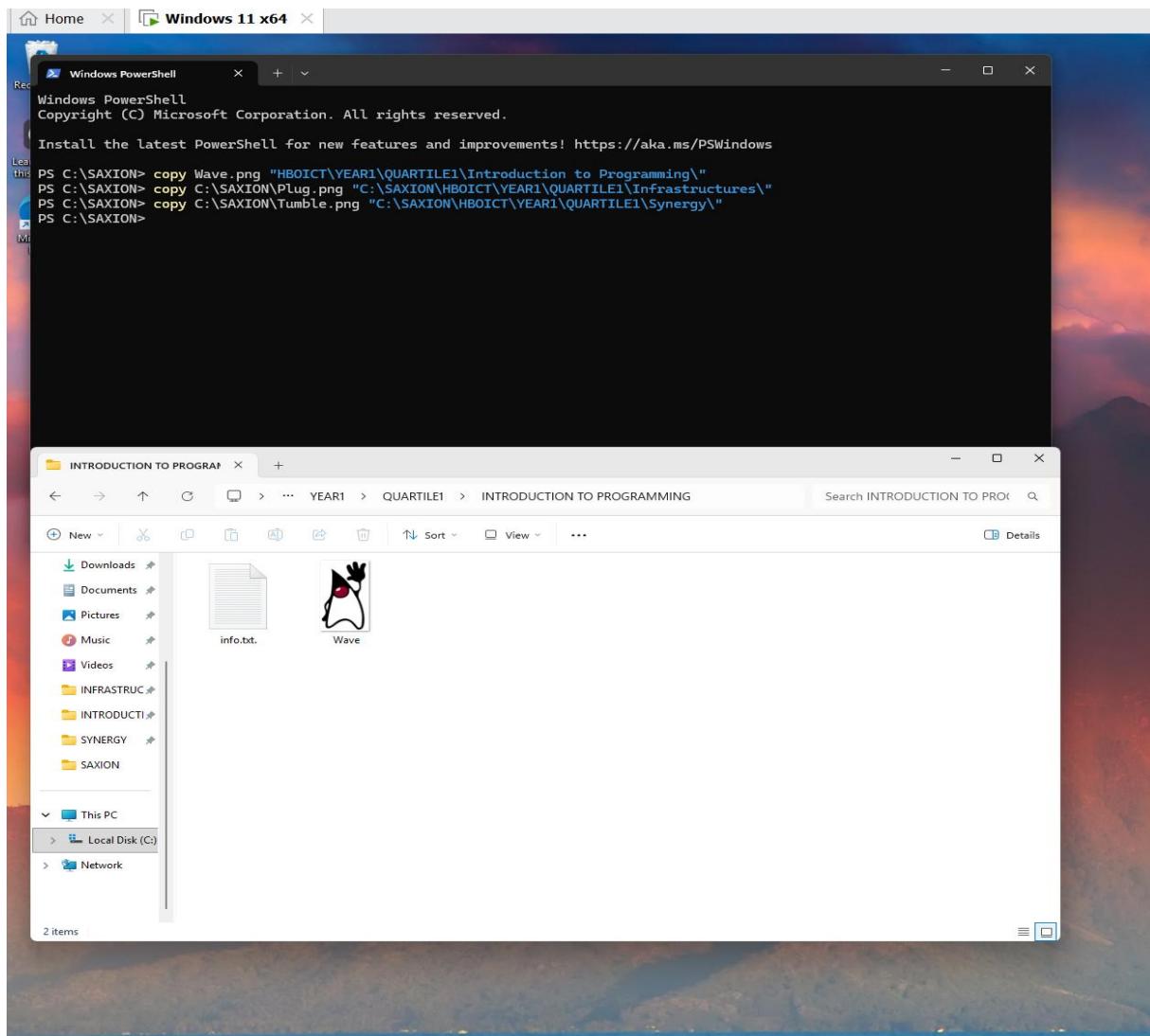
We use Windows + L

- g) Open the Run screen with a key combination. On this screen, type CMD and press <enter>. Take a screenshot of this result and paste it into this template.

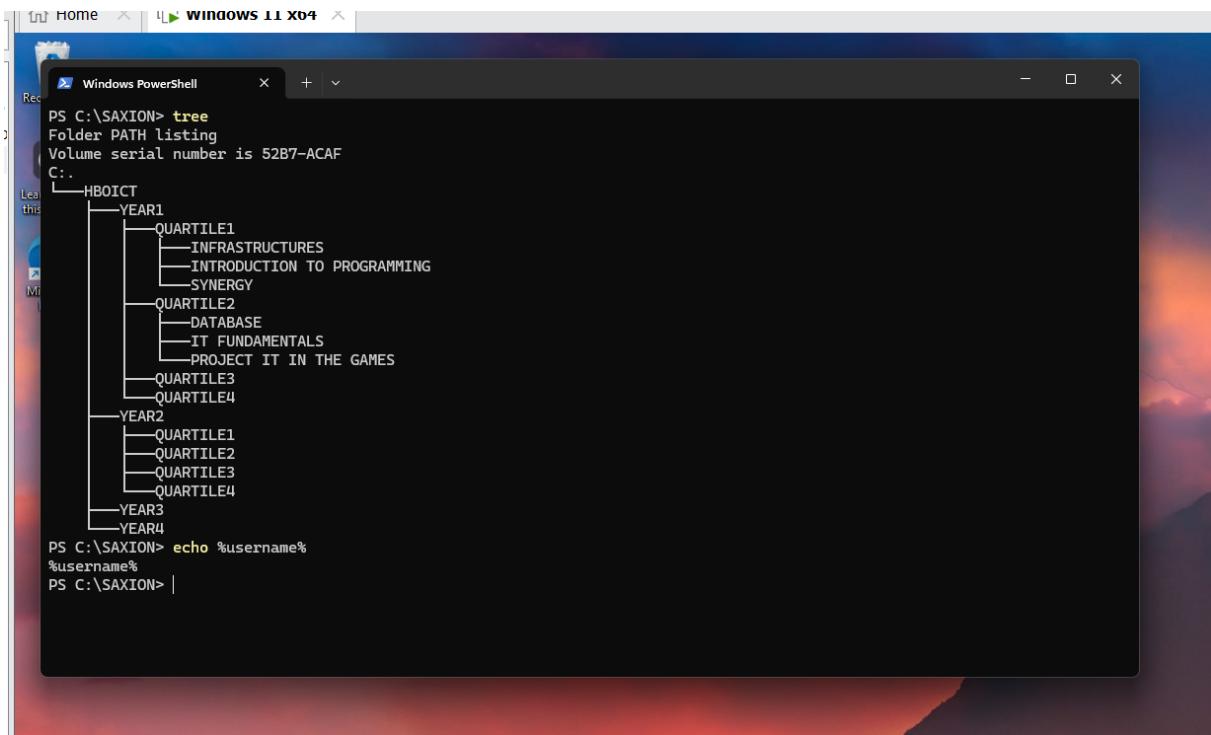


Working in the File Explorer

Relevant screenshots **copy** command:



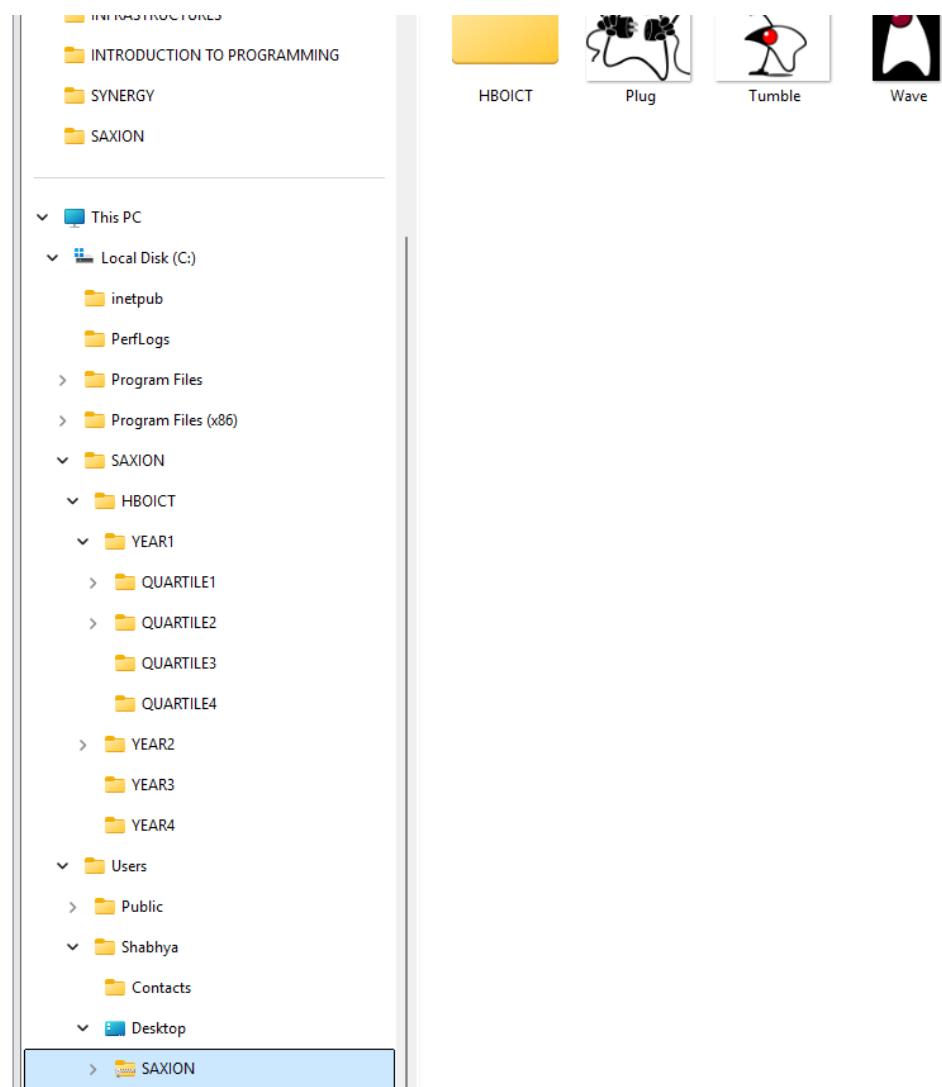
Relevant screenshots **tree** command:



```
PS C:\SAXION> tree
Folder PATH listing
Volume serial number is 52B7-ACAF
C:.
└── HBOICT
    ├── YEAR1
    │   ├── QUARTILE1
    │   │   ├── INFRASTRUCTURES
    │   │   ├── INTRODUCTION TO PROGRAMMING
    │   │   └── SYNERGY
    │   ├── QUARTILE2
    │   │   ├── DATABASE
    │   │   ├── IT FUNDAMENTALS
    │   │   └── PROJECT IT IN THE GAMES
    │   ├── QUARTILE3
    │   ├── QUARTILE4
    ├── YEAR2
    │   ├── QUARTILE1
    │   ├── QUARTILE2
    │   ├── QUARTILE3
    │   ├── QUARTILE4
    ├── YEAR3
    └── YEAR4

PS C:\SAXION> echo %username%
%username%
PS C:\SAXION> |
```

Relevant screenshots in the file explorer of the folder c:\Saxion + created zip file.



Terminating Processes

Relevant Screenshots Task Manager Window:

Processes		Run new task	End task	Efficiency mode	...
Name	Status	3% CPU	64% Memory	0% Disk	0% Network
Apps (2)					
> Calculator (2)	(II)	0%	2,3 MB	0 MB/s	0 Mbps
> Task Manager					
		3,4%	44,5 MB	0 MB/s	0 Mbps
Background processes (39)					
> Antimalware Core Service		0%	2,5 MB	0 MB/s	0 Mbps
> Antimalware Service Executable		0%	131,9 MB	0 MB/s	0 Mbps
Application Frame Host		0%	5,4 MB	0 MB/s	0 Mbps
COM Surrogate		0%	0,1 MB	0 MB/s	0 Mbps
> COM Surrogate		0%	1,0 MB	0 MB/s	0 Mbps
CTF Loader		0%	2,6 MB	0 MB/s	0 Mbps
Host Process for Windows Tasks		0%	1,7 MB	0 MB/s	0 Mbps
Microsoft (R) Aggregator Host		0%	1,7 MB	0 MB/s	0 Mbps
> Microsoft Content	(II)	0%	0 MB	0 MB/s	0 Mbps
> Microsoft Distributed Transact...		0%	0,1 MB	0 MB/s	0 Mbps
> Microsoft Edge (6)		0%	16,6 MB	0 MB/s	0 Mbps
Microsoft Edge Update (32 bit)		0%	0,1 MB	0 MB/s	0 Mbps
> Microsoft Network Realtime I...		0%	0,7 MB	0 MB/s	0 Mbps
Microsoft OneDrive		0%	4,7 MB	0 MB/s	0 Mbps
Microsoft OneDrive Sync Servi...		0%	1,0 MB	0 MB/s	0 Mbps
Microsoft OneDriveFile Co-Au...		0%	0,1 MB	0 MB/s	0 Mbps
> Microsoft Windows Search In...		0%	11,2 MB	0 MB/s	0 Mbps

Install Software

Relevant screenshots that the following software is installed with winget:

- WinSCP

```
C:\Windows\System32>winget install -e --id WinSCP.WinSCP
Found WinSCP [WinSCP.WinSCP] Version 6.5.5
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://sourceforge.net/projects/winscp/files/WinSCP/6.5.5/WinSCP-6.5.5-Setup.exe/download
[██████████] 11.6 MB / 11.6 MB
Successfully verified installer hash
Starting package install...
Successfully installed

C:\Windows\System32>
```

- Notepad++

```
C:\Windows\System32>winget install -e --id Notepad++.Notepad++
Found Notepad++ [Notepad++.Notepad++] Version 8.9
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://github.com/notepad-plus-plus/notepad-plus-plus/releases/download/v8.9/npp.8.9.Installer.x64.exe
[██████████] 6.54 MB / 6.54 MB
Successfully verified installer hash
Starting package install...
Successfully installed

C:\Windows\System32>
```

- 7zip

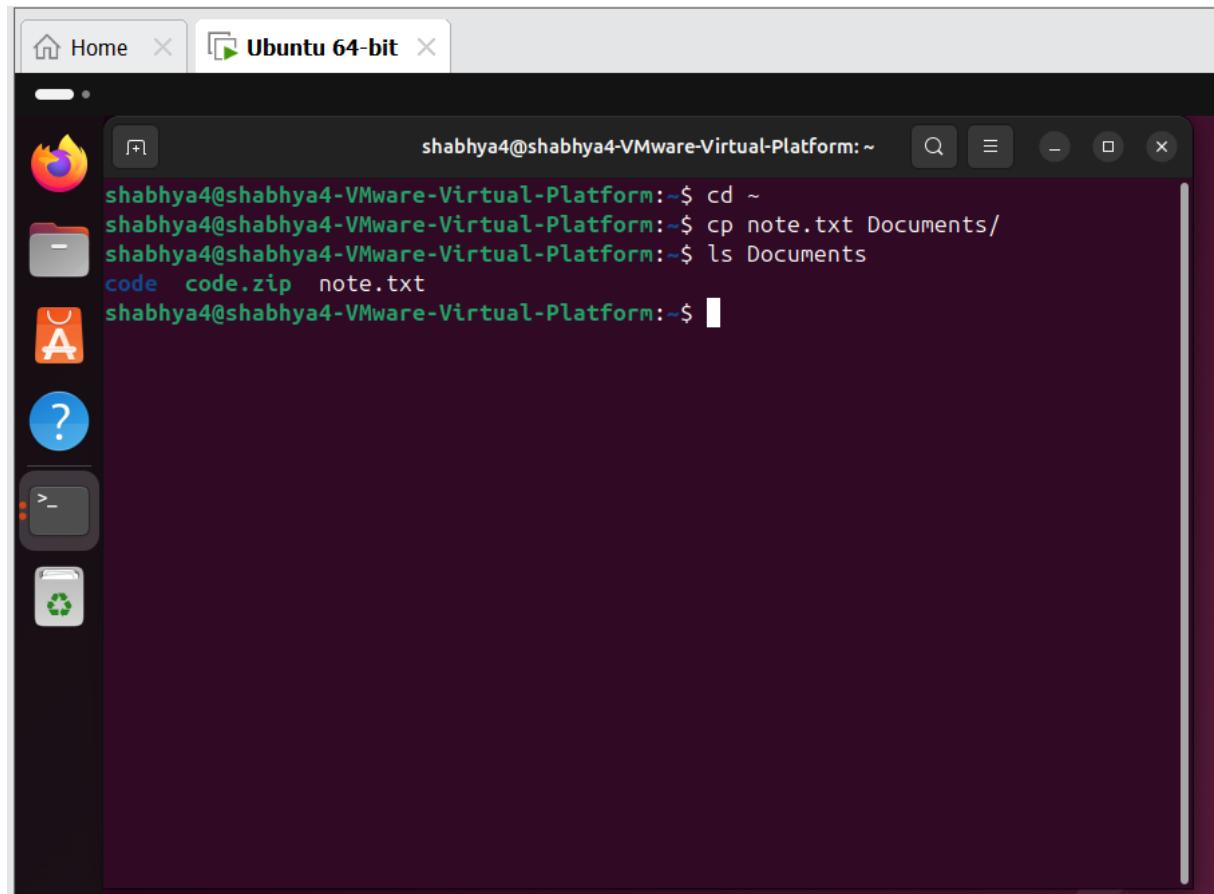
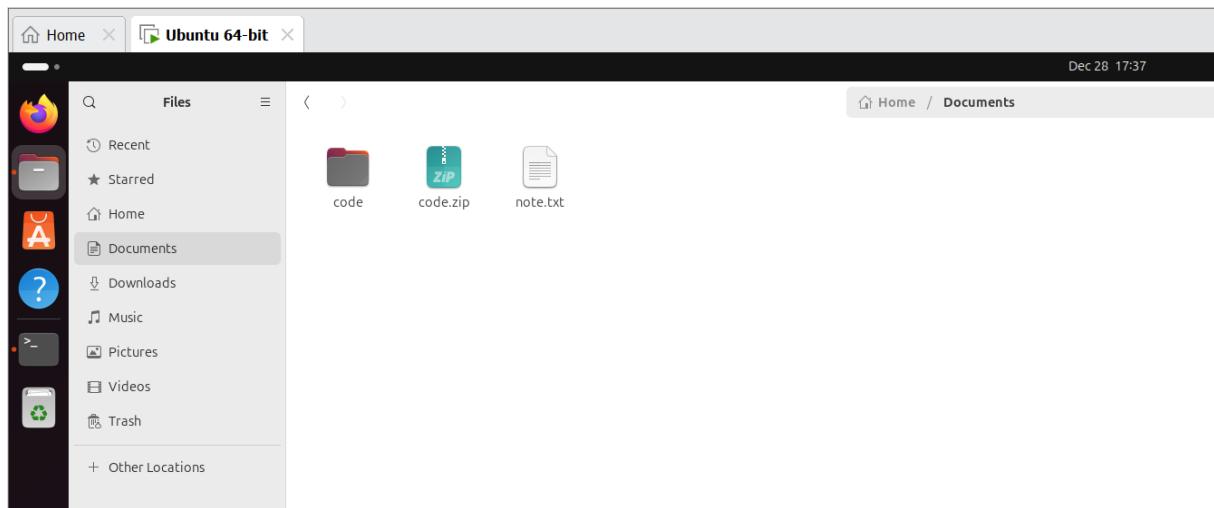
```
C:\Windows\System32>winget install -e --id 7zip.7zip
Found 7-Zip [7zip.7zip] Version 25.01
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://7-zip.org/a/7z2501-x64.exe
[██████████] 1.56 MB / 1.56 MB
Successfully verified installer hash
Starting package install...
Successfully installed

C:\Windows\System32>
```

Assignment 5.4: Working with Linux

Relevant screenshots + motivation

- Copying files



I Created a file named "note.txt" within my home directory. Then I Copied the file to the "Documents" directory once using the file explorer first screenshot , and then using the terminal command "cp note.txt Documents/" in Second screenshot.

- Navigating the file structure

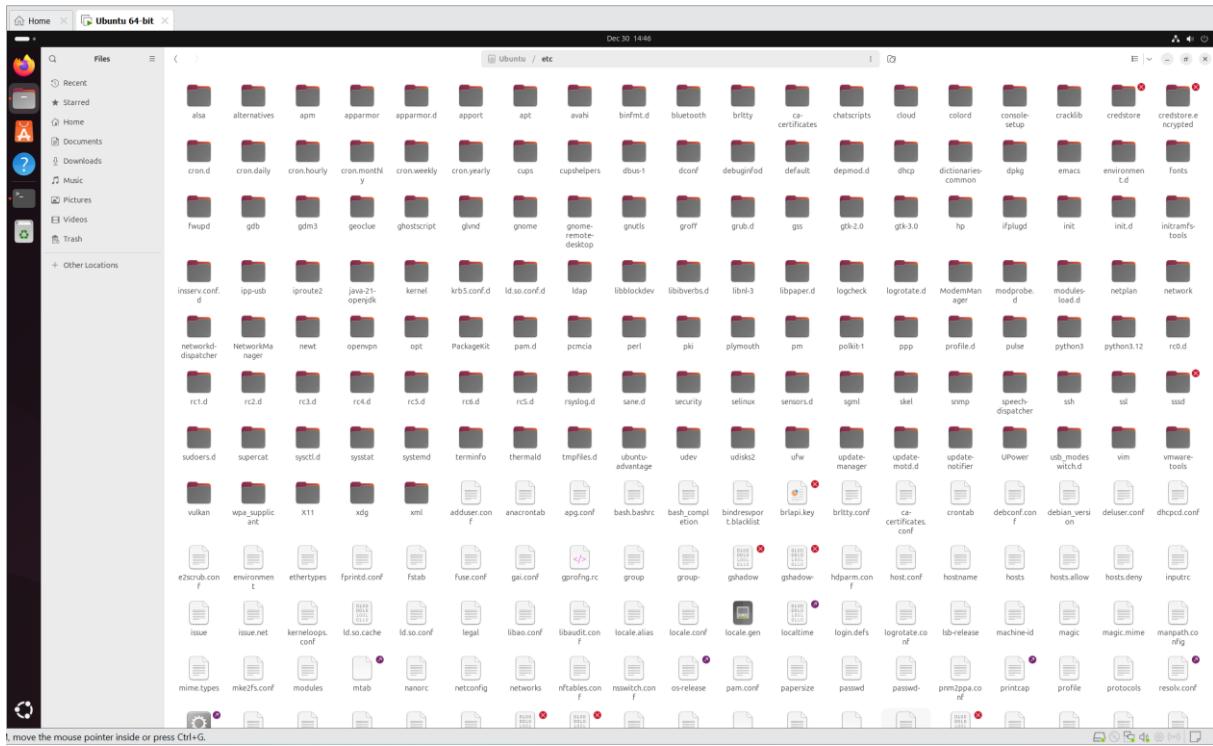


Fig: Navigate to the /etc folder in the file explorer.

The screenshot shows a desktop environment with a terminal window open. The terminal window is titled "Ubuntu 64-bit". The terminal content is as follows:

```
shabhy4@shabhy4-Virtual-Platform:~$ cd /etc
shabhy4@shabhy4-Virtual-Platform:~$ pwd
shabhy4@shabhy4-Virtual-Platform:/etc$
```

Here I used `cd /etc` to go to the `/etc` directory, and `pwd` confirms my current location.

A screenshot of a Linux desktop environment, specifically Ubuntu 64-bit, running in a virtual machine. The desktop interface includes a dock with icons for Home, Dash, and a terminal window. The terminal window is active and shows the following command-line session:

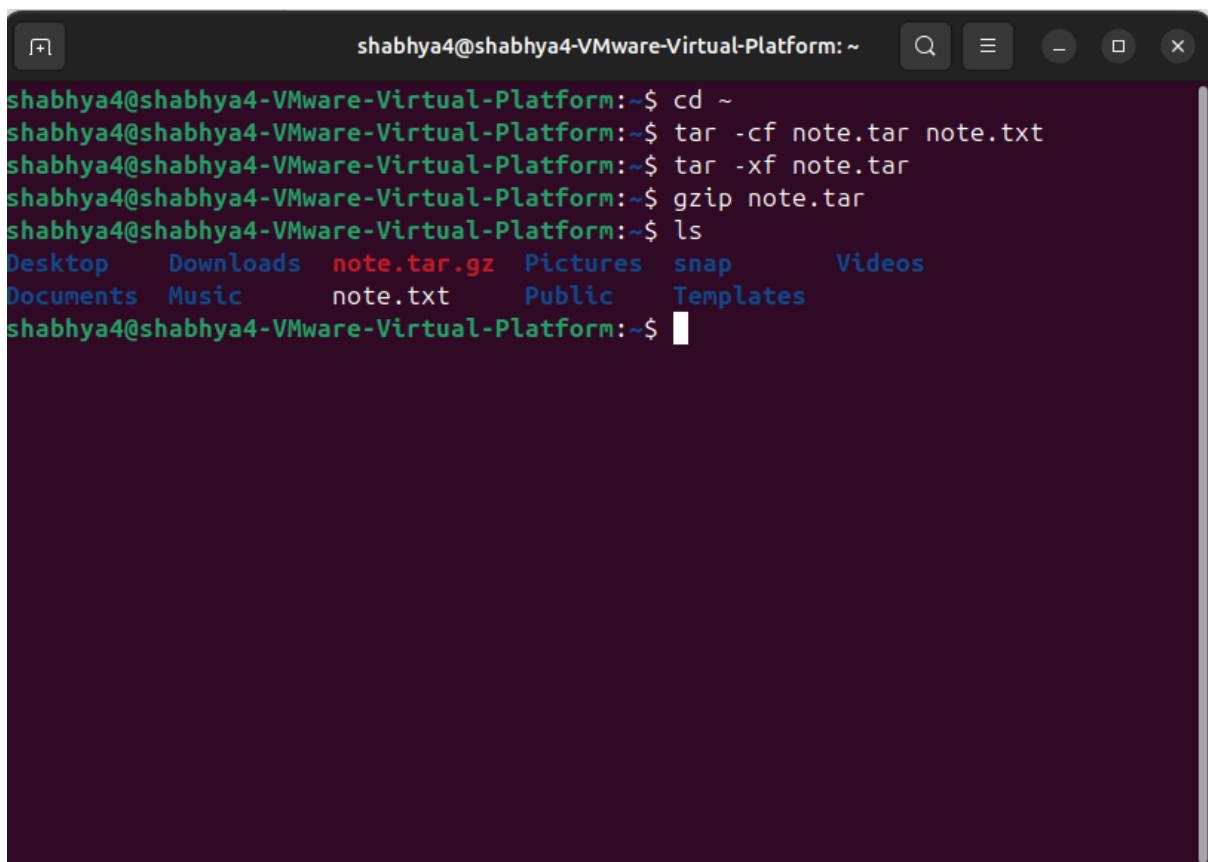
```
shabhy4@shabhy4-Virtual-Platform:~$ cd /etc  
pwd  
/etc  
shabhy4@shabhy4-Virtual-Platform:/etc$ cd ~  
pwd  
/home/shabhy4  
shabhy4@shabhy4-Virtual-Platform:~$
```

I Used cd ~ to go back to the home folder.

I. Also, in comparison to Windows, Linux has one tree starting at /, no drive letters.

II. /etc is used for system-wide configuration files.

- Compress files



A screenshot of a terminal window titled "shabhy4@shabhy4-VMware-Virtual-Platform:~". The terminal shows the following command sequence:

```
shabhy4@shabhy4-VMware-Virtual-Platform:~$ cd ~  
shabhy4@shabhy4-VMware-Virtual-Platform:~$ tar -cf note.tar note.txt  
shabhy4@shabhy4-VMware-Virtual-Platform:~$ tar -xf note.tar  
shabhy4@shabhy4-VMware-Virtual-Platform:~$ gzip note.tar  
shabhy4@shabhy4-VMware-Virtual-Platform:~$ ls  
Desktop Downloads note.tar.gz Pictures snap Videos  
Documents Music note.txt Public Templates  
shabhy4@shabhy4-VMware-Virtual-Platform:~$ █
```

I created a tar archive with `tar -cf note.tar note.txt`, extracted it with `tar -xf note.tar`, then compressed the archive with `gzip note.tar`, which produced `note.tar.gz`.

- **View processes**

The screenshot shows an htop session on a Linux system named shabhy4. At the top, it displays system statistics: CPU usage (0.6%), Tasks (118, 387), Load average (0.07, 0.03, 0.01), Uptime (01:18:08), and memory usage (Mem: 1.24G/3.78G, Swap: 0K/3.78G). Below this is a header for the process list:

Main	I/O	PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
7746	shabhy4	20	0	11572	5156	3608	R	2.0	0.1	0:01.29	htop		

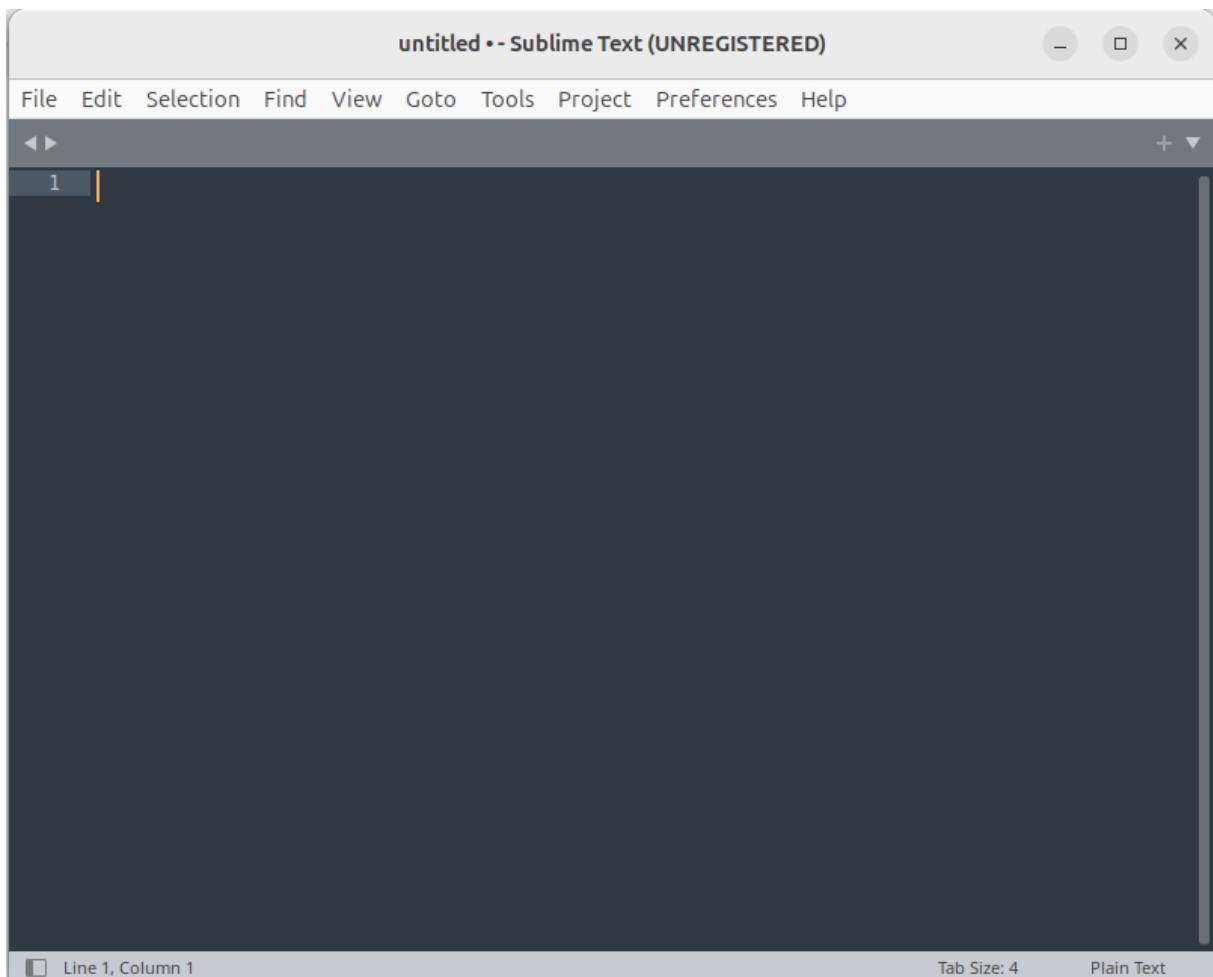
The main body of the table lists various processes, mostly owned by root, with their respective PID, user, priority (PRI), nice value (NI), virtual memory (VIRT), resident memory (RES), shared memory (SHR), state (S), CPU usage percentage (CPU%), memory usage percentage (MEM%), and command name. The processes include system daemons like vmto, init, and systemd, along with several vmware-related tasks.

At the bottom of the htop interface, there is a row of function keys:

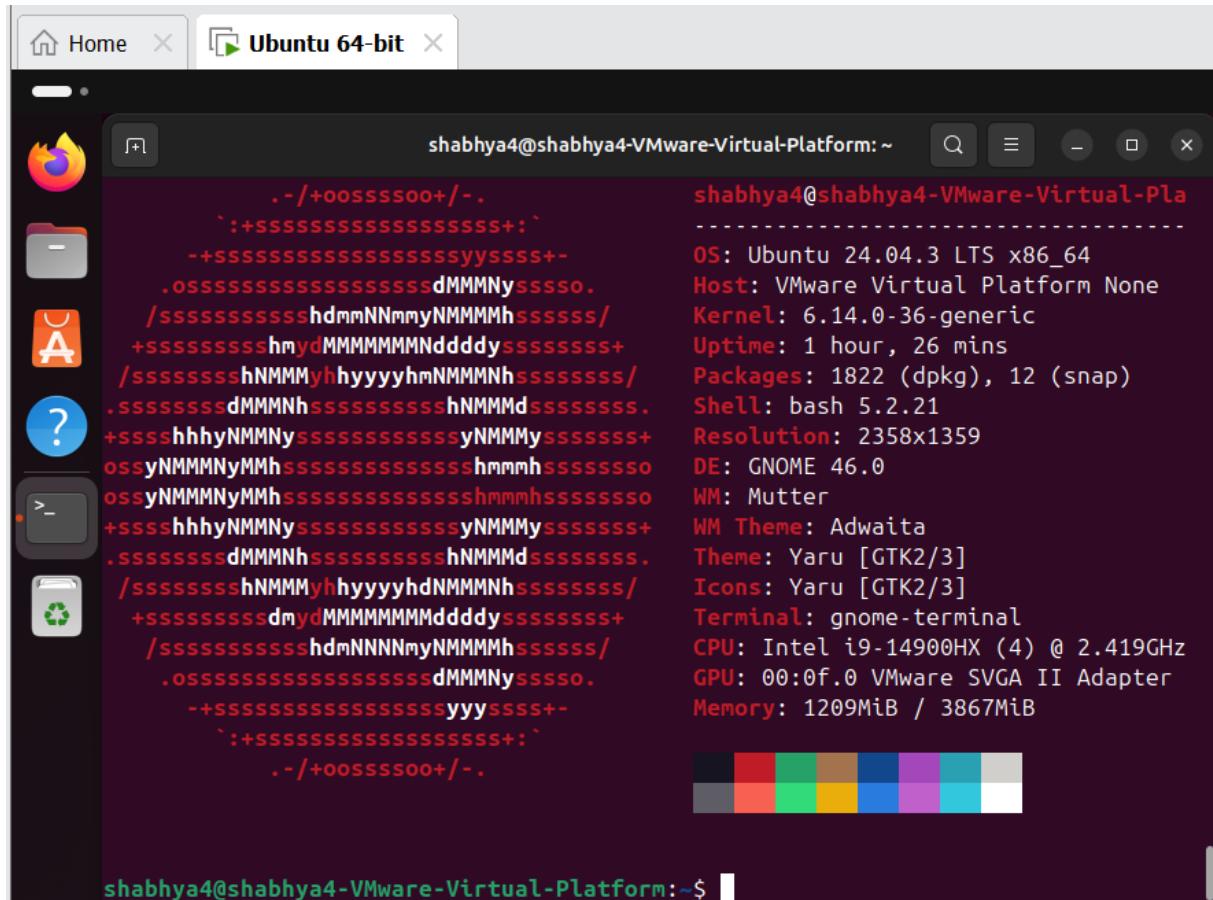
F1 Help F2 Setup F3 Search F4 Filter F5 Tree F6 Sort By F7 Nice - F8 Nice + F9 Kill F10 Quit

Here I launched htop, an interactive process viewer. It shows CPU and memory usage, system load, and all running processes with their PID, user, CPU% and MEM%.

- **Install Software**



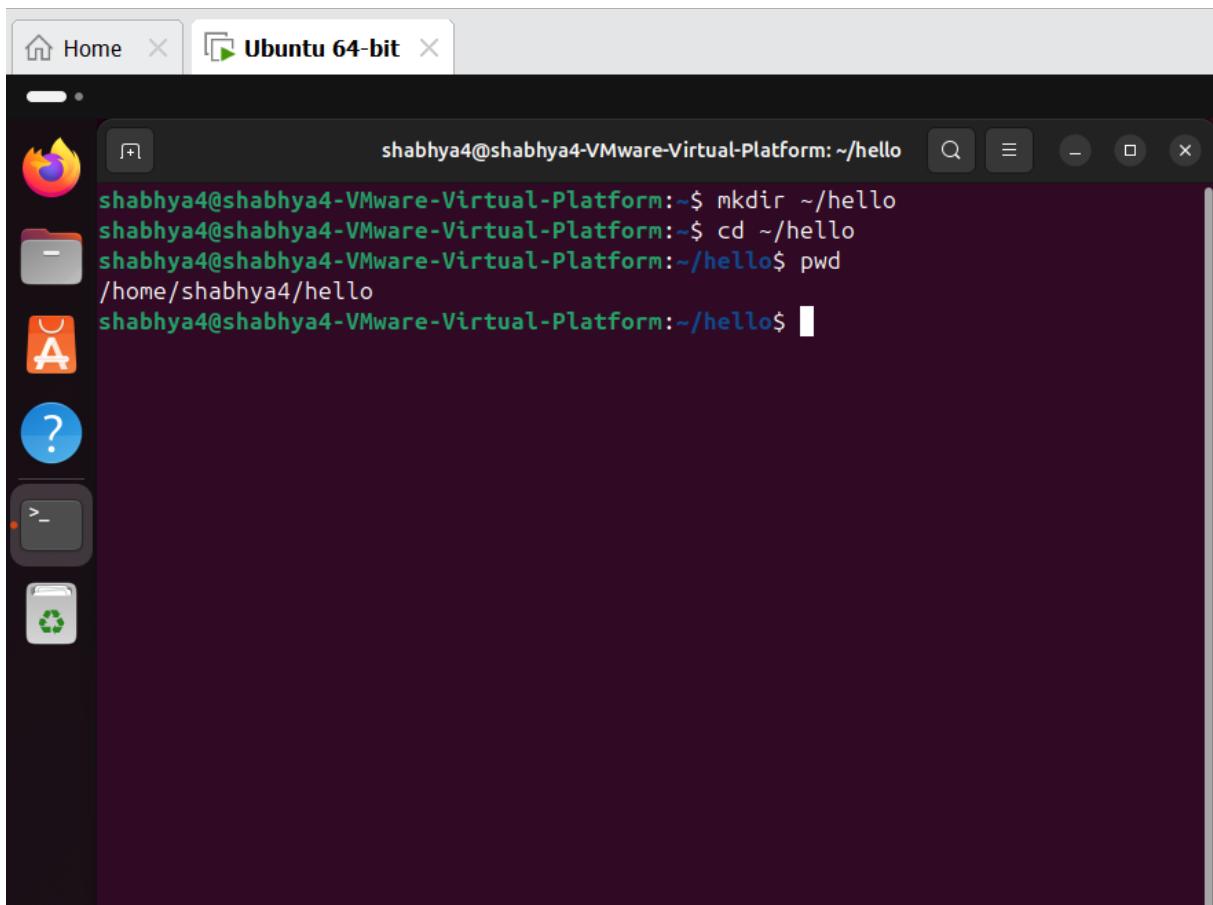
I installed Sublime Text via the Software Center and launched it.



I installed neofetch with `sudo apt install neofetch` and ran it. neofetch shows a summary of system information such as OS, kernel, uptime, number of packages, desktop environment, CPU, GPU, and memory usage.

Assignment 5.5: Users and permissions on Linux

Relevant screenshots + motivation



The screenshot shows a terminal window titled "Ubuntu 64-bit". The terminal window has a dark background and contains the following command history:

```
shabhy4@shabhy4-Virtual-Platform:~$ mkdir ~/hello
shabhy4@shabhy4-Virtual-Platform:~$ cd ~/hello
shabhy4@shabhy4-Virtual-Platform:~/hello$ pwd
/home/shabhy4/hello
shabhy4@shabhy4-Virtual-Platform:~/hello$
```

The terminal window is part of a desktop environment, as evidenced by the docked icons on the left side, which include a browser, file manager, terminal, help, and system status icons.

I created a new directory ~/hello and changed into it.

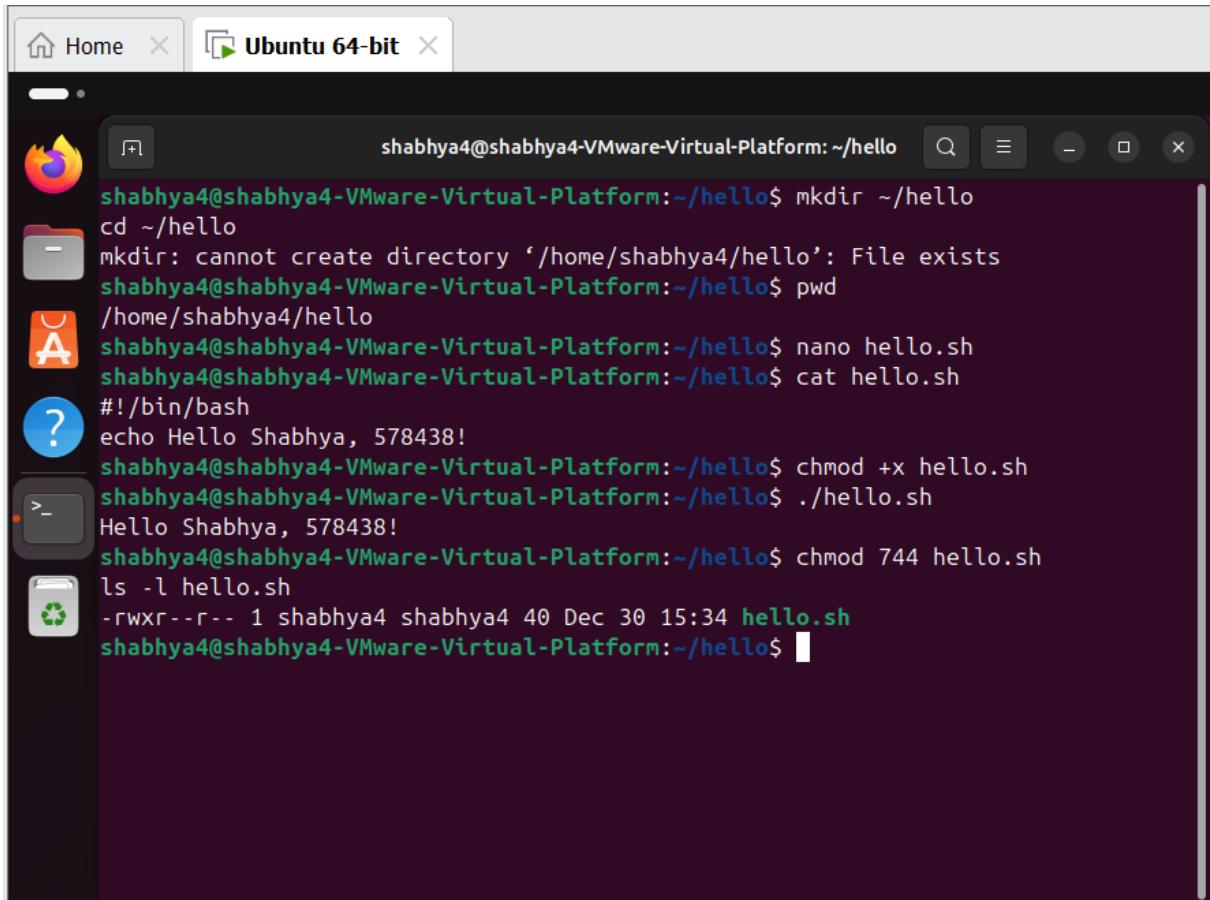
```
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ mkdir ~/hello
cd ~/hello
mkdir: cannot create directory '/home/shabhy4/hello': File exists
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ pwd
/home/shabhy4/hello
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ nano hello.sh
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ cat hello.sh
#!/bin/bash
echo Hello Shabhy4, 578438!
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$
```

Inside ~/hello I created hello.sh containing a bash shebang and a line that prints my name and student number.

The screenshot shows a terminal window titled "Ubuntu 64-bit" with the command line interface. The user, shabhy4, is in their home directory (~). They run the command `mkdir ~/hello` to create a new directory named "hello". When attempting to create a subdirectory within it, they receive an error message: "mkdir: cannot create directory '/home/shabhy4/hello': File exists". They then check the current directory with `pwd` and see they are in `/home/shabhy4/hello`. They edit a file named "hello.sh" using the nano editor with the command `nano hello.sh`. The contents of the file are a shebang line `#!/bin/bash` and the command `echo Hello Shabhy4, 578438!`. After saving the file, they change its permissions with `chmod +x hello.sh` and run it with `./hello.sh`, which outputs the greeting "Hello Shabhy4, 578438!".

```
shabhy4@shabhy4-Virtual-Platform:~/hello$ mkdir ~/
shabhy4@shabhy4-Virtual-Platform:~/hello$ cd ~/
shabhy4@shabhy4-Virtual-Platform:~/hello$ mkdir hello
shabhy4@shabhy4-Virtual-Platform:~/hello$ cd hello
shabhy4@shabhy4-Virtual-Platform:~/hello$ nano hello.sh
shabhy4@shabhy4-Virtual-Platform:~/hello$ cat hello.sh
#!/bin/bash
echo Hello Shabhy4, 578438!
shabhy4@shabhy4-Virtual-Platform:~/hello$ chmod +x hello.sh
shabhy4@shabhy4-Virtual-Platform:~/hello$ ./hello.sh
Hello Shabhy4, 578438!
```

With chmod +x hello.sh I added execute permission and ran the script with ./hello.sh, which prints my greeting



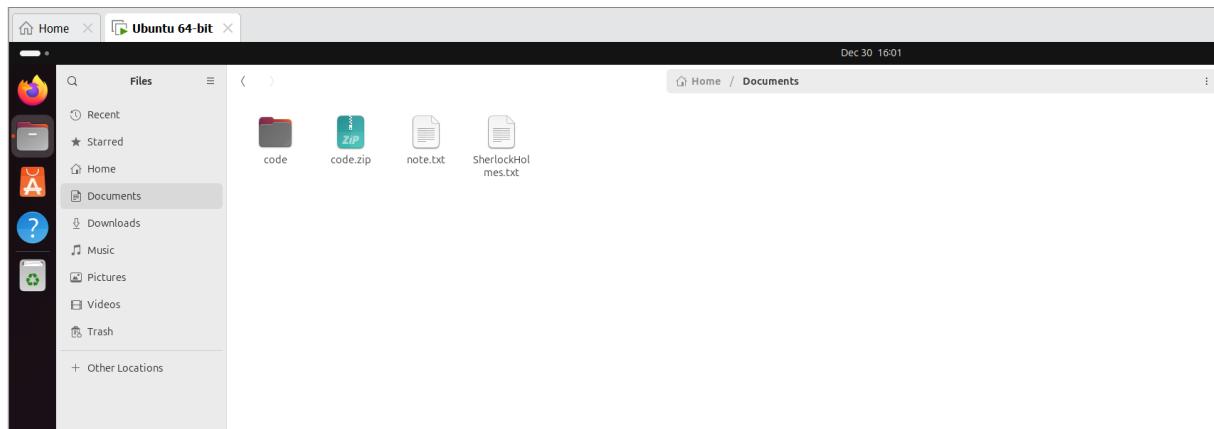
```
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ mkdir ~/hello
cd ~/hello
mkdir: cannot create directory '/home/shabhy4/hello': File exists
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ pwd
/home/shabhy4/hello
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ nano hello.sh
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ cat hello.sh
#!/bin/bash
echo Hello Shabhy4, 578438!
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ chmod +x hello.sh
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ ./hello.sh
Hello Shabhy4, 578438!
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$ chmod 744 hello.sh
ls -l hello.sh
-rwxr--r-- 1 shabhy4 shabhy4 40 Dec 30 15:34 hello.sh
shabhy4@shabhy4-VMware-Virtual-Platform:~/hello$
```

With `chmod 744 hello.sh` I set the permissions to `rwxr--r--`: the owner can read, write, and execute; group and others can only read the file.

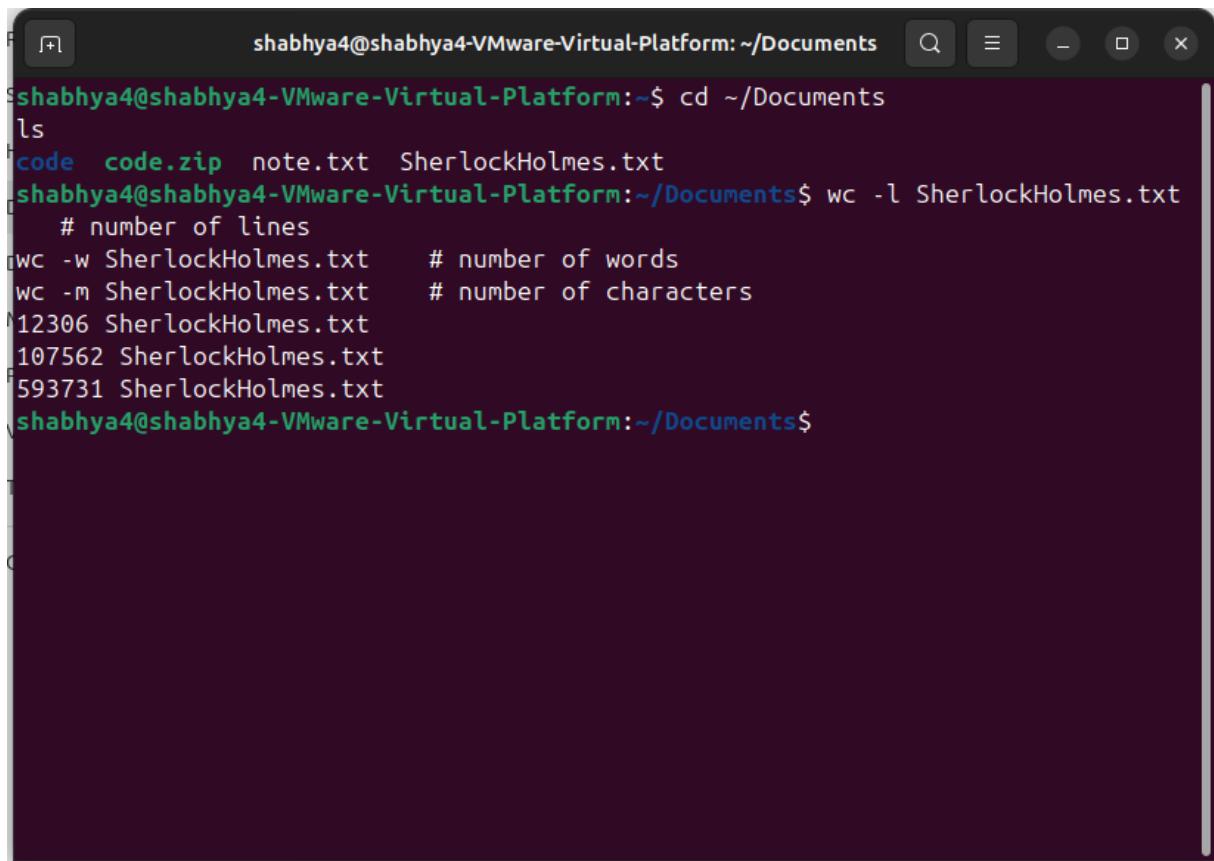
Assignment 5.6: View the contents of files

Relevant screenshots + motivation

- cat file: It prints the whole file to the terminal or concatenates several files.
- wc file: It counts lines, words, and bytes/characters.
- less file: It will open the file in a scrollable viewer page up/down, q to quit.
- head file: It shows the first 10 lines of a file “use head -n N for N lines”.
- tail file: It shows the last 10 lines of a file “use tail -n N for N lines”.
- grep pattern file: It prints lines matching pattern.
grep -n also shows the line number.



I downloaded the Sherlock Holmes text into my Documents folder.



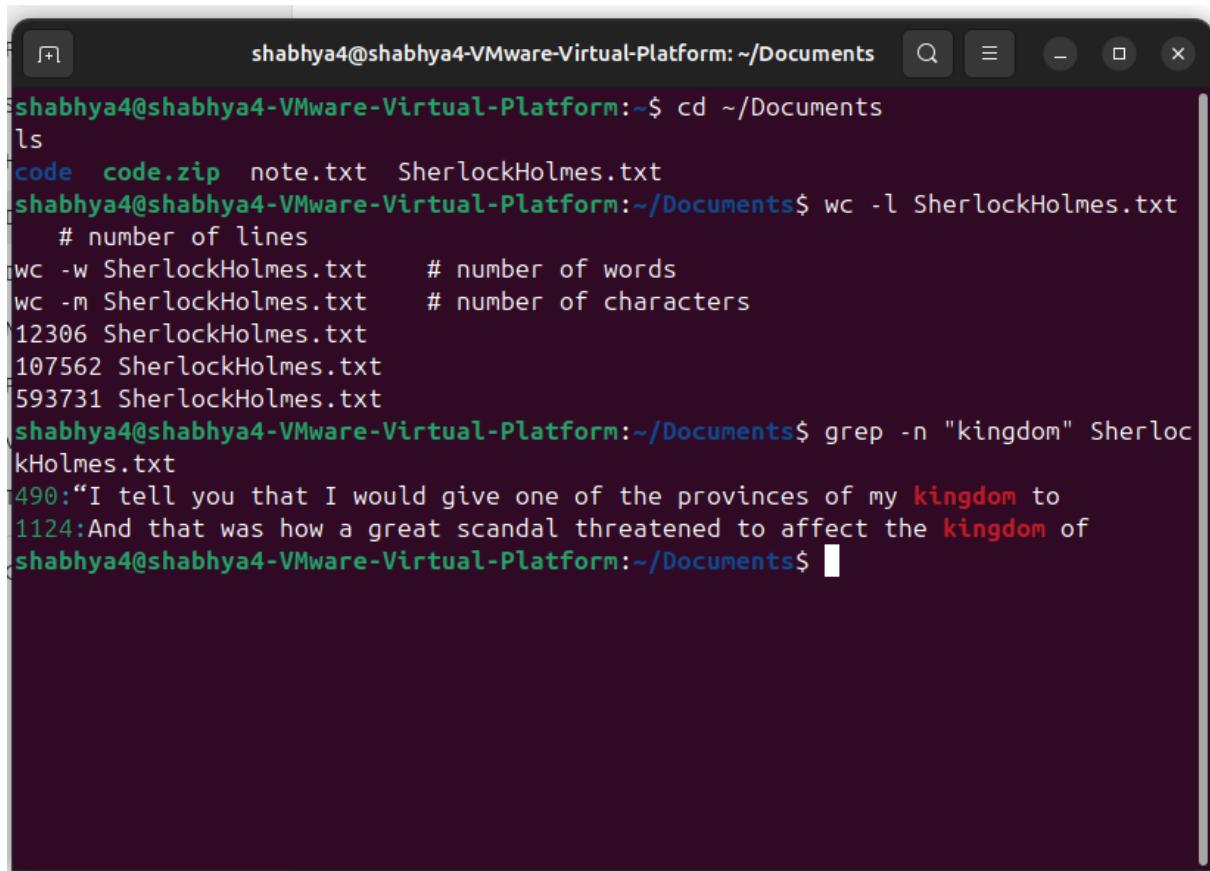
```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ cd ~/Documents
ls
code code.zip note.txt SherlockHolmes.txt
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ wc -l SherlockHolmes.txt
# number of lines
wc -w SherlockHolmes.txt      # number of words
wc -m SherlockHolmes.txt      # number of characters
12306 SherlockHolmes.txt
107562 SherlockHolmes.txt
593731 SherlockHolmes.txt
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$
```

Using `wc -l`, `wc -w` and `wc -m` I counted lines, words and characters in `SherlockHolmes.txt`.

It has 12306 lines

It has 107562 words

It has 593731 characters



The terminal window shows the following session:

```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents
shabhy4@shabhy4-VMware-Virtual-Platform:~$ cd ~/Documents
ls
code code.zip note.txt SherlockHolmes.txt
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ wc -l SherlockHolmes.txt
# number of lines
wc -w SherlockHolmes.txt      # number of words
wc -m SherlockHolmes.txt      # number of characters
12306 SherlockHolmes.txt
107562 SherlockHolmes.txt
593731 SherlockHolmes.txt
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ grep -n "kingdom" SherlockHolmes.txt
490:"I tell you that I would give one of the provinces of my kingdom to
1124:And that was how a great scandal threatened to affect the kingdom of
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$
```

The word “kingdom” appears on lines 490 and 1124.

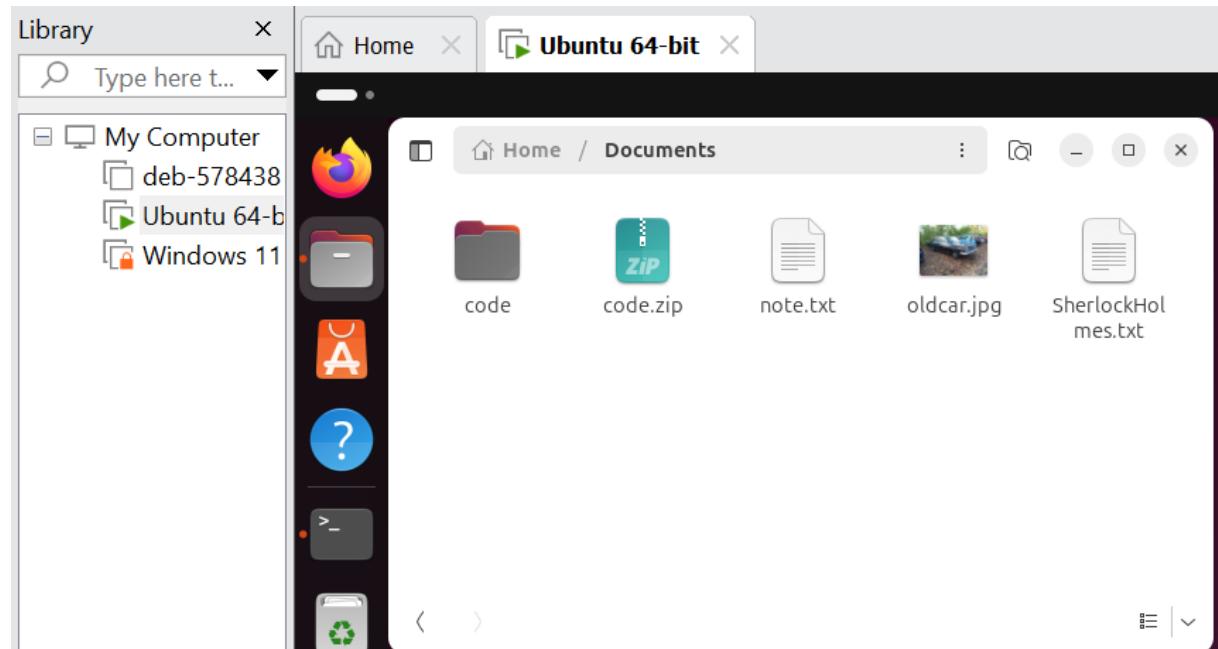
```
shabhya4@shabhya4-VMware-Virtual-Platform:~/Documents$ grep -n "kingdom" SherlockHolmes.txt
490:"I tell you that I would give one of the provinces of my kingdom to
1124:And that was how a great scandal threatened to affect the kingdom of
shabhya4@shabhya4-VMware-Virtual-Platform:~/Documents$ head -n 500 SherlockHolmes.txt
| tail -n 21
"Then I shall drop you a line to let you know how we progress."
"Pray do so. I shall be all anxiety."
"Then, as to money?"
"You have _carte blanche_."
"Absolutely?"
"I tell you that I would give one of the provinces of my kingdom to
have that photograph."
"And for present expenses?"
The King took a heavy chamois leather bag from under his cloak and laid
it on the table.
"There are three hundred pounds in gold and seven hundred in notes," he
said.

shabhya4@shabhya4-VMware-Virtual-Platform:~/Documents$
```

I used grep -n "kingdom" to find the line numbers (490 and 1124), then head -n 500 SherlockHolmes.txt | tail -n 21 to display 10 lines above and below the first occurrence on line 490.

Assignment 5.7: Digital forensics

Relevant screenshots + motivation



oldcar.jpg in Documents folder.

```
Type here ...
```

```
Dec 30 10:24
shahyad@shahyad-VirtualPlatform: ~/Documents
```

```
Processing triggers for liblc-bl (2.39-0ubuntu8.6) ...
shahyad@shahyad-VirtualPlatform: ~/Documents$ cd ~/Documents
shahyad@shahyad-VirtualPlatform: ~/Documents$ exiftool oldcar.jpg
ExifTool Version Number : 12.76
File Name               : oldcar.jpg
File Size                : 2.4 MB
File Modification Date/Time : 2025:12:30 16:21:26+01:00
File Access Date/Time   : 2025:12:30 16:21:26+01:00
File Change Date/Time   : 2025:12:30 16:21:26+01:00
File Permissions        : rwxr--r--
File Type                : JPEG
File Type Extension     : jpg
File Type Mime           : image/jpeg
File Version              : 1.01
Exif Byte Order          : Big-endian (Motorola, MM)
Make                     : motorola
Camera Model Name       : moto g(6) play
X Resolution             : 72
Y Resolution             : 72
Resolution Unit         : lmacs
Software Name            : 1motorola_9_PPP529.55-35-18-7_6a0d0 release-keys
Modify Date              : 2020:11:07 15:08:57
YCbCr Positioning       : Centered
Exposure Time           : 1/33
F Number                 : 1.9
Exposure Program        : Program AE
ISO                      : 64
Exif Version             : 0220
Date/Time Original      : 2020:11:07 15:08:57
Create Date              : 2020:11:07 15:08:57
Components Configuration: Y, Cb, Cr, -
Shutter Speed Value     : 1/33
Aperture Value          : 1.9
Brightness Value        : 0
Exposure Compensation   : 0
Max Aperture Value      : 1.9
Histogram Mode          : Center-weighted average
Flash                   : Auto, Did not fire
Focal Length             : 3.5 mm
Build Number             : BACK.mot_5831B
Sensor Model            : 140ctx018
Manufacture Date        : Flashpix Version
Color Space              : sRGB
Exif Image Width         : 4160
Exif Image Height        : 3120
Interoperability Index  : R98 - DCF basic file (<sRGB>)
Interoperability Version: 0100
Scene Type               : Directly photographed
Custom Rendered          : Normal
Exposure Mode            : Auto
White Balance             : Auto
Digital Zoom Ratio       : 1
Scene Capture Type       : Standard
Contrast                 : Normal
Saturation               : Low
Sharpness                : Soft
```

The photo was taken with a Motorola moto g(6) play phone.

```
library deb-578438
Home Ubuntu 64-bit Type here L... Dec 30 10:27
My Computer shabby4@shabby4-Virtual-Platform: /Documents
Ubuntu 64-bit
Ubuntu 64-bit Components Configuration Create Date : 2020:11:07 15:08:57
Components Configuration : Y, Cb, Cr
Shutter Speed Value : 1/33
Aperture Value : 2.0
Exposure Value : 1.0
Exposure Compensation : 0
Max Aperture Value : 2.0
Metering Mode : Center-weighted average
Thumbnail Image : Astro, did not fire
Focal Length : 3.5 mm
Build Number : PPPS29-55-35-1B-7
Sensor Size : BACK_MOT_55K3LB
HDMI Feature Data : 1400010010101
Flashpix Version : B100
Color Space : sRGB
EXIF Image Width : 4000
EXIF Image Height : 3120
Interoperability Index : 898 - DCF basic file (sRGB)
Interoperability Version : B100
Scene Type : Directly photographed
Camera Model Name : Sony Alpha a7C
Exposure Mode : Auto
White Balance : Auto
Digital Zoom Ratio : 1
Scene Capture Type : Standard
Contrast : Normal
Saturation : Low
Sharpness : Soft
GPS Information ID : 2.2.6.0
GPS Latitude Ref : North
GPS Longitude Ref : East
GPS Altitude Ref : Above Sea Level
GPS Altitude Stamp : 14000107
GPS Map Datum : WGS-84
GPS Processing Method : ASCII
GPS Date Stamp : 2020:11:07
Comments : 1976 (old-style)
Thumbnail Offset : 2862
Thumbnail Length : 59453
Image Width : 4000
Image Height : 3120
Encoding Process : Baseline DCT, Huffman coding
Bits Per Sample : 8
Color Components : 3
YCbCr4:2:0 (2,2)
Aperture : 2.0
Image Size : 1400x3120
Megapixels : 1.0
Shutter Speed : 1/33
Thumbnail Image : (Binary data 59453 bytes, use -b option to extract)
GPS Altitude : 42 m Above Sea Level
GPS AltitudeStamp : 2020:11:07 14:05:57Z
GPS Latitude : 33 deg 11' 39.68" N
GPS Longitude : 6 deg 32' 12.98" E
Focal Length : 3.5 mm
GPS LightValue : 53 deg 11' 39.68" N, 6 deg 32' 12.98" E
Light Value : 1.7
shabby4@shabby4-Virtual-Platform: /Documents$
```

- GPS Latitude Ref North
 - GPS Longitude Ref East
 - GPS Altitude 42 m Above Sea Level
 - GPS Date/Time 2020:11:07 14:08:57Z
 - GPS Latitude 53 deg 19' 39.68" N
 - GPS Longitude 6 deg 32' 12.90" E

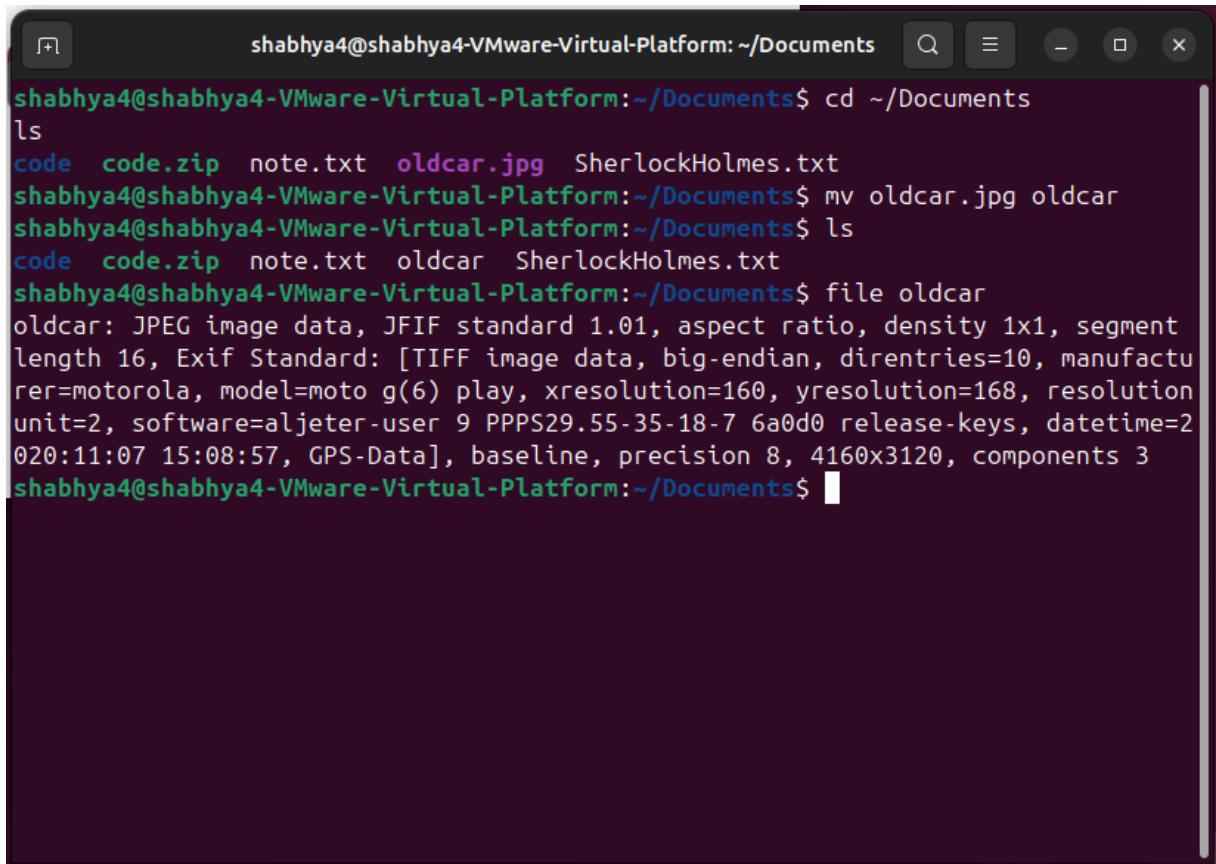
Yes, GPS coordinates are known

Latitude: 53° 19' 39.68" N

Longitude: $6^{\circ} 32' 12.90'' E$

Altitude: 42 m above sea level

The GPS coordinates (53°19'39.68"N, 6°32'12.90"E, altitude 42 m) point to a rural area in the Netherlands, in the province of Groningen, near the village of Harnem, municipality of Eemsdelta.

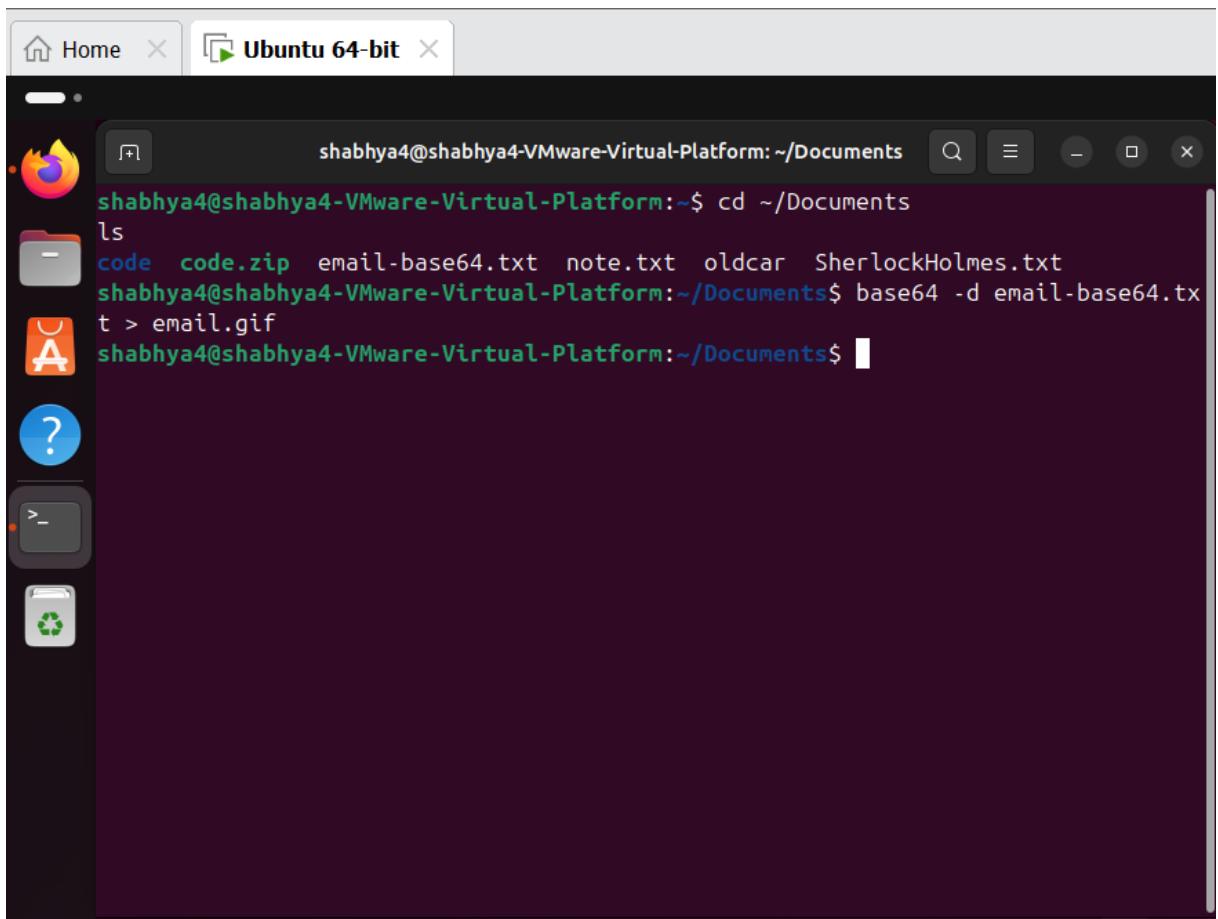


A screenshot of a terminal window titled "shabhy4@shabhy4-VMware-Virtual-Platform: ~/Documents". The terminal shows the following command sequence:

```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ cd ~/Documents
ls
code code.zip note.txt oldcar.jpg SherlockHolmes.txt
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ mv oldcar.jpg oldcar
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ ls
code code.zip note.txt oldcar SherlockHolmes.txt
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ file oldcar
oldcar: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment
length 16, Exif Standard: [TIFF image data, big-endian, direntries=10, manufac-
rer=motorola, model=moto g(6) play, xresolution=160, yresolution=168, resolution
unit=2, software=aljeter-user 9 PPPS29.55-35-18-7 6a0d0 release-keys, datetime=2
020:11:07 15:08:57, GPS-Data], baseline, precision 8, 4160x3120, components 3
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$
```

Fig: **Filename extension**

I renamed oldcar.jpg to oldcar (removed the extension). The file command still identifies it as 'JPEG image data', so Ubuntu still treats it as a JPEG. Linux uses the file's internal header (magic bytes), not just the .jpg extension.



```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents
```

```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ cd ~/Documents
```

```
ls
```

```
code code.zip email-base64.txt note.txt oldcar SherlockHolmes.txt
```

```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ base64 -d email-base64.txt > email.gif
```

```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$
```

email-base64.txt is listed in Documents.

- base64: It program to encode/decode Base64 data.
- -d or –decode: It decode instead of encode.
- email-base64.txt: It input text file.
- > : It redirect the binary output into a new file.
- email.gif: the output GIF image file.

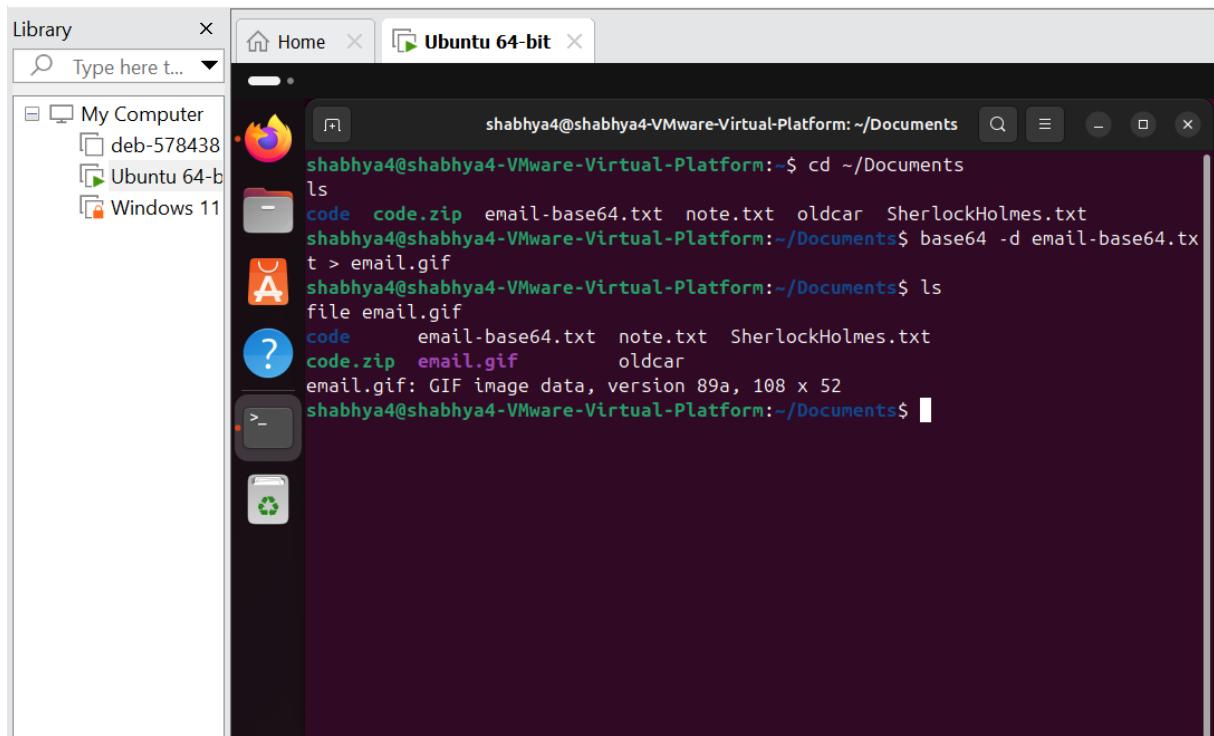


Fig: **BASE64 decoding**

I decoded the BASE64 string with `base64 -d email-base64.txt > email.gif`. The `-d` option tells `base64` to decode, and `>` redirects the binary output to the file `email.gif`. The `file` command shows that `email.gif` is GIF image data, and opening it in the image viewer displays the image

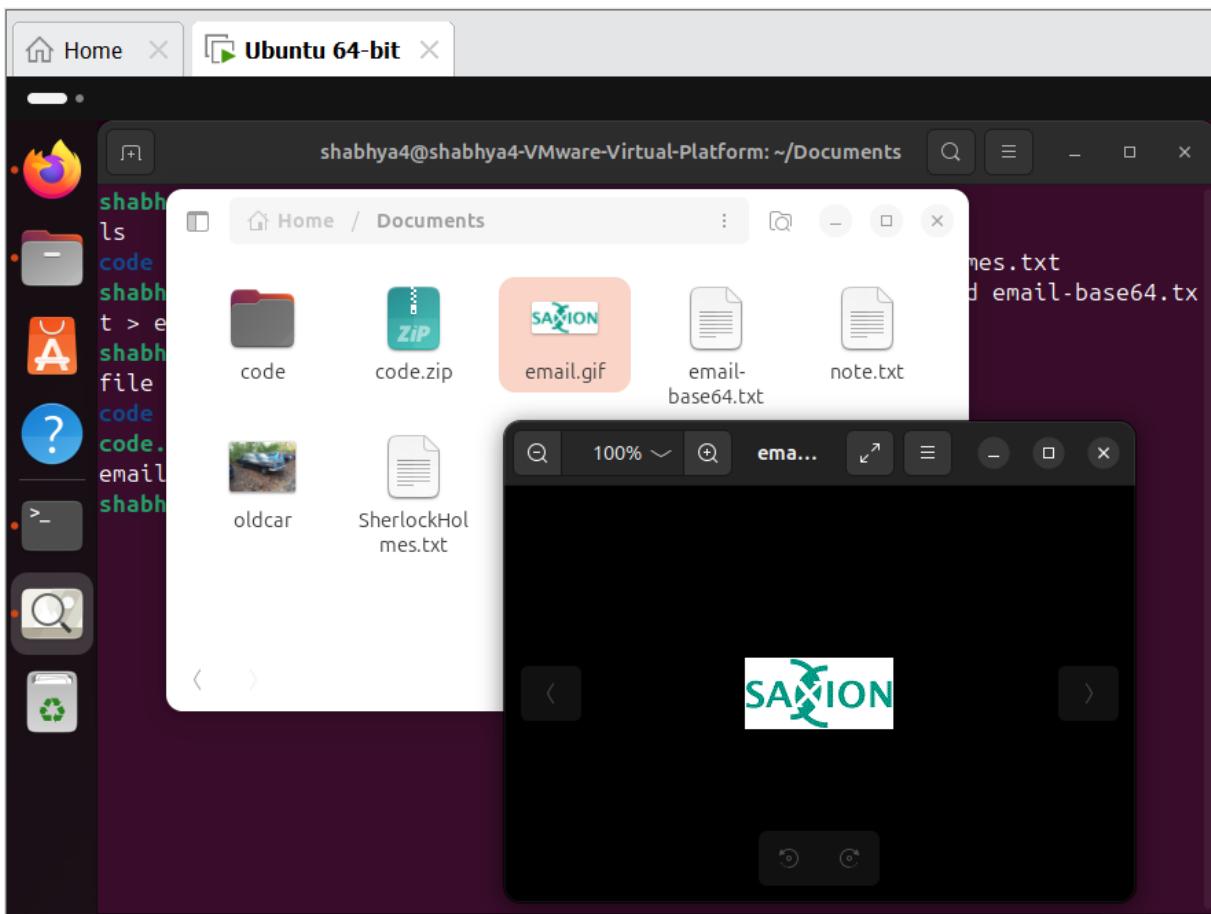


Fig: Screenshot of the email GIF open in the image viewer

Assignment 5.8: Steganography

Relevant screenshots + motivation

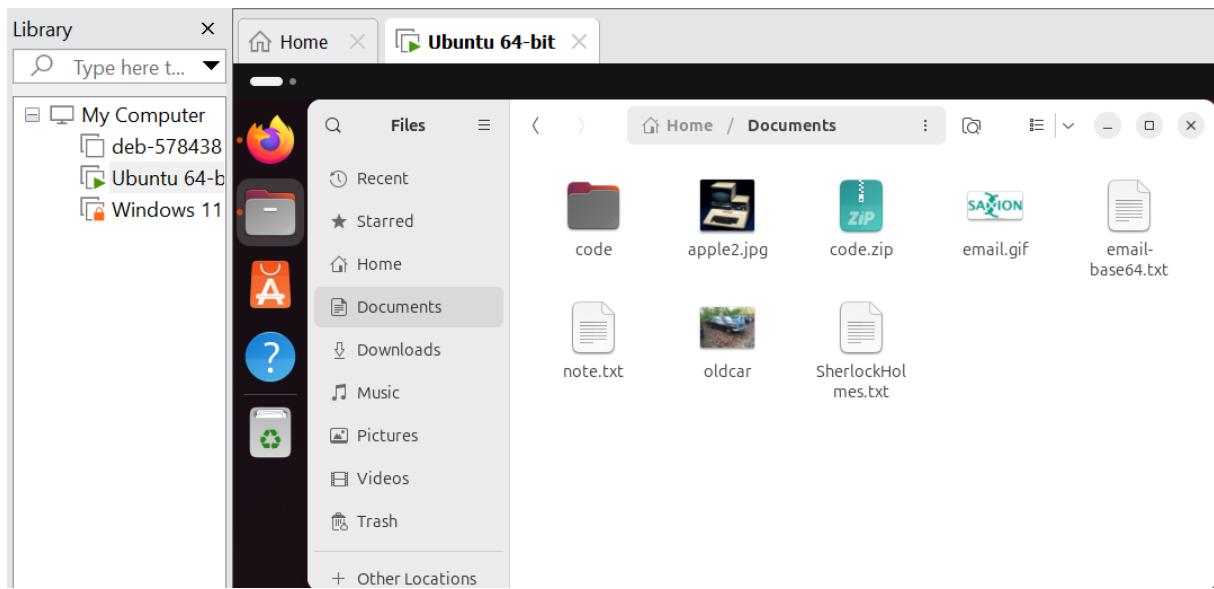


Fig: I saved apple2.jpg in my Documents folder.

```
Reading state information... Done
steghide is already the newest version (0.5.1-15).
The following package was automatically installed and is no longer required:
  liblvm19
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 119 not upgraded.
shabhy4@shabhy4-VMware-Virtual-Platform:~$ steghide --help
steghide version 0.5.1

the first argument must be one of the following:
embed, --embed      embed data
extract, --extract  extract data
info, --info        display information about a cover- or stego-file
info <filename>    display information about <filename>
encinfo, --encinfo  display a list of supported encryption algorithms
version, --version   display version information
license, --license  display steghide's license
help, --help        display this usage information

embedding options:
-ef, --embedfile    select file to be embedded
-ef <filename>     embed the file <filename>
-cf, --coverfile   select cover-file
-cf <filename>     embed into the file <filename>
-p, --passphrase    specify passphrase
-p <passphrase>    use <passphrase> to embed data
-sf, --stegofile   select stego file
-sf <filename>     write result to <filename> instead of cover-file
-e, --encryption   select encryption parameters
-e <a>[<m>]<m>[<a>] specify an encryption algorithm and/or mode
-e none            do not encrypt data before embedding
-z, --compress     compress data before embedding (default)
-z <l>             using level <l> (1 best speed...9 best compression)
-Z, --dontcompress do not compress data before embedding
-K, --nochecksum   do not embed crc32 checksum of embedded data
-N, --dontembedname do not embed the name of the original file
-f, --force         overwrite existing files
-q, --quiet        suppress information messages
-v, --verbose      display detailed information

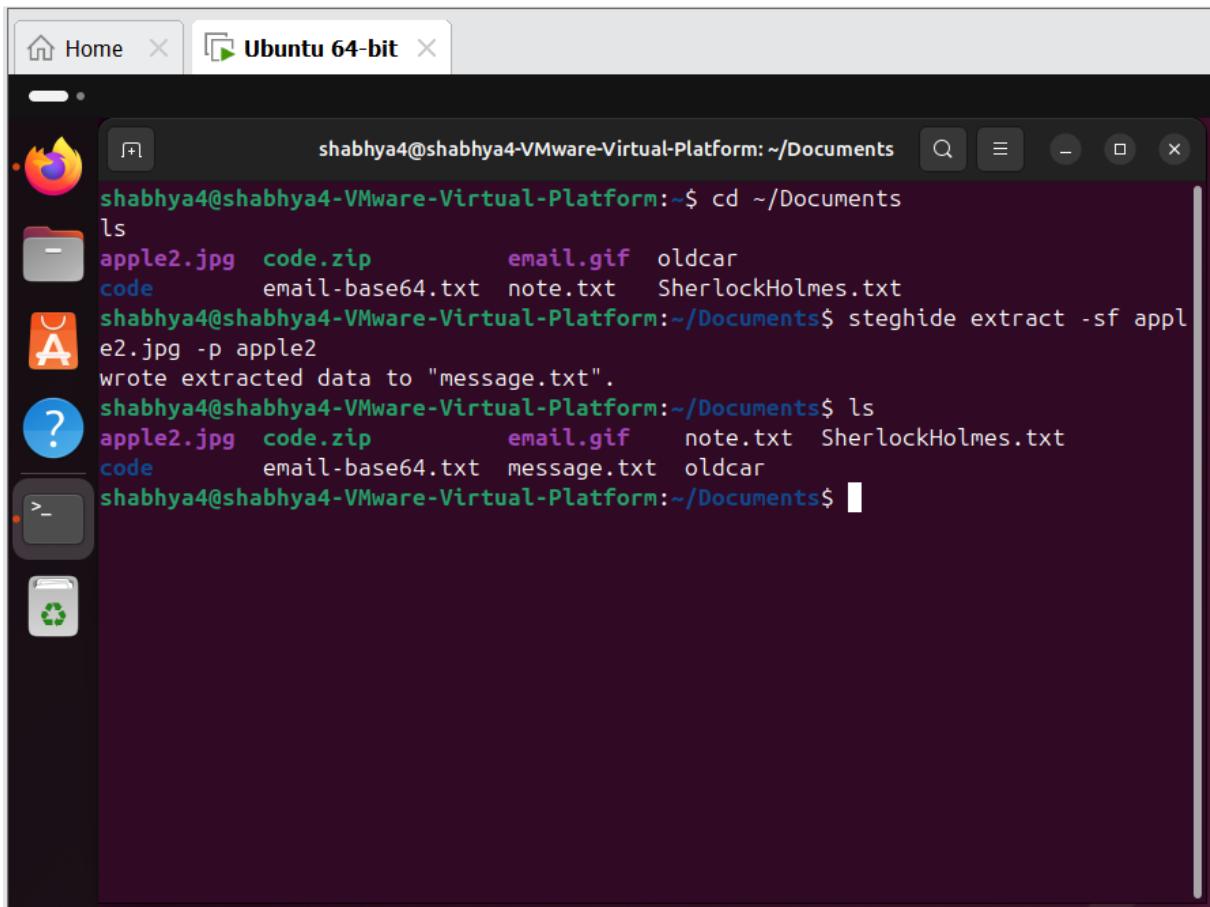
extracting options:
-sf, --stegofile   select stego file
-sf <filename>     extract data from <filename>
-p, --passphrase    specify passphrase
-p <passphrase>    use <passphrase> to extract data
-xf, --extractfile select file name for extracted data
-xf <filename>     write the extracted data to <filename>
-f, --force         overwrite existing files
-q, --quiet        suppress information messages
-v, --verbose      display detailed information

options for the info command:
-p, --passphrase    specify passphrase
-p <passphrase>    use <passphrase> to get info about embedded data

To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
To extract embedded data from stg.jpg: steghide extract -sf stg.jpg
shabhy4@shabhy4-VMware-Virtual-Platform:~$
```

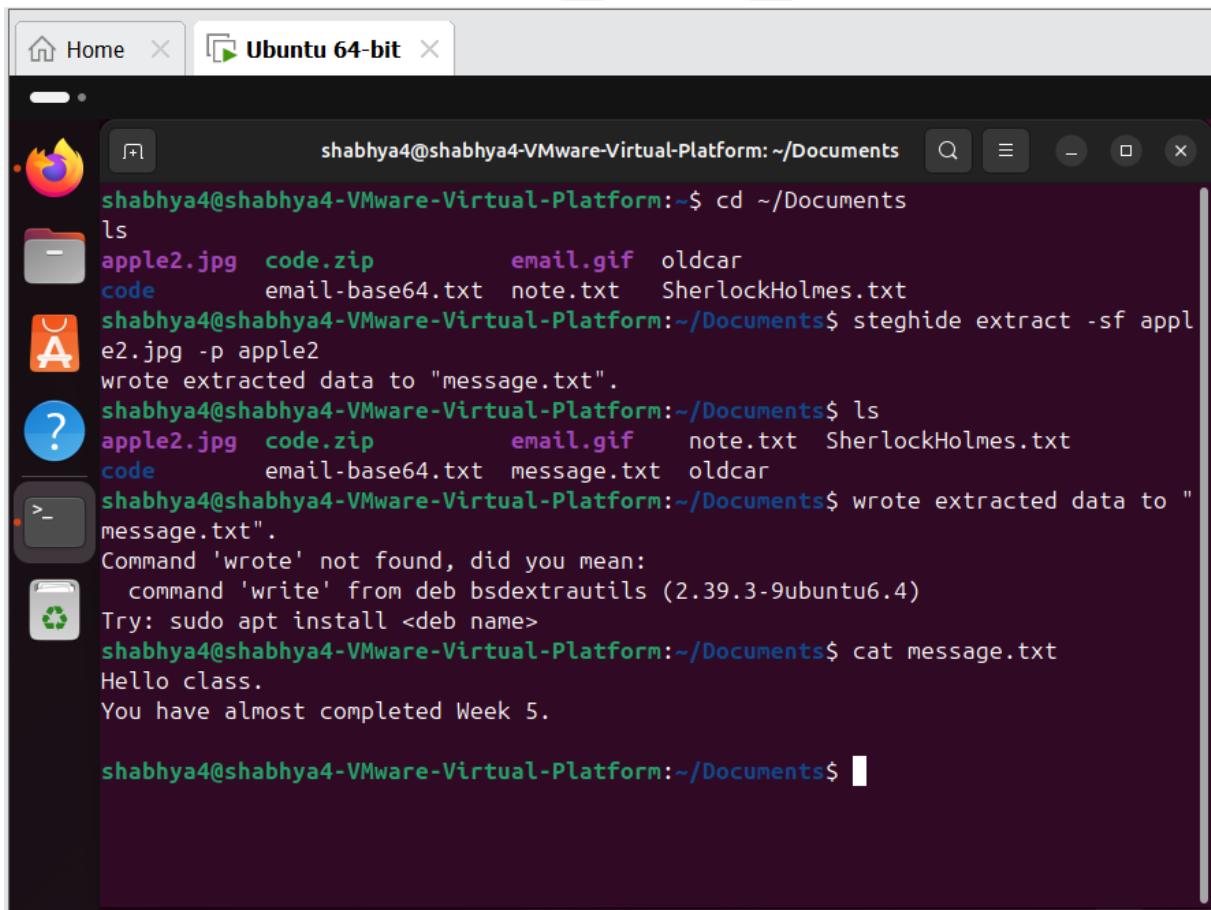
Fig: Terminal showing the output of steghide --help

I used steghide --help to see how to extract hidden data (steghide extract -sf file -p password)

A screenshot of a Linux terminal window titled "Ubuntu 64-bit". The terminal is running on a virtual machine with the host name "shabhy4@shabhy4-VMware-Virtual-Platform". The user is in the "/Documents" directory. They run the command "steghide extract -sf apple2.jpg -p apple2" to extract hidden data from the image "apple2.jpg". The terminal output shows the command being run, followed by the message "wrote extracted data to \"message.txt\".", and finally a list of files in the directory: "apple2.jpg", "code.zip", "email.gif", "note.txt", "oldcar", "code", "email-base64.txt", "message.txt", and "SherlockHolmes.txt".

```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ cd ~/Documents
ls
apple2.jpg  code.zip      email.gif  oldcar
code        email-base64.txt  note.txt  SherlockHolmes.txt
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ steghide extract -sf apple2.jpg -p apple2
wrote extracted data to "message.txt".
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ ls
apple2.jpg  code.zip      email.gif  note.txt  SherlockHolmes.txt
code        email-base64.txt  message.txt  oldcar
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$
```

Using steghide extract -sf apple2.jpg -p apple2 I extracted the hidden text file from the image; steghide reports wrote extracted data to "..." and the new file appears in the directory



The screenshot shows a terminal window titled "Ubuntu 64-bit" running on a virtual machine. The terminal session is as follows:

```
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ cd ~/Documents
ls
apple2.jpg  code.zip      email.gif  oldcar
code        email-base64.txt note.txt  SherlockHolmes.txt
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ steghide extract -sf appl
e2.jpg -p apple2
wrote extracted data to "message.txt".
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ ls
apple2.jpg  code.zip      email.gif  note.txt  SherlockHolmes.txt
code        email-base64.txt message.txt  oldcar
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ wrote extracted data to "
message.txt".
Command 'wrote' not found, did you mean:
  command 'write' from deb bsdextrautils (2.39.3-9ubuntu6.4)
Try: sudo apt install <deb name>
shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$ cat message.txt
Hello class.
You have almost completed Week 5.

shabhy4@shabhy4-VMware-Virtual-Platform:~/Documents$
```

I used command line tool steghide --help to extract this text file

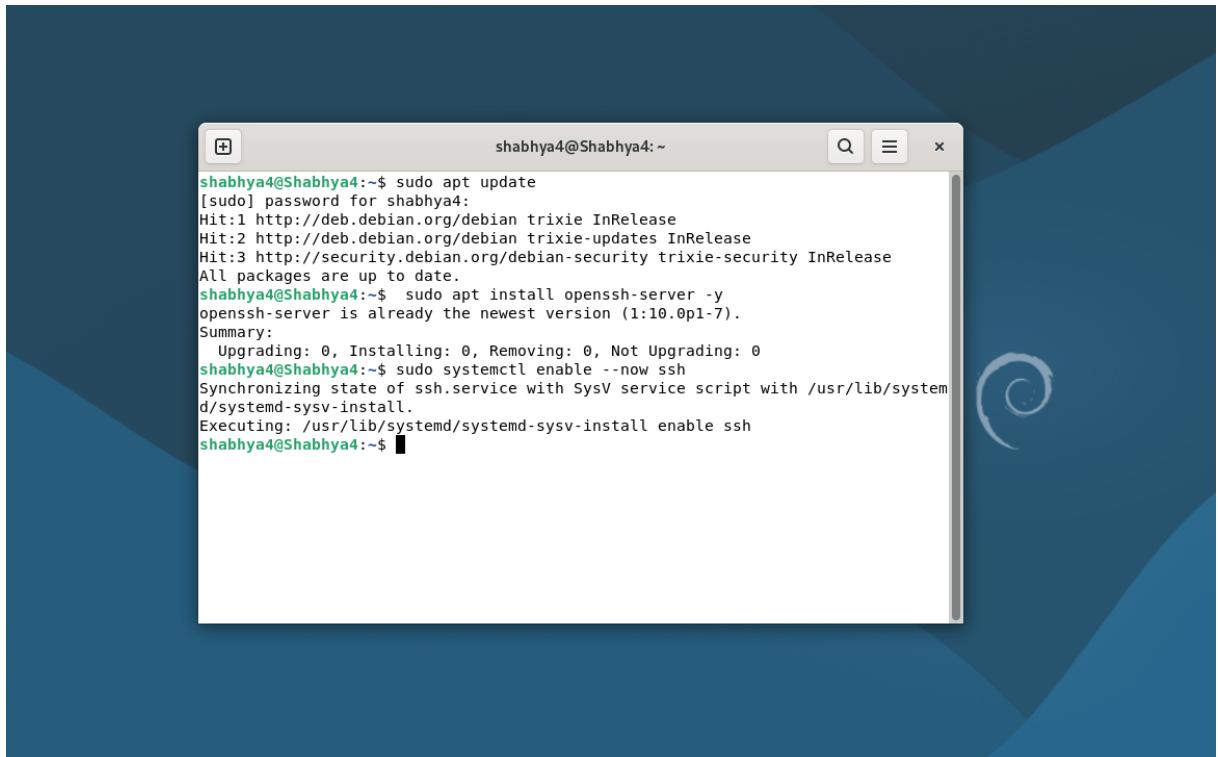
Which says Hello class.

You have almost completed week 5.

Assignment 5.9: Capture disk images

Make relevant screenshots + motivation:

- Proof that the Debian 13 server stored a back-up image of the Ubuntu 24.04 Desktop VM.



The screenshot shows a terminal window titled "shabhy4@Shabhy4:~". The terminal displays the following command-line session:

```
shabhy4@Shabhy4:~$ sudo apt update
[sudo] password for shabhy4:
Hit:1 http://deb.debian.org/debian trixie InRelease
Hit:2 http://deb.debian.org/debian trixie-updates InRelease
Hit:3 http://security.debian.org/debian-security trixie-security InRelease
All packages are up to date.
shabhy4@Shabhy4:~$ sudo apt install openssh-server -y
openssh-server is already the newest version (1:10.0p1-7).
Summary:
Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
shabhy4@Shabhy4:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
shabhy4@Shabhy4:~$
```

Fig. 1: Installing and enabling the SSH server on the Debian image server

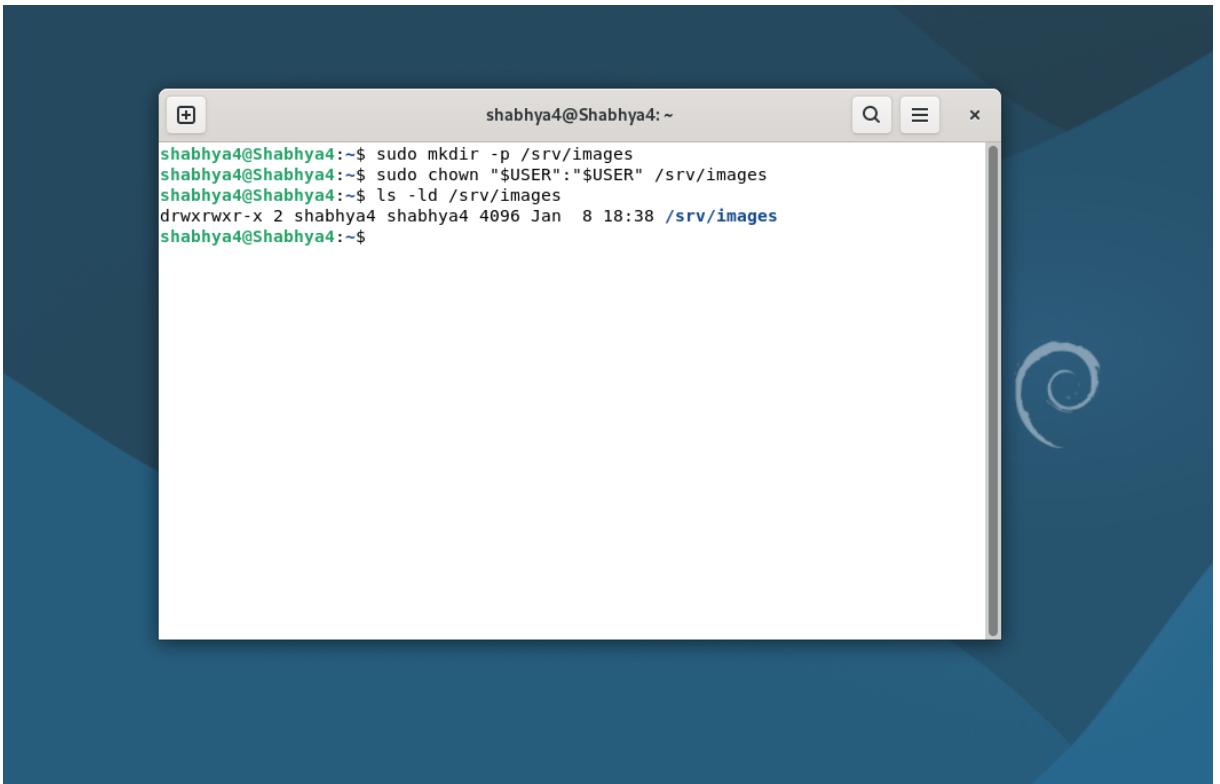


Fig. 2: Creating the /srv/images directory on the Debian image server and changing its ownership to my user so it can store the VM backup images.

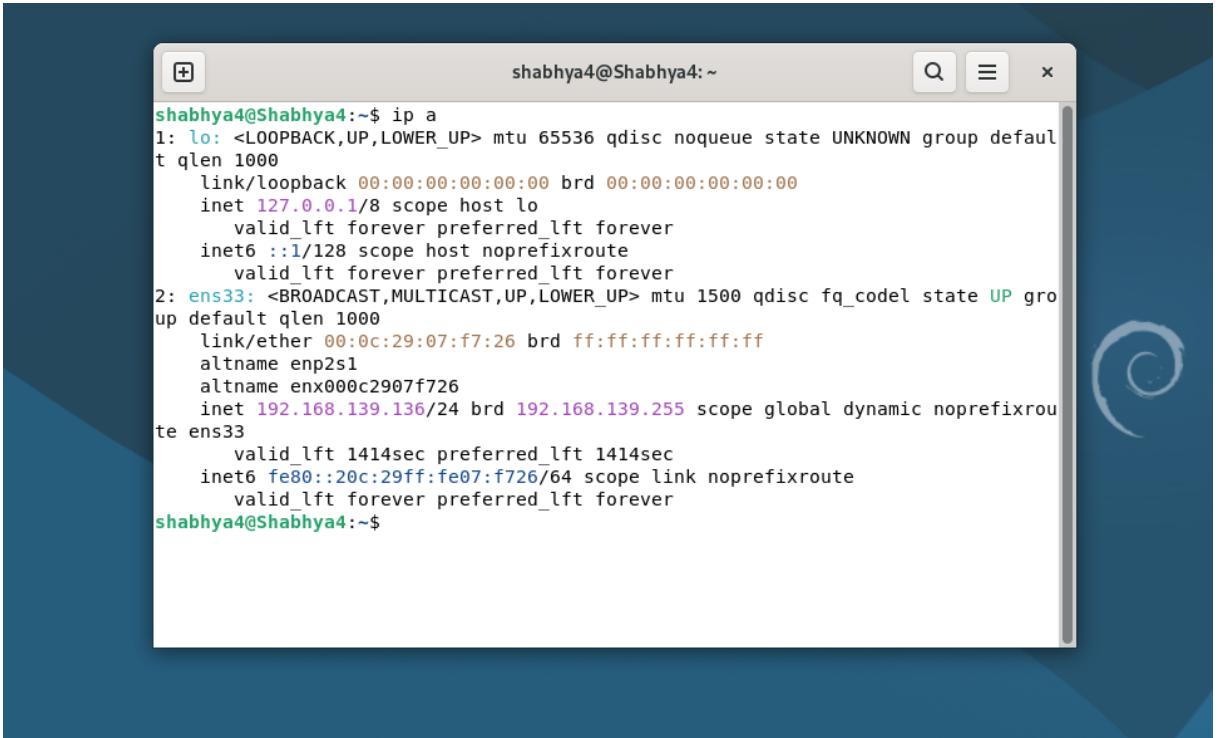


Fig. 3: Checking the IP configuration of the Debian image server with ip a; interface ens33 has address 192.168.139.136/24 on the 192.168.139.0/24 network.

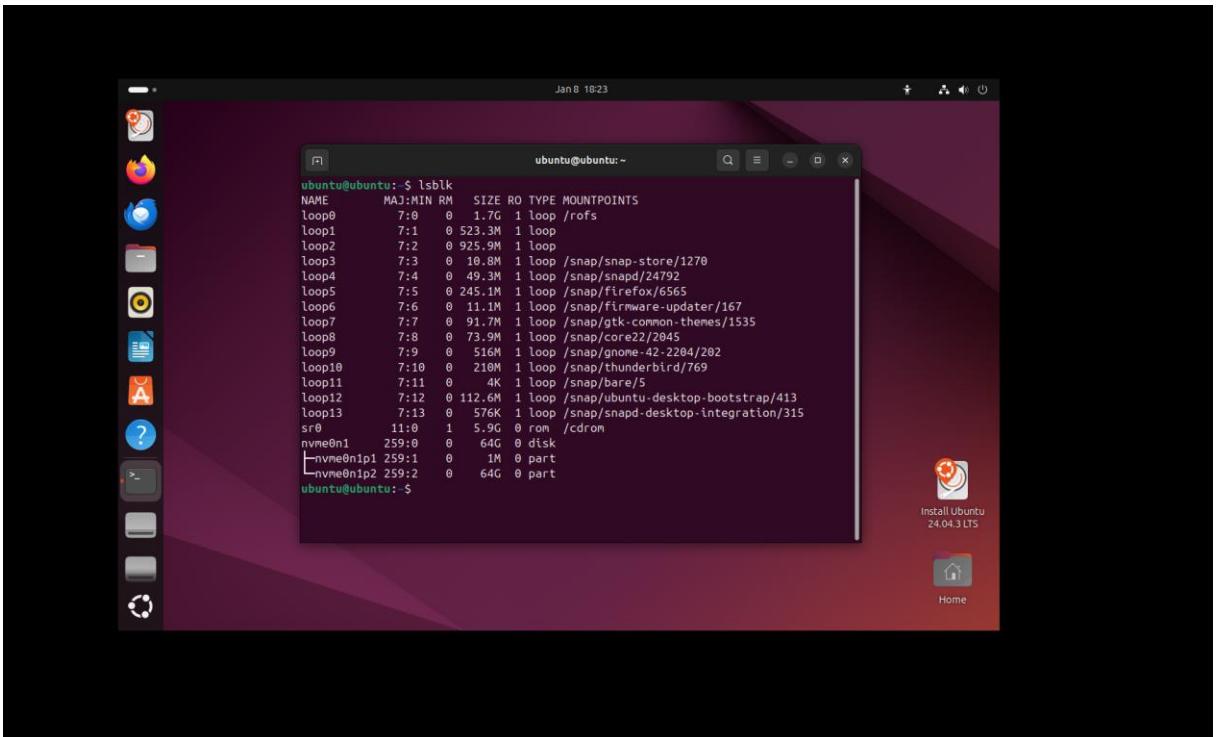


Fig. 4: Using lsblk on the original Ubuntu VM to identify the 64 GB system disk (/dev/nvme0n1) that will be imaged

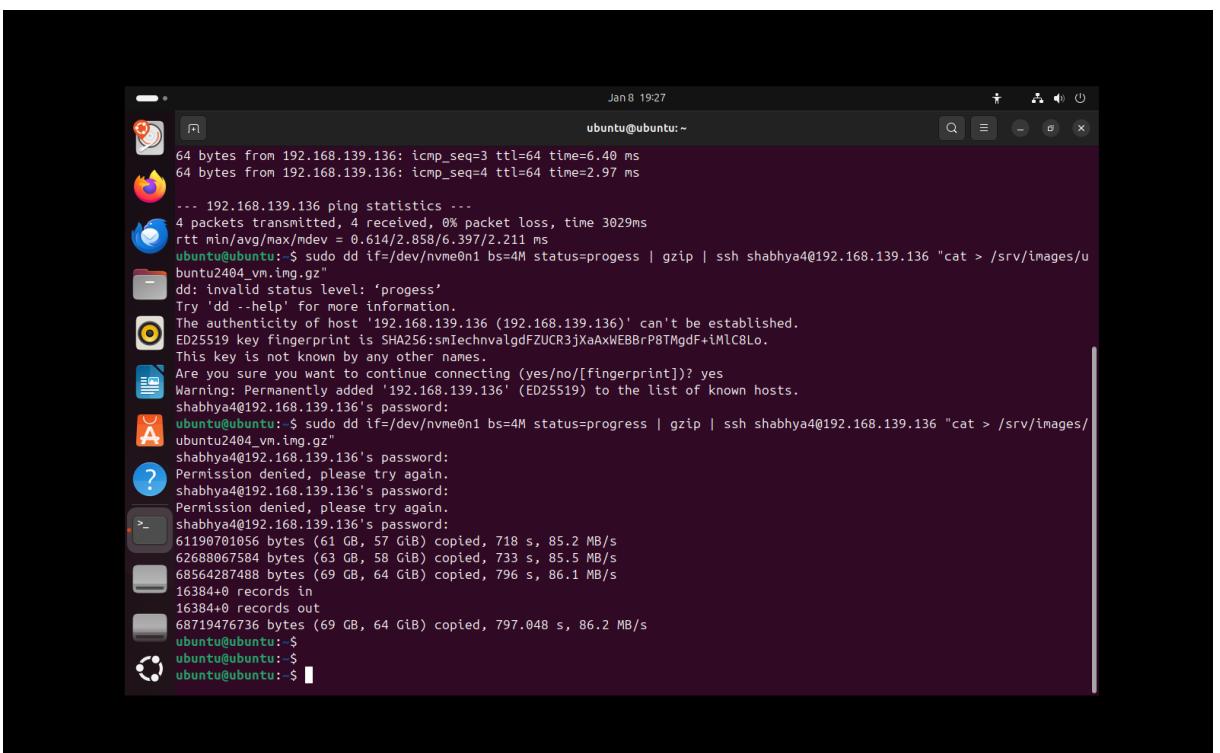
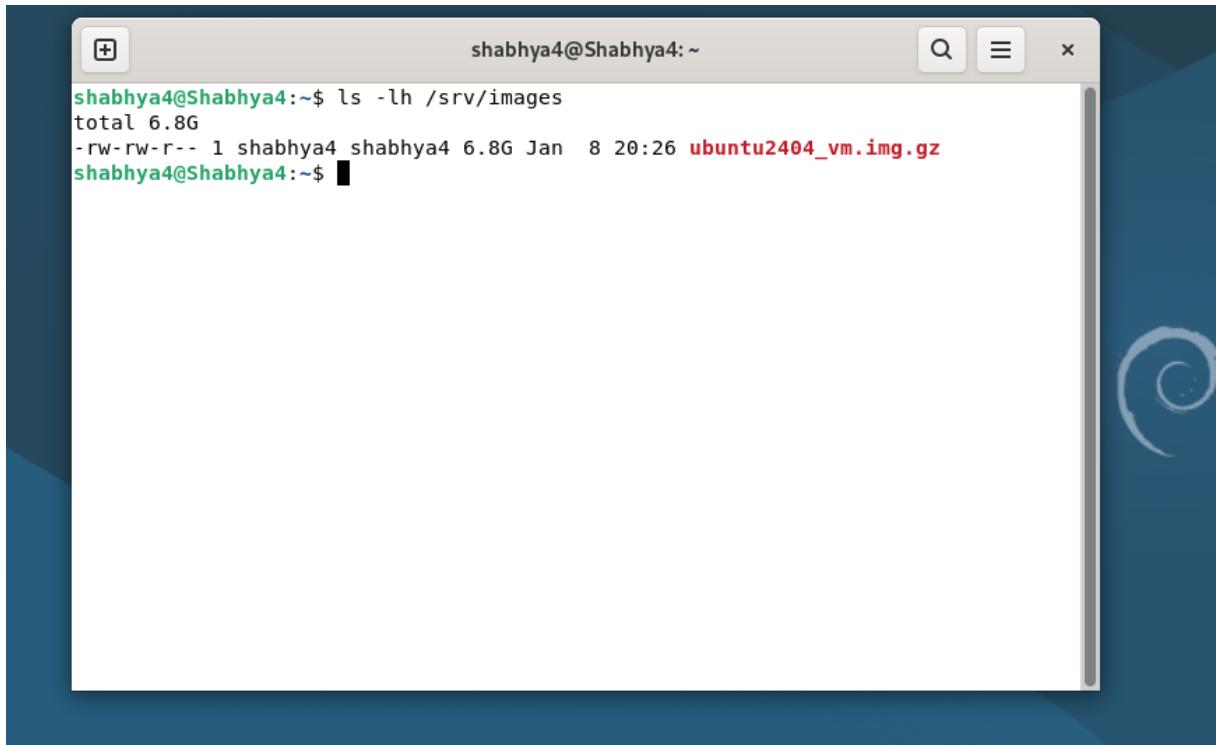


Fig. 5: Creating a compressed backup image of /dev/nvme0n1 and streaming it over SSH to the Debian image server (dd ... | gzip | ssh ... > /srv/images/ubuntu2404_vm.img.gz), showing that ~64 GB was copied successfully



A screenshot of a terminal window titled "shabhy4@Shabhy4: ~". The window shows the command "ls -lh /srv/images" being run, which lists a single file: "ubuntu2404_vm.img.gz" with a size of "6.8G". The terminal is set against a dark blue background with a faint "Debian" logo watermark.

```
shabhy4@Shabhy4:~$ ls -lh /srv/images
total 6.8G
-rw-rw-r-- 1 shabhy4 shabhy4 6.8G Jan  8 20:26 ubuntu2404_vm.img.gz
shabhy4@Shabhy4:~$
```

Fig. 6: Verifying on the Debian image server that the compressed backup image `ubuntu2404_vm.img.gz` (≈ 6.8 GB) is present in `/srv/images`

- Proof that you can restore the back-up image into an empty VM.

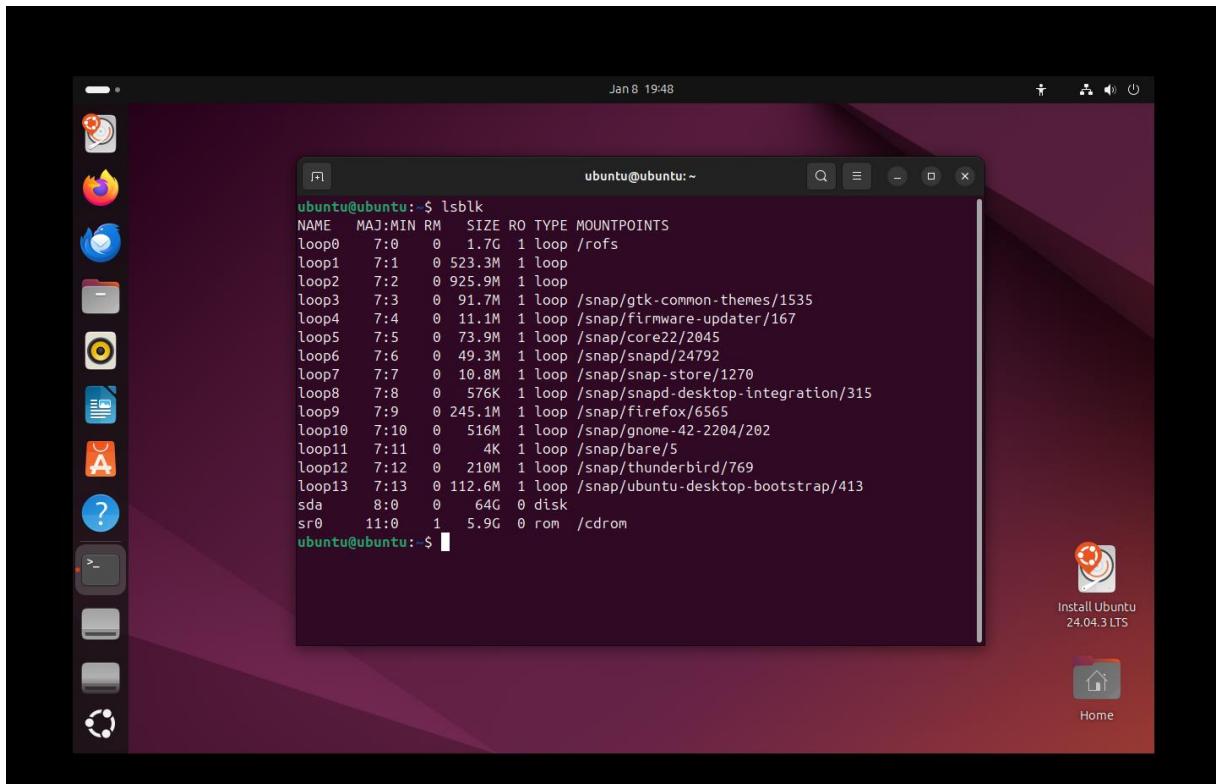


Fig. 7: Verifying the empty target VM in Ubuntu Live with `lsblk`; `/dev/sda` is the 64 GB virtual disk that will be overwritten by the restored backup image.

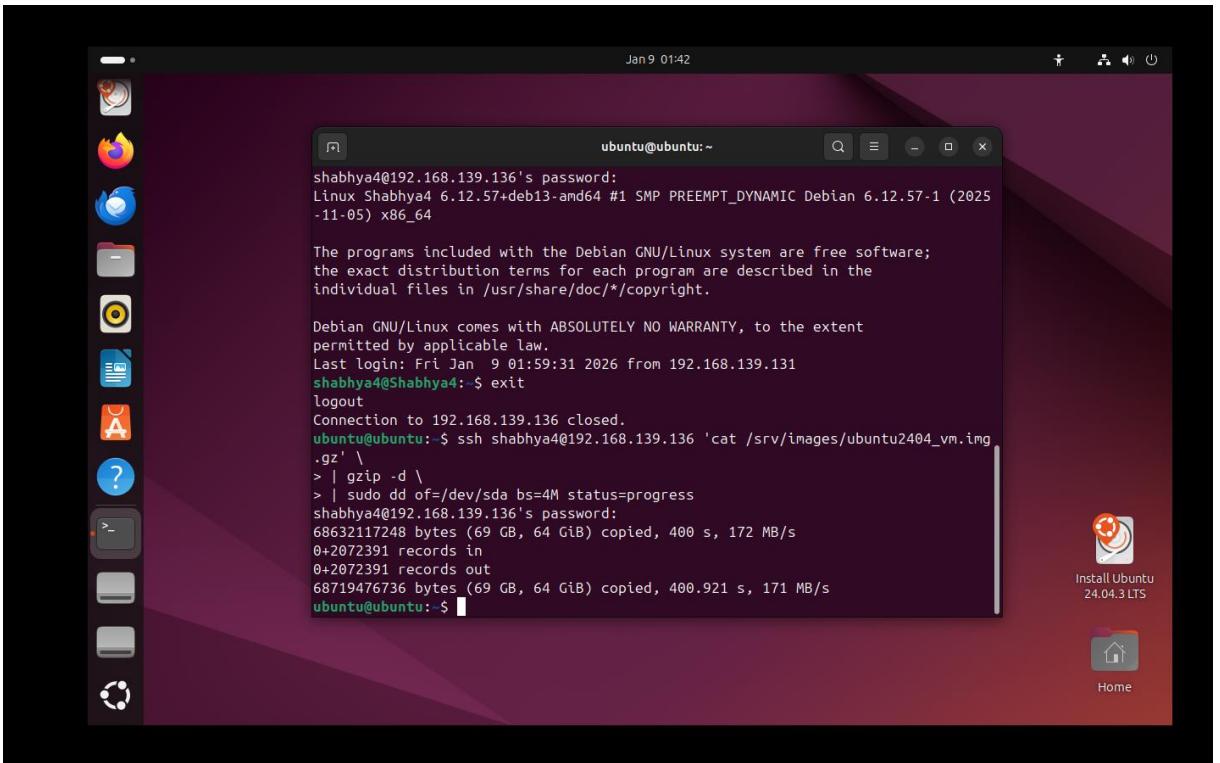


Fig. 8: Restoring the backup image from the Debian server to the empty Ubuntu VM using ssh '`cat /srv/images/ubuntu2404_vm.img.gz' | gzip -d | sudo dd of=/dev/sda ...;` dd reports ~69 GB copied, confirming the restore completed successfully.

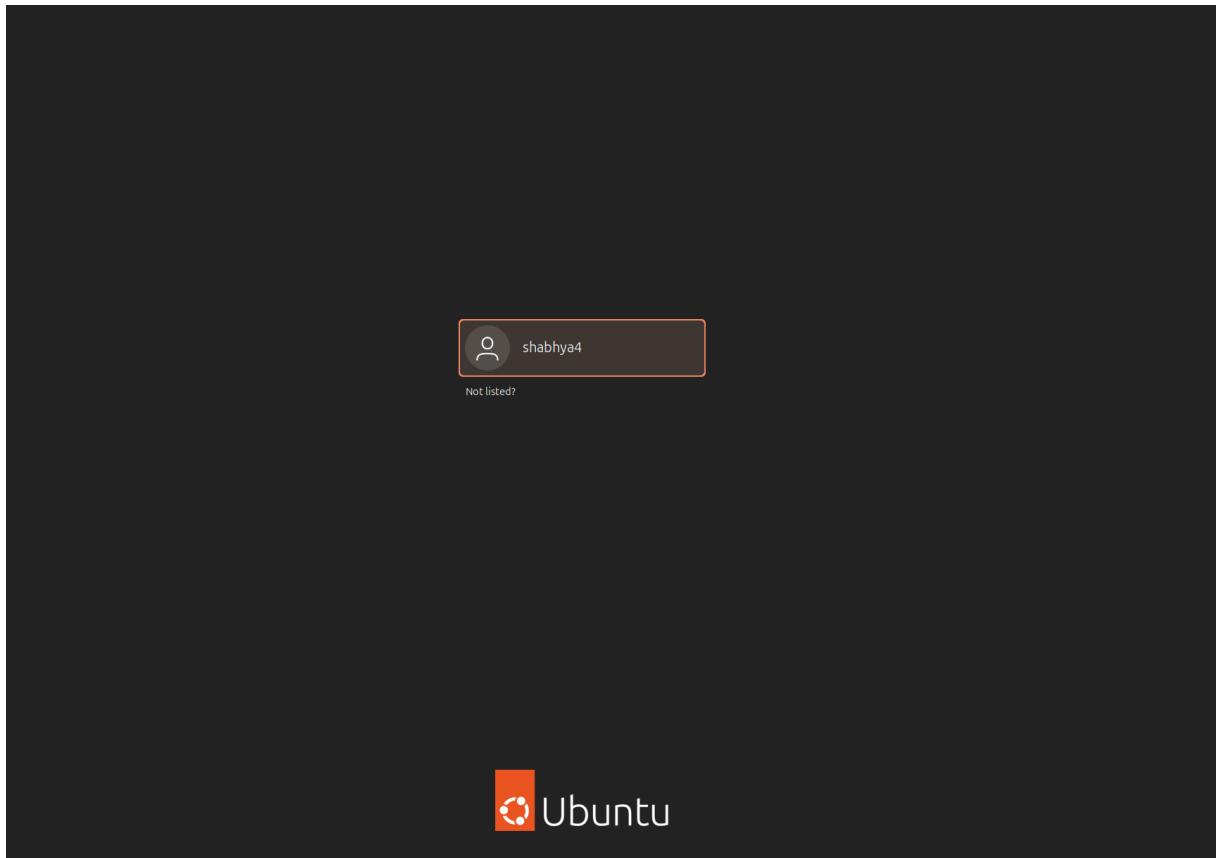


Fig. 9: The restored Ubuntu VM successfully booting to the login screen, proving that the disk image was restored correctly

Ready? Save this file and export it as a pdf file with the name: [week5.pdf](#)