



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Assignment 2:

Operating Systems

CertHE Computing Skills for Workplace

Computer Architecture and Operating Systems.

ACCA4022_LON

Kwame Assa Agyei (Lecturer)

University of Wales Trinity Saint David

(London Campus)

Student Name: Sinan Ilhan Kyazim

Student Number: 2513882

Group: 04

1.2 Memory Management

Memory management ensures that programs have enough RAM to run efficiently. I checked memory usage using system monitoring tools in both operating systems. From my observation, Windows provides a visual graph which is easier to understand, while Linux displays raw values that require more interpretation. However, Linux gives more precise control for advanced users [5].

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	8 K
System	4	Services	0	148 K
Secure System	172	Services	0	47,976 K
Registry	204	Services	0	40,500 K
smss.exe	792	Services	0	1,140 K
csrss.exe	1080	Services	0	6,208 K
wininit.exe	1176	Services	0	7,888 K
services.exe	1380	Services	0	12,148 K
LsaIso.exe	1344	Services	0	3,724 K
lsass.exe	1360	Services	0	33,408 K
svchost.exe	1084	Services	0	86,052 K
fontdrvhost.exe	1532	Services	0	3,764 K
WUDFHost.exe	1548	Services	0	26,476 K
svchost.exe	1648	Services	0	55,536 K
svchost.exe	1696	Services	0	13,068 K
svchost.exe	1820	Services	0	9,916 K
svchost.exe	1828	Services	0	5,224 K
svchost.exe	1840	Services	0	13,648 K
svchost.exe	1864	Services	0	10,580 K
svchost.exe	1912	Services	0	15,912 K
svchost.exe	1928	Services	0	17,316 K
svchost.exe	2004	Services	0	14,100 K
svchost.exe	896	Services	0	10,864 K
svchost.exe	1168	Services	0	10,196 K
svchost.exe	1164	Services	0	10,984 K
svchost.exe	2140	Services	0	9,820 K
IntelCpHDCPSvc.exe	2168	Services	0	5,240 K
svchost.exe	2196	Services	0	17,532 K
svchost.exe	2216	Services	0	23,940 K
svchost.exe	2224	Services	0	10,020 K
WUDFHost.exe	2288	Services	0	9,908 K
svchost.exe	2436	Services	0	7,828 K
svchost.exe	2488	Services	0	7,320 K
svchost.exe	2792	Services	0	11,196 K
svchost.exe	2836	Services	0	9,612 K
svchost.exe	2900	Services	0	25,948 K
svchost.exe	2936	Services	0	11,416 K
YMC.exe	3016	Services	0	9,560 K
svchost.exe	2620	Services	0	21,292 K
WUDFHost.exe	3116	Services	0	5,748 K
svchost.exe	3152	Services	0	15,336 K
svchost.exe	3160	Services	0	5,788 K
svchost.exe	3184	Services	0	8,032 K
svchost.exe	3244	Services	0	7,036 K

USER	PID	PPID	CPU	MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0	0.0	0.0	16640	11024	?	S	17:18	0:00	/sbin/init auto noprompt splash
root	2	0	0.0	0.0	0	0	?	S	17:18	0:00	[kthreadd]
root	3	0	0.0	0.0	0	0	?	S	17:18	0:00	[pool_worker_release]
root	4	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-crcu_0]
root	5	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-crcu_1]
root	6	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-crcu_2]
root	7	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-netns]
root	11	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/u256:0-ext4-rsv-conversion]
root	12	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-mq]
root	13	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-crcu_3]
root	14	0	0.0	0.0	0	0	?	S	17:18	0:00	[rcu_tasks_kthreadd]
root	15	0	0.0	0.0	0	0	?	S	17:18	0:00	[rcu_tasks_trace_kthreadd]
root	16	0	0.0	0.0	0	0	?	S	17:18	0:00	[ksoftirqd/0]
root	17	0	0.0	0.0	0	0	?	S	17:18	0:00	[rcu_gp]
root	18	0	0.0	0.0	0	0	?	S	17:18	0:00	[migration/0]
root	19	0	0.0	0.0	0	0	?	S	17:18	0:00	[idle_task/0]
root	20	0	0.0	0.0	0	0	?	S	17:18	0:00	[cpuhp/0]
root	21	0	0.0	0.0	0	0	?	S	17:18	0:00	[cpuhp/1]
root	22	0	0.0	0.0	0	0	?	S	17:18	0:00	[idle_inject/1]
root	23	0	0.0	0.0	0	0	?	S	17:18	0:00	[migration/1]
root	24	0	0.0	0.0	0	0	?	S	17:18	0:00	[ksoftirqd/1]
root	25	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/1:00-events_highpri]
root	26	0	0.0	0.0	0	0	?	S	17:18	0:00	[kdevtmpfs]
root	28	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-tnet_]
root	31	0	0.0	0.0	0	0	?	S	17:18	0:00	[kautd]
root	33	0	0.0	0.0	0	0	?	S	17:18	0:00	[kthundask]
root	34	0	0.0	0.0	0	0	?	S	17:18	0:00	[com_reader]
root	37	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-write]
root	38	0	0.0	0.0	0	0	?	S	17:18	0:00	[kcompactd0]
root	39	0	0.0	0.0	0	0	?	SN	17:18	0:00	[kswapd]
root	42	0	0.0	0.0	0	0	?	SN	17:18	0:00	[khupeaged]
root	43	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-kint]
root	44	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-kbldc]
root	45	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-bldc]
root	46	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-cpu]
root	47	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-tpm_d]
root	48	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-ata_x]
root	49	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-md]
root	50	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-md_b]
root	51	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-ndac]
root	52	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-devfr]
root	53	0	0.0	0.0	0	0	?	S	17:18	0:00	[kswapd0]
root	56	0	0.0	0.0	0	0	?	S	17:18	0:00	[kswapd0]
root	57	0	0.0	0.0	0	0	?	S	17:18	0:00	[cryptfs_kthreadd]
root	58	0	0.0	0.0	0	0	?	S	17:18	0:00	[kworker/R-kthreadd]
root	59	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/24-pciehp]
root	60	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/25-pciehp]
root	61	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/26-pciehp]
root	62	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/27-pciehp]
root	63	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/28-pciehp]
root	64	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/29-pciehp]
root	65	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/30-pciehp]
root	66	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/31-pciehp]
root	67	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/32-pciehp]
root	68	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/33-pciehp]
root	69	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/34-pciehp]
root	70	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/35-pciehp]
root	71	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/36-pciehp]
root	72	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/37-pciehp]
root	73	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/38-pciehp]
root	74	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/39-pciehp]
root	75	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/40-pciehp]
root	76	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/41-pciehp]
root	77	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/42-pciehp]
root	78	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/43-pciehp]
root	79	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/44-pciehp]
root	80	0	0.0	0.0	0	0	?	S	17:18	0:00	[irq/45-pciehp]

Figure 2. Memory Management

1.3 File System Management

File management allows users to organize and access data. Commands such as `dir` in Windows and `ls` in Linux provide a way to view directory contents [1]. Although both systems perform similar tasks, I found Linux commands slightly faster once I became familiar with them.

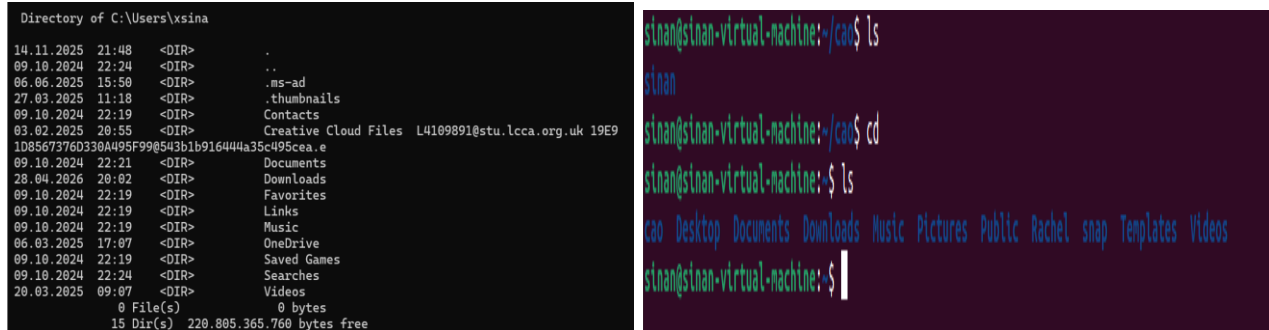


Figure 3. File System Management

1.4 Security and Protection

Security is another important function. I used commands like `net user` in Windows and `sudo adduser` in Linux to explore user account management [1]. Operating systems like macOS also implement similar security features.

2. Use of Windows OS CLI commands.

In this section, I tested several command-line tools using Command Prompt and PowerShell.

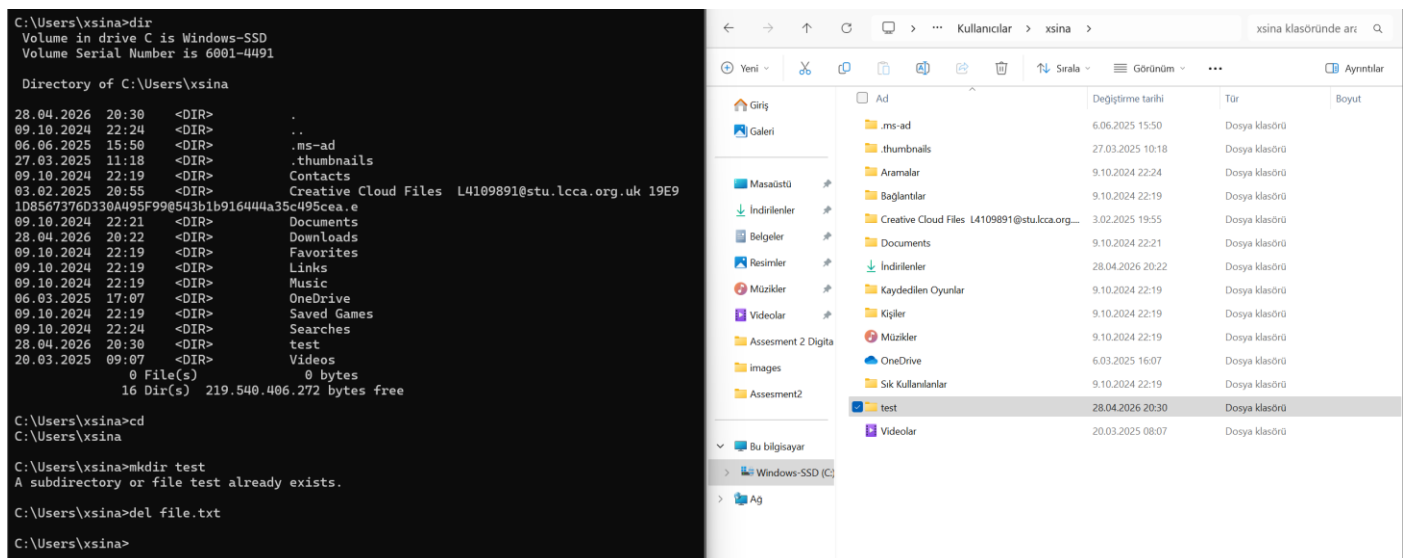


Figure 4. File and Directory Management

When I tried these commands in Command Prompt, I understood better how file and directory management works in practice. First, I used the “**dir**” command and it showed me all the folders and files inside my user directory. I noticed that it lists everything in an ordered way, including system folders and the ones I created before. This helped me see how the computer keeps everything structured. Then I used the “**cd**” command. Since I was already in the correct directory, nothing really changed, but it showed me how I can move between folders if needed. After that, I tried to create a new folder using “**mkdir test**”. However, I got an error message saying that the folder already exists. From this, I understood that the system does not allow creating two folders with the same name in the same location. Finally, I used the “**del file.txt**” command to delete a file. After running this command, the file was removed, which showed me how easy it is to manage and clean files using simple commands. Overall, by trying these commands myself, I learned that file and directory management is not just theory. The commands directly affect the files and folders on the computer. It made me realize how important it is to be careful when using commands like delete, because they can remove files permanently.

The image consists of two side-by-side screenshots of a Windows Command Prompt window. The left screenshot shows the output of the `ipconfig` command, displaying network configuration for various adapters. The right screenshot shows the output of the `ping google.com` and `netstat` commands, displaying network connectivity and active connections.

```
C:\Users\xsina>ipconfig

Windows IP Configuration

Wireless LAN adapter Yerel Ağ Bağlantısı* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Wireless LAN adapter Yerel Ağ Bağlantısı* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : 

Ethernet adapter VMware Network Adapter VMnet1:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::4c6:5ala:5d39:e017%38
    IPv4 Address. . . . . : 192.168.126.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Ethernet adapter VMware Network Adapter VMnet8:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::dae3:28db:85df:7831%39
    IPv4 Address. . . . . : 192.168.101.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . : home
    IPv6 Address. . . . . : 2a00:23c5:9bb9:501:5daa:c49d:e5dc:f970
    Temporary IPv6 Address. . . . . : 2a00:23c5:9bb9:501:313c:b221:3714:aabc
    Link-local IPv6 Address . . . . . : fe80::1eda:572e:3cbc:2a46%14
    IPv4 Address. . . . . : 192.168.1.98
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::e675:dcff:fec3:7baf%14
                                192.168.1.254

Ethernet adapter Bluetooth Ağ Bağlantısı:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :
```

```
C:\Users\xsina>ping google.com

Pinging google.com [2a00:1450:4009:c15::65] with 32 bytes of data:
Reply from 2a00:1450:4009:c15::65: time=7ms
Reply from 2a00:1450:4009:c15::65: time=8ms
Reply from 2a00:1450:4009:c15::65: time=6ms
Reply from 2a00:1450:4009:c15::65: time=5ms

Ping statistics for 2a00:1450:4009:c15::65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 5ms, Maximum = 8ms, Average = 6ms

C:\Users\xsina>netstat

Active Connections

Proto Local Address          Foreign Address         State
TCP    10.8.104.185:49804      72.153.5.136:https      TIME_WAIT
TCP    10.8.104.185:49807      173.46.83.203:http      TIME_WAIT
```

Figure 5. Network Monitoring

When I tried network commands in Command Prompt, I started with **ipconfig**. This command showed me my computer’s network information, such as the IP address, subnet mask, and default gateway. From this, I understood that my device has its own IP address when it connects to a

network, and this is how it is identified on the internet or local network. It helped me see the basic structure of my network connection.

After that, I used the **ping google.com** command. It sent data packets to Google and received replies back. I saw response times like 5ms, 6ms, 7ms, and 8ms, and there was 0% packet loss. This showed me that my internet connection was working properly and the connection was stable. I also learned that ping is useful to test if a website or server is reachable and to check the speed of the connection.

Then I used the **netstat** command. It showed active connections on my computer, including local addresses, foreign addresses, and states like "TIME_WAIT." This made me realize that my computer is always communicating with different servers in the background, even when I am not actively doing anything online. It was interesting to see how many connections are happening at the same time.

Overall, by trying these commands myself, I understood better how my computer connects to the internet and how data is being sent and received. Each command showed a different part of the network, and together they gave me a clearer idea of how networking works in real life.

```
C:\Users\xsina>net user

\\DESKTOP-GV69TMK Kullanıcı Hesapları

-----
Administrator      Guest      VarsayılanHesap
WDAGUtilityAccount  xsina
Komut başarıyla tamamlandı.

C:\Users\xsina>schtasks

Folder: \
TaskName
=====
Adobe Acrobat Update Task      29.04.2026 13:00:00      Ready
Launch Adobe CCXProcess        29.04.2026 15:04:00      Ready
OneDrive Per-Machine Standalone Update T 29.04.2026 20:32:42      Ready
OneDrive Reporting Task-S-1-5-21-2827236 29.04.2026 18:31:32      Ready
OneDrive Startup Task-S-1-5-21-282723661 N/A                      Ready

Folder: \GoogleSystem
TaskName
=====
INFO: There are no scheduled tasks presently available at your access level.

Folder: \Lenovo
TaskName
=====
LenovoNowLauncher              N/A                      Ready
LenovoNowQuarterlyLaunch       27.07.2026 14:38:00      Ready
LenovoNowTask                  N/A                      Ready

Folder: \Lenovo\ImController
TaskName
=====
INFO: There are no scheduled tasks presently available at your access level.

Folder: \Lenovo\UDC
TaskName
=====
Lenovo UDC Diagnostic Scan      N/A                      Disabled
Lenovo UDC Idle Monitor         N/A                      Ready

Folder: \Lenovo\Vantage
TaskName
=====
INFO: There are no scheduled tasks presently available at your access level.
```

Figure 6. User Management & Task Scheduling

When I used the **net user** command, it showed me all the user accounts on my computer, such as Administrator, Guest, and my own account. From this, I understood that a Windows system can have multiple user profiles, and each one has different access levels. Then I used the **schtasks**

command to see scheduled tasks. It listed different background tasks like Adobe update tasks, OneDrive tasks, and Lenovo system tasks. It also showed their next run times and status (like Ready or Disabled). I realized that many programs automatically run in the background for updates or system maintenance, even when I am not using them. Overall, these commands helped me see that my computer is not just running visible programs, but also many hidden processes and scheduled tasks in the background.

3. Use of Linux OS CLI commands.

In this section, I used a Linux terminal to test Bash commands [3].

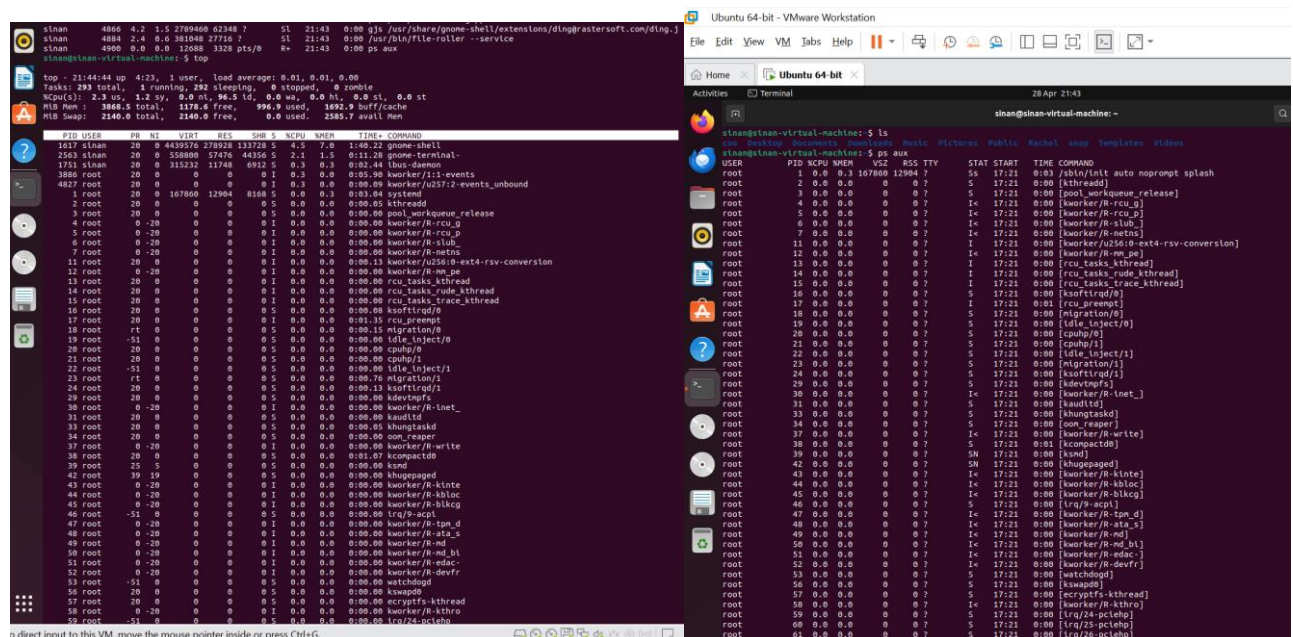


Figure 7. Process Management

When I used Linux commands like **ps aux** and **top**, I learned how to see running processes on the system. With **ps aux**, I could see a full list of all running processes, including system and user processes. It showed details like process ID (PID), CPU usage, memory usage, and which command started the process. From this, I understood that Linux is always running many background processes, even when I am not directly using them. Then I used the **top** command, which gave me a real-time view of system performance. It showed which processes are currently using the most CPU and memory, and the list kept updating automatically. This helped me see which programs are affecting system performance at that moment. Overall, these commands helped me understand how Linux manages processes and how I can monitor system activity in real time.

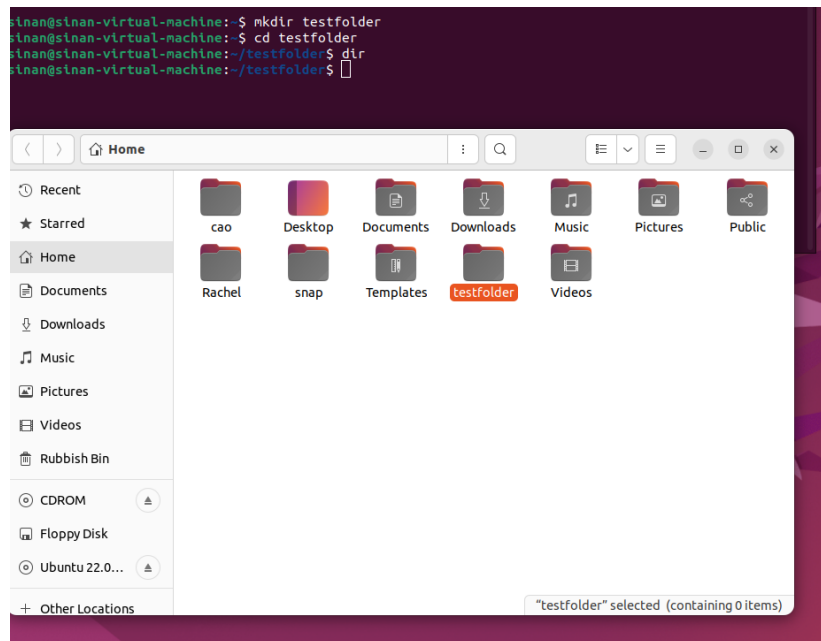


Figure 8. File

I created a folder using the **mkdir testfolder** command and then entered it with **cd testfolder**. After that, I used the **dir** command to see what was inside the folder. At first, I expected to see something, but nothing was listed because the folder was empty. Then I realized that when you create a new folder, it doesn't contain anything by default. This helped me understand how basic terminal commands work and how to navigate between directories. I also noticed that even though **dir** worked, in Linux people usually use the **ls** command for listing files. Overall, it was a simple exercise but useful for getting used to the terminal.

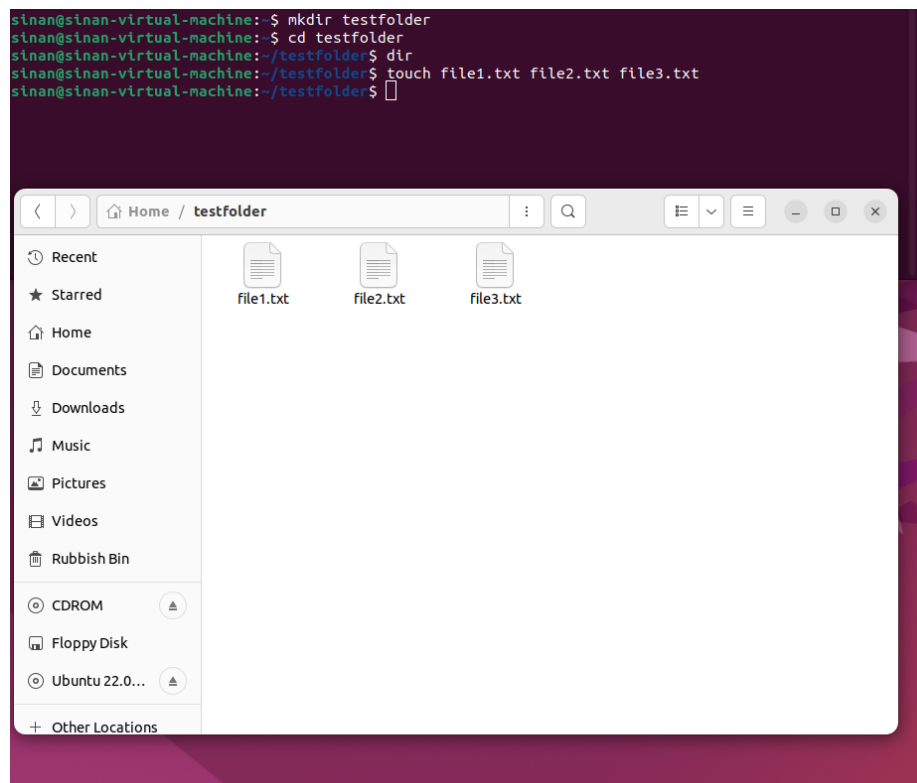
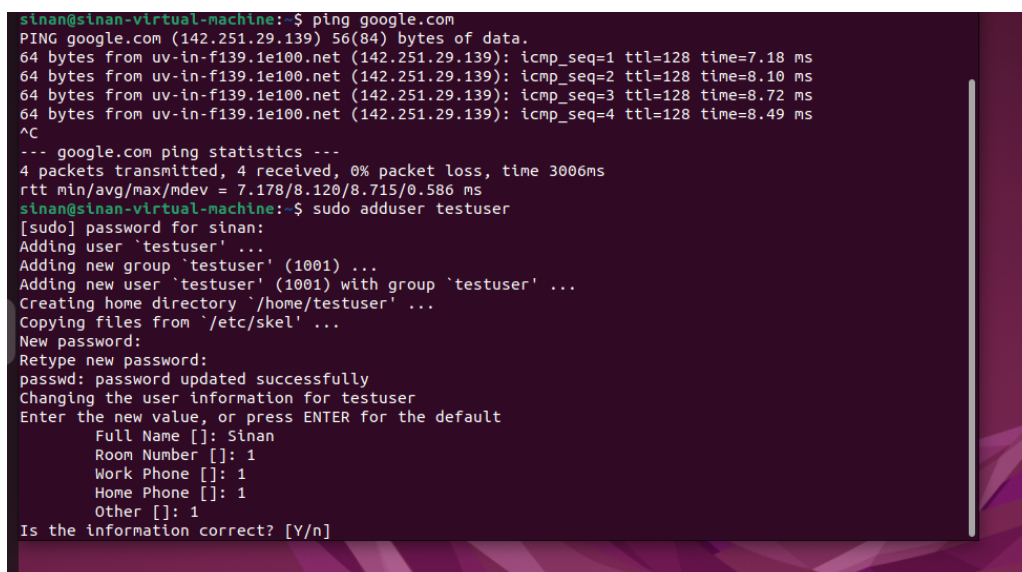


Figure 9. Multiple Files

After creating and entering the folder, I used the touch file1.txt file2.txt file3.txt command to create multiple files at the same time. Before that, when I checked the folder with dir, it was empty. After running the touch command, I didn't see any output in the terminal, but when I looked at the folder, I could see that three files were created instantly. This showed me that touch is not only used to create a single file, but it can also create multiple files in one command, which saves time. It also helped me understand that some terminal commands don't give visible feedback, but they still work in the background. Overall, it made file creation faster and more practical compared to creating files one by one.

A terminal window with a dark purple background and white text. The text shows a series of commands and their outputs. First, a 'ping google.com' command is executed, showing four successful responses with varying times. Then, the 'sudo adduser testuser' command is run. This prompts for a password, then displays the process of adding the user, including creating a group, adding the user to the group, creating a home directory, and copying files. It then prompts for a new password and its retype. Finally, it asks for additional user information like full name, room number, and phone numbers, all of which are entered as '1'. The process concludes by asking if the information is correct, with a '[Y/n]' prompt.

```
sinan@sinan-virtual-machine:~$ ping google.com
PING google.com (142.251.29.139) 56(84) bytes of data:
64 bytes from uv-in-f139.1e100.net (142.251.29.139): icmp_seq=1 ttl=128 time=7.18 ms
64 bytes from uv-in-f139.1e100.net (142.251.29.139): icmp_seq=2 ttl=128 time=8.10 ms
64 bytes from uv-in-f139.1e100.net (142.251.29.139): icmp_seq=3 ttl=128 time=8.72 ms
64 bytes from uv-in-f139.1e100.net (142.251.29.139): icmp_seq=4 ttl=128 time=8.49 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 7.178/8.120/8.715/0.586 ms
sinan@sinan-virtual-machine:~$ sudo adduser testuser
[sudo] password for sinan:
Adding user `testuser' ...
Adding new group `testuser' (1001) ...
Adding new user `testuser' (1001) with group `testuser' ...
Creating home directory `/home/testuser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for testuser
Enter the new value, or press ENTER for the default
  Full Name []: Sinan
    Room Number []: 1
    Work Phone []: 1
    Home Phone []: 1
      Other []: 1
Is the information correct? [Y/n]
```

Figure 10. Network & User

In this screenshot, I first ran the ping google.com command to check my network connection. The command successfully sent 4 packets and received 4 responses, showing that my internet connection is working with no packet loss and low latency. After that, I used the sudo adduser testuser command to create a new user named "testuser." When I ran the command, the system asked for my password (since I used sudo) to proceed with administrative tasks. Then, I was prompted to set a password for the new user and enter some additional information like the user's full name, room number, and phone numbers. The system displayed the information I entered and asked if it was correct. If I had made any mistakes, I could have corrected them or just pressed Enter to keep the default values. This experience showed me how to manage user accounts in Linux using commands like adduser, how to configure user details, and the process of verifying information before finalizing it.

Conclusion

In this report, I explored the core functions of operating systems and tested both Windows and Linux command-line tools. Through practical experimentation, I gained a better understanding of how systems manage processes, memory, files, and users [5].

References

- [1] Microsoft, "Windows Command-Line Documentation," 2025.
- [2] Microsoft, "PowerShell Documentation," 2025.
- [3] Linux Foundation, "Introduction to Linux," 2024.
- [4] Ubuntu Documentation, "Basic Linux Commands," 2024.
- [5] A. Silberschatz, Operating System Concepts, Wiley, 2018.