COMPUTER ARCHITECTURE AND OPERATING SYSTEMS (ACCA4022)

Marvin Alexander Valencia Alfaro (2501143)
UNIVERSITY OF WALES TYNITY ST' DAVIS |

Table of Contents

1. INTRODUCTION 3
2. KEY FUNCTIONS OF OPERATING SYSTEMS 4
2.1 Process and Memory Management 4
2.2 Device and File Management 5
2.3 File Management and Storage 6
2.4 User management and security 7
Practical Analysis 7
3. USE OF WINDOWS OS CLI COMMANDS 8
3.1 Network Testing and Monitoring 9
3.2 Process and User Management 10
4. USE OF LINUX OS CLI COMMANDS 11
4.1 Process and System Monitoring 12
4.2 Network and user identification 13
5. Use of Linux OS Scripts 14
6. SCRIPTS AND AUTOMATION 15
Conclusion 17
APPENDICES 19
ONLINE CERTIFIED COURSE 20

1. INTRODUCTION

An Operating System (OS) is far more than just a graphical interface that displays folders and icons; it is the most critical software layer of any computer system. It acts as an intermediary between the physical hardware, such as the processor and memory, and the application software used by the user. Through this interaction, the operating system translates user requests into machine instructions that the hardware can execute efficiently [1].

As technology has evolved, operating systems have transformed from simple task executors into highly complex resource managers. Their primary objective is to guarantee efficiency by controlling the distribution of hardware resources among multiple applications while also maintaining security and system stability. According to Stallings, an operating system must support multitasking, memory allocation, process scheduling, and secure data access without compromising performance [2].

Today, operating systems are essential not only for personal computers but also in mobile devices, cloud computing environments, and enterprise servers. Systems such as Windows and Linux dominate professional environments because they provide strong networking capabilities, process management, and scripting tools that simplify administration tasks.

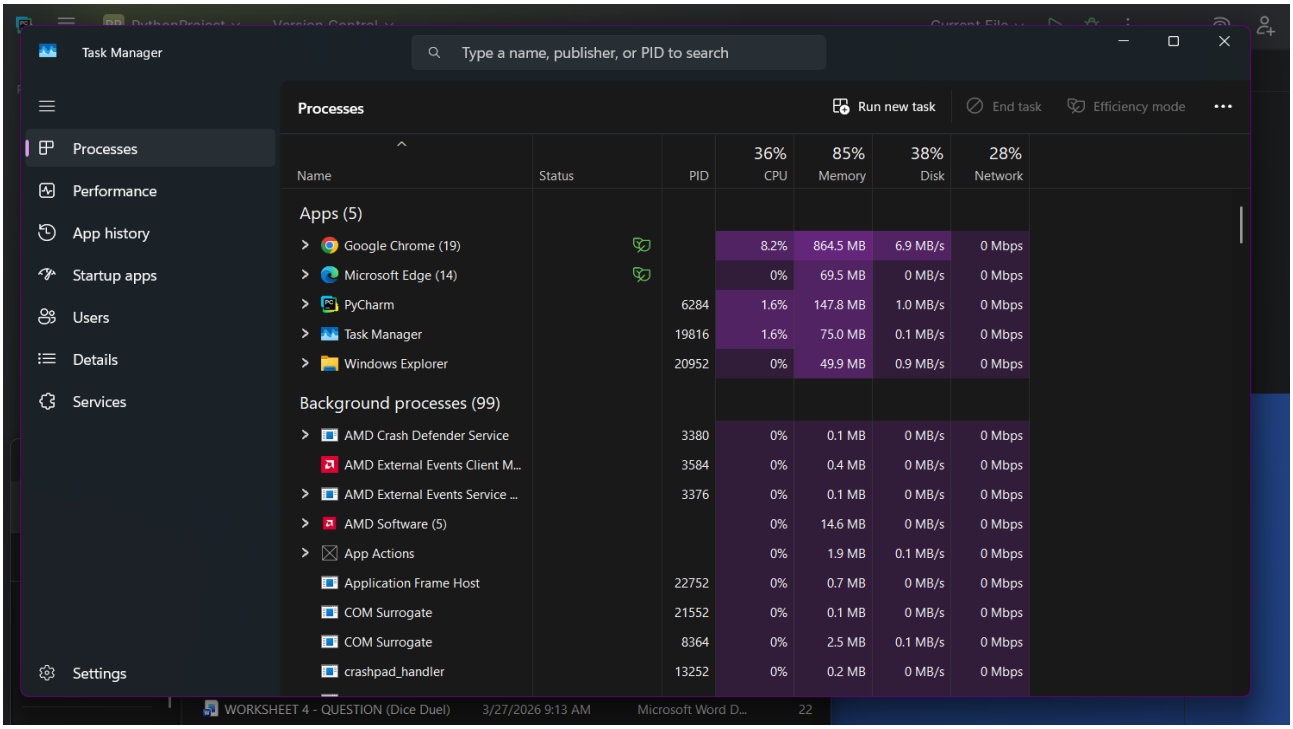
In this report, I will explore the practical management of these resources in both Windows and Linux environments. I will also demonstrate how command-line tools and scripting techniques can simplify administrative workflows and improve productivity.

2. KEY FUNCTIONS OF OPERATING SYSTEMS

2.1 Process and Memory Management

Process management is arguably one of the most important responsibilities of an operating system. The system must decide which program receives CPU attention at each moment in order to maintain smooth performance and avoid interruptions. In practice, this means that the operating system constantly balances active processes, background tasks, and user requests so the computer can work efficiently without crashing.[3]

Practical Analysis: While observing Task Manager in Figure 1, I noticed how the operating system keeps many background processes active at the same time. This shows how process management works in a real environment, since the operating system must create, schedule, and terminate processes depending on system demand. Task Manager also makes it easier to monitor CPU usage, memory consumption, and application behavior in a simple visual way.



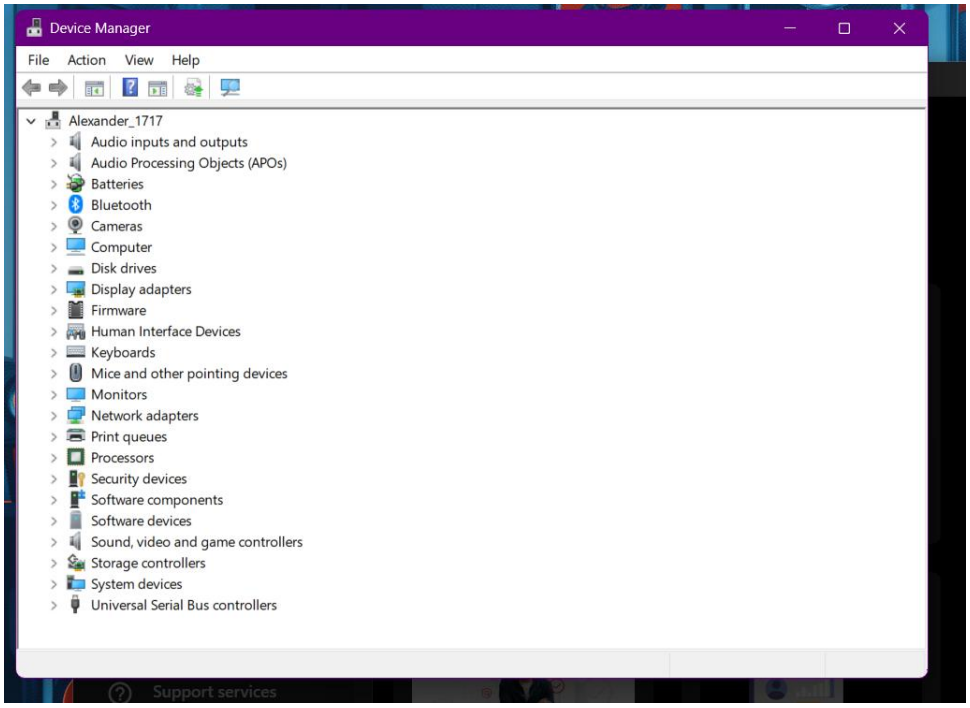
"Figure 1: As seen in Figure 1, the Windows Task Manager allows users to monitor active processes and their real-time CPU consumption, which is a core management task of the operating system.

2.2 Device and File Management

Memory Control: The operating system also manages RAM carefully to ensure that no single application exhausts the system's total capacity. Memory allocation is essential because several programs often run simultaneously, and each one requires space to function correctly. If memory is not distributed well, the whole system can become slow or unstable.

Practical Analysis: In Figure 3, the properties of the C: drive reveal the **NTFS** file system. It is fascinating to consider how the OS maintains a strict hierarchical structure, knowing exactly which physical sector of the disk holds every piece of data.

Security: This management also includes access permissions, ensuring that only authorized users can modify sensitive data.



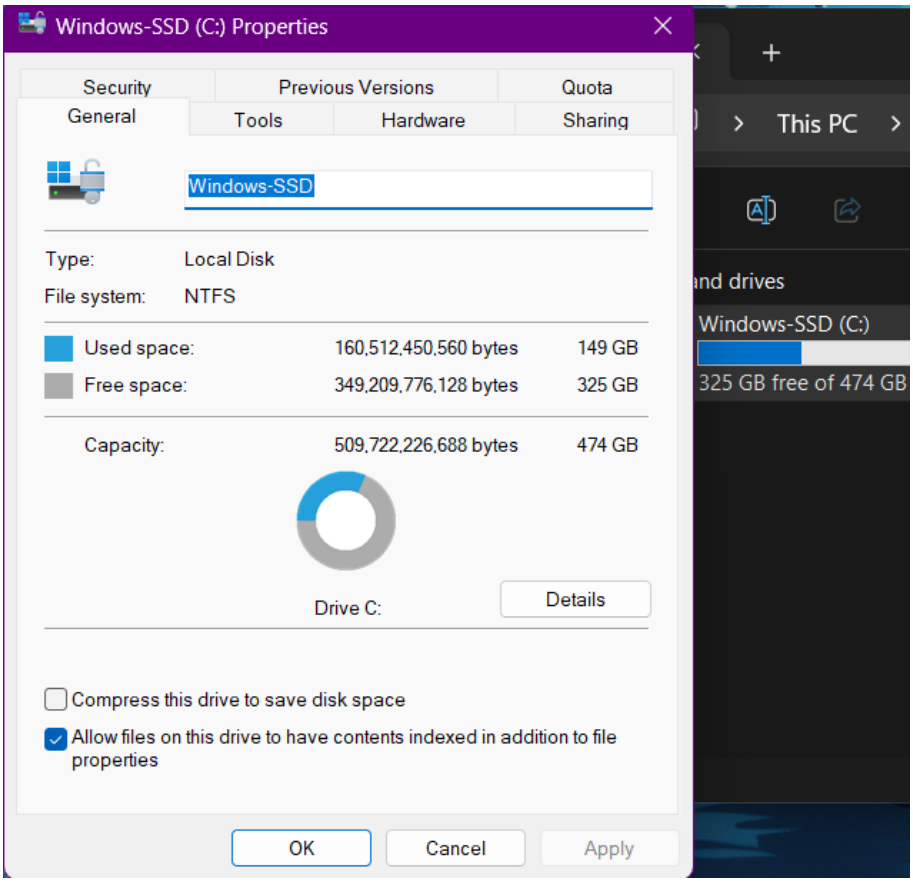
"Figure 2: illustrates the physical memory (RAM) allocation, showing how the operating system prevents applications from exhausting available resources,

2.3 File Management and Storage

The operating system also functions as both a digital librarian and a hardware translator. Through the Device Manager, it uses drivers to communicate with peripherals such as keyboards, monitors, and storage devices. This abstraction allows users to interact with hardware without needing to understand the technical details behind each device,

Practical Analysis: In Figure 3, the properties of the C: drive reveal the NTFS file system. This is important because the operating system uses a hierarchical structure to store and retrieve data efficiently. File systems such as NTFS organize information in a way that allows the user to locate files quickly while also protecting data integrity,

Security: This management also includes access permissions, ensuring that only authorized users can modify sensitive data. In modern operating systems, file permissions are a basic but essential security measure because they help protect the system from accidental or unauthorized changes.



"Figure 3: displays the file system type (NTFS) and the distribution of used versus free space on the primary storage drive."

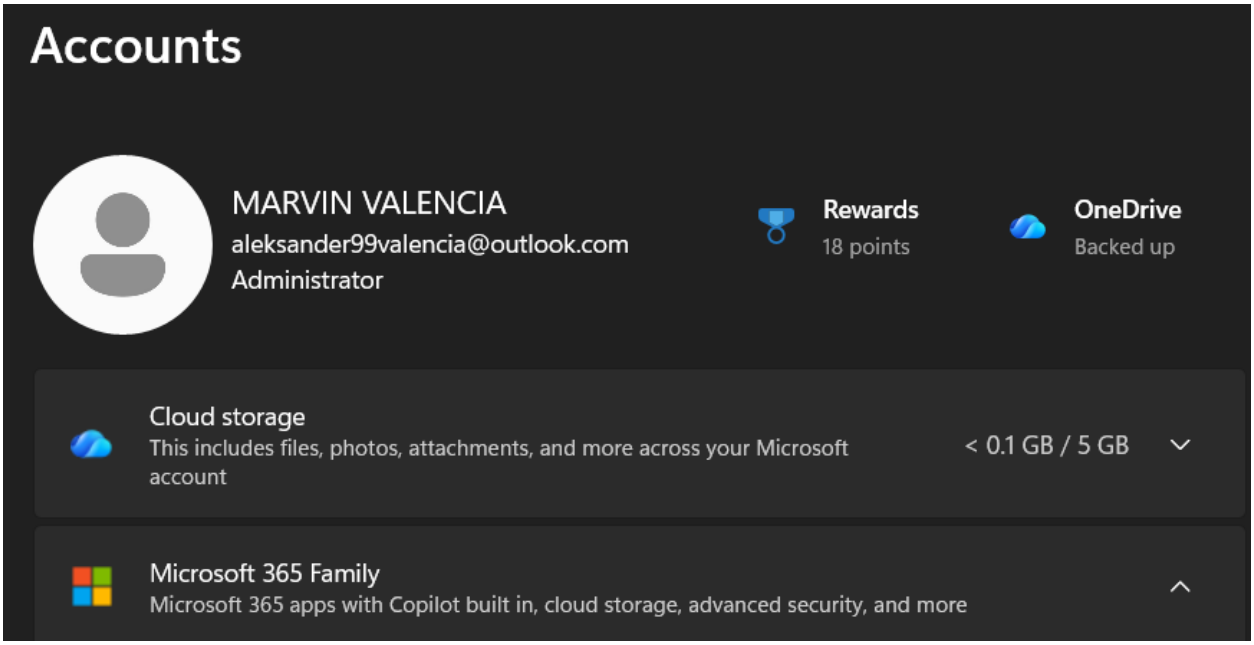
2.4 User management and security

Security is a critical responsibility of modern operating systems. The OS controls user authentication, access permissions, and account privileges in order to protect data and prevent unauthorized access.

Operating systems use usernames, passwords, encryption, and permission policies to ensure that users can only access the resources they are authorized to use [1]. In professional environments, these controls are fundamental for protecting organizational information.

Practical Analysis

By analyzing user account settings and administrative permissions, it becomes clear how the operating system separates standard users from administrators. This separation reduces the risk of accidental system modifications or malware infections.



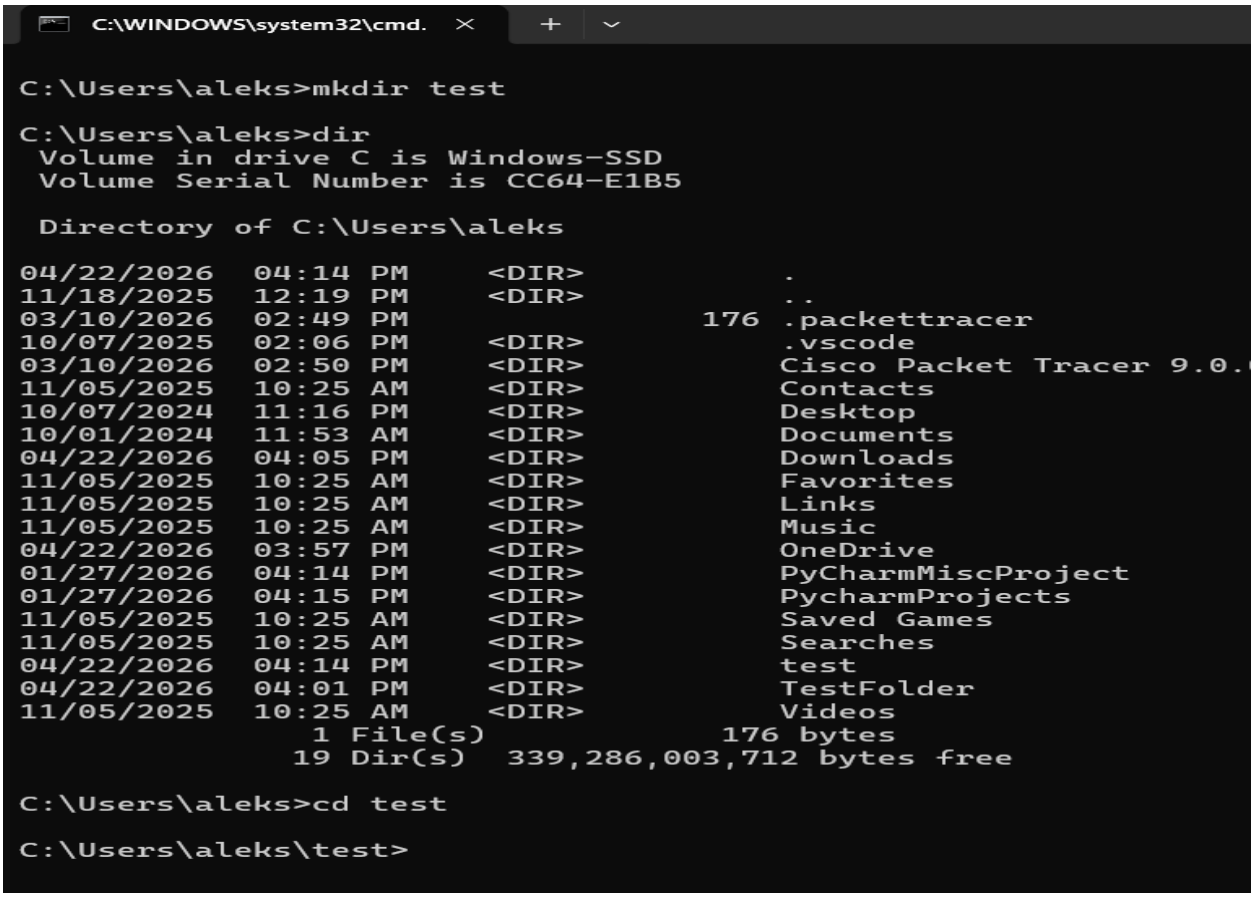
"Figure 4: User account management and security settings.*"*

3. USE OF WINDOWS OS CLI COMMANDS

For this section, I used Command Prompt (CMD) and PowerShell to interact directly with the operating system. Compared to the graphical interface, command-line tools provide faster and more precise system administration [4].

File Management: I used the mkdir command to create a test directory and dir to list the contents (Figure 5). As Microsoft documentation suggests, the CLI allows for faster data manipulation than the graphical interface.

Networking: Commands like ipconfig and ping proved vital for network monitoring. They allowed me to identify my local IP address and test connectivity to external servers like Google.



```
C:\WINDOWS\system32\cmd.

C:\Users\aleks>mkdir test

C:\Users\aleks>dir
 Volume in drive C is Windows-SSD
 Volume Serial Number is CC64-E1B5

 Directory of C:\Users\aleks

04/22/2026  04:14 PM    <DIR>          .
11/18/2025  12:19 PM    <DIR>          ..
03/10/2026  02:49 PM               176 .packettracer
10/07/2025  02:06 PM    <DIR>          .vscode
03/10/2026  02:50 PM    <DIR>          Cisco Packet Tracer 9.0.
11/05/2025  10:25 AM    <DIR>          Contacts
10/07/2024  11:16 PM    <DIR>          Desktop
10/01/2024  11:53 AM    <DIR>          Documents
04/22/2026  04:05 PM    <DIR>          Downloads
11/05/2025  10:25 AM    <DIR>          Favorites
11/05/2025  10:25 AM    <DIR>          Links
11/05/2025  10:25 AM    <DIR>          Music
04/22/2026  03:57 PM    <DIR>          OneDrive
01/27/2026  04:14 PM    <DIR>          PyCharmMiscProject
01/27/2026  04:15 PM    <DIR>          PycharmProjects
11/05/2025  10:25 AM    <DIR>          Saved Games
11/05/2025  10:25 AM    <DIR>          Searches
04/22/2026  04:14 PM    <DIR>          test
04/22/2026  04:01 PM    <DIR>          TestFolder
11/05/2025  10:25 AM    <DIR>          Videos
               1 File(s)            176 bytes
              19 Dir(s)  339,286,003,712 bytes free

C:\Users\aleks>cd test

C:\Users\aleks\test>
```

Figure 5: *Creating and navigating directories using CMD.*

In Figure 5, the commands mkdir were used to create a folder, dir to list the items, and cd to navigate. These tools allow for faster data manipulation than the graphical interface [4].

3.1 Network Testing and Monitoring

```
C:\WINDOWS\system32\cmd.

Microsoft Windows [Version 10.0.26100.7623]
(c) Microsoft Corporation. All rights reserved.

C:\Users\aleks> ipconfig

Windows IP Configuration


Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 2:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : home.arpa
   IPv6 Address. . . . . . . . . . . : 2a0e:1d47:cf1a:c000:9e6:743e:a41e:3243
   IPv6 Address. . . . . . . . . . . : 2a0e:1d47:cf1a:c000:c266:af8d:a3ed:5bae
   Temporary IPv6 Address. . . . . . : 2a0e:1d47:cf1a:c000:4837:d5cc:4281:a2ff
   Link-local IPv6 Address . . . . . : fe80::d57c:f63d:a881:7e35%5
   IPv4 Address. . . . . . . . . . . : 192.168.1.27
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::eefc:2fff:fe41:4557%5
                                       192.168.1.1

C:\Users\aleks>ping google.com

Pinging google.com [2a00:1450:4009:c0f::66] with 32 bytes of data:
Reply from 2a00:1450:4009:c0f::66: time=13ms
Reply from 2a00:1450:4009:c0f::66: time=13ms
Reply from 2a00:1450:4009:c0f::66: time=17ms
Reply from 2a00:1450:4009:c0f::66: time=25ms

Ping statistics for 2a00:1450:4009:c0f::66:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 13ms, Maximum = 25ms, Average = 17ms

C:\Users\aleks>
```

Figure 6: *Checking connectivity and network parameters.*

The `ipconfig` command was run to obtain the local IP address, and `ping google.com` was used to measure connection latency. These are critical functions for diagnosing communication problems in the operating system.

3.2 Process and User Management

```
C:\WINDOWS\system32\cmd.

Microsoft Windows [Version 10.0.26100.7623]
(c) Microsoft Corporation. All rights reserved.

C:\Users\aleks>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0      4,408 K
Secure System                  140 Services                   0     43,408 K
Registry                       184 Services                   0     31,552 K
smss.exe                       656 Services                   0          4 K
csrss.exe                     1172 Services                   0      2,768 K
wininit.exe                   1336 Services                   0        456 K
services.exe                  1468 Services                   0      6,620 K
LsaIso.exe                    1500 Services                   0      1,708 K
lsass.exe                     1520 Services                   0     14,024 K
svchost.exe                   1656 Services                   0     41,200 K
fontdrvhost.exe               1676 Services                   0         24 K
WUDFHost.exe                  1776 Services                   0        200 K
svchost.exe                   1800 Services                   0     16,616 K
```

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

User accounts for \\ALEXANDER_1717

-------------------------------------------------------------------------------
Administrator            aleks                    DefaultAccount
Guest                    WDAGUtilityAccount       WsiAccount
The command completed successfully.


C:\Users\aleks>
```

Figure 7: List of active processes and local user accounts.

The operating system uses tasklist to display all running programs, allowing the administrator to monitor resource consumption. The `net user` command was used to verify user accounts with access to the system.

4. USE OF LINUX OS CLI COMMANDS

In Linux, via **Google Cloud Shell**, I explored a **Bash**-based environment. Linux is renowned for its power in server administration, and using it helped me understand why it is preferred in professional environments.[5]

Key Commands: I utilized ls -la to view hidden files and top to monitor active processes in real-time (Figure 8). The top command is particularly impressive as it shows the OS managing multitasking dynamically to ensure stability.

User Identification: The whoami command confirmed the active session, reflecting the OS's role in security and user management.

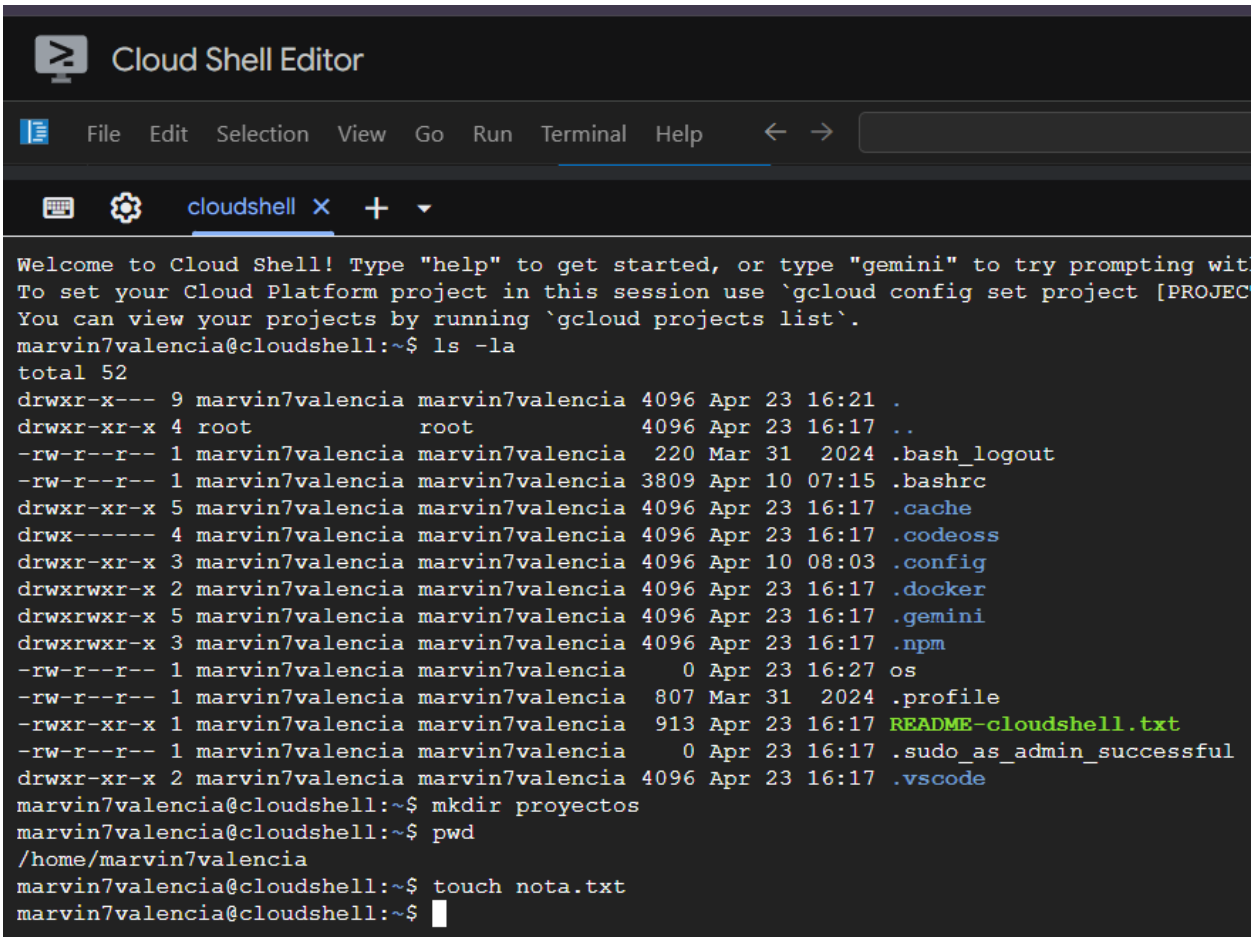


Figure 8: *Directory management in a Linux environment.*

As shown in **Figure 8**, the command mkdir was used to create a directory, and ls -l was employed to verify permissions and ownership. These commands are fundamental for the OS to maintain a structured file system.

4.1 Process and System Monitoring

```
File  Edit  Selection  View  Go  Run  Terminal  Help

cloudshell ×

top - 16:37:15 up 27 min,  0 user,  load average: 0.00, 0.04, 0.06
Tasks:  32 total,   1 running,  31 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.8 us,  1.3 sy,  0.0 ni, 96.7 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   7947.4 total,   4526.8 free,   2234.2 used,   1431.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   5713.2 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    462 root      20   0 1793200  45712  31964 S   0.3   0.6   0:01.88 containerd
    662 marvin7+  10 -10    7148   2680   1992 S   0.3   0.0   0:00.04 tmux
      1 root      20   0    4324   3372   3092 S   0.0   0.0   0:00.06 bash
      9 syslog    20   0  222140   3712   2372 S   0.0   0.0   0:00.02 rsyslogd
     25 root      20   0   38768  28020  10908 S   0.0   0.3   0:00.73 python
     26 root      20   0    6196   2640   2400 S   0.0   0.0   0:00.00 logger
     68 root      10 -10   12024   4088   2972 S   0.0   0.1   0:00.01 sshd
    345 root      20   0 2021964  85060  61224 S   0.0   1.0   0:00.62 dockerd
    418 root      20   0 1234268  13432   8268 S   0.0   0.2   0:01.04 editor-proxy
    419 root      20   0   16328   6416   5488 S   0.0   0.1   0:00.01 sudo
    436 root      20   0 1226100   2696   1988 S   0.0   0.0   0:00.00 tmux-agent
    565 root      10 -10   13996   9796   8424 S   0.0   0.1   0:00.03 sshd
    615 root      10 -10   13996   9672   8300 S   0.0   0.1   0:00.02 sshd
    641 marvin7+  10 -10   14256   6448   4796 S   0.0   0.1   0:00.00 sshd

cloudshell ×  cloudshell ×

Welcome to Cloud Shell! Type "help" to get started, or type "gemini" to try prompting with Gemini CLI.
To set your Cloud Platform project in this session use `gcloud config set project [PROJECT_ID]`.
You can view your projects by running `gcloud projects list`.
marvin7valencia@cloudshell:~$ df -h
Filesystem                      Size  Used Avail Use% Mounted on
overlay                          95G   50G   45G  53% /
tmpfs                            64M     0   64M   0% /dev
/dev/sda1                        95G   50G   45G  53% /root
/dev/disk/by-id/google-home-part1  4.8G  157M  4.4G   4% /home
/dev/root                       2.0G  1.3G  715M  64% /usr/lib/modules
shm                              64M     0   64M   0% /dev/shm
marvin7valencia@cloudshell:~$ uptime
 16:39:29 up 29 min,  0 user,  load average: 0.06, 0.03, 0.05
marvin7valencia@cloudshell:~$
```

Figure 9: *Real-time process monitoring using the 'top' command.*

The top command (Figure 8) displays active processes and their CPU/RAM consumption. This allows the OS to manage multitasking and ensure system stability. Additionally, df -h provides information on disk space allocation.

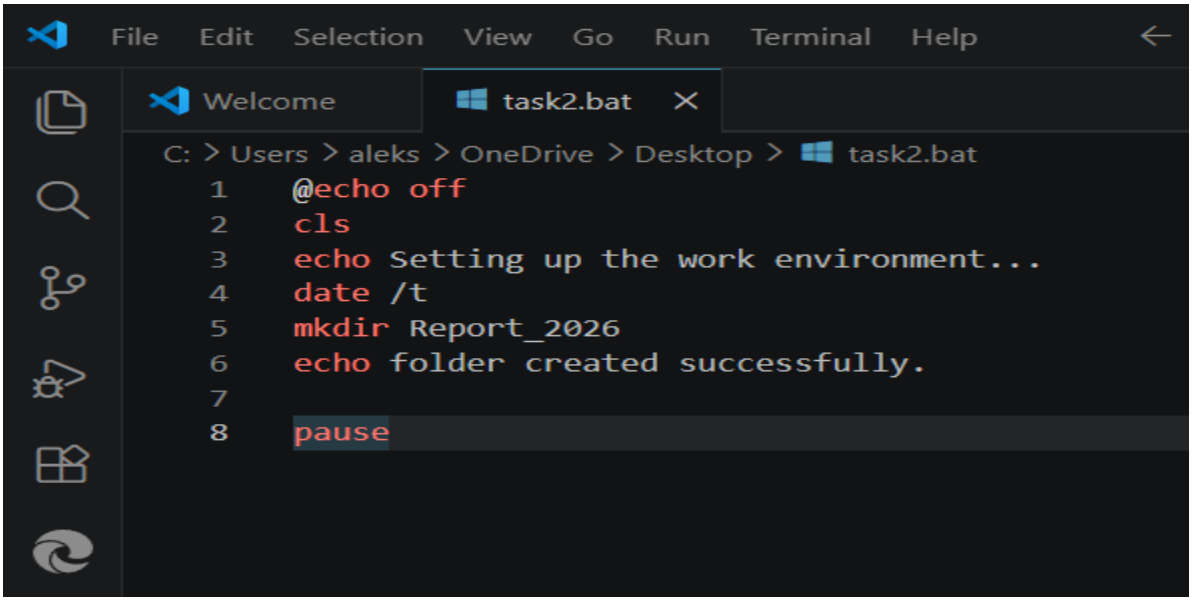
4.2 Network and user identification

```
cloudshell ×   cloudshell ×
Welcome to Cloud Shell! Type "help" to get started, or type "gemini" to try prompting with Gem
ini CLI.
To set your Cloud Platform project in this session use `gcloud config set project [PROJECT_ID]
`.
You can view your projects by running `gcloud projects list`.
marvin7valencia@cloudshell:~$ whoami
marvin7valencia
marvin7valencia@cloudshell:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: eth0@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1460 qdisc noqueue state UP group default q
len 1000
    link/ether d6:b8:7b:48:6c:3d brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.88.0.4/16 brd 10.88.255.255 scope global eth0
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1460 qdisc noqueue state DOWN group defaul
t
    link/ether 62:17:9f:3f:55:87 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
marvin7valencia@cloudshell:~$ uname -a
Linux cs-908712598781-default 6.6.123+ #1 SMP PREEMPT_DYNAMIC Sun Apr  5 08:56:29 UTC 2026 x86
_64 x86_64 x86_64 GNU/Linux
marvin7valencia@cloudshell:~$
```

Figure 10: *Identifying network interfaces and the current user.*

The ip addr command displays network configuration, which is vital for communication functions. Finally, whoami confirms the active user session, reflecting the OS's role in user management and security.

5. Use of Linux OS Scripts



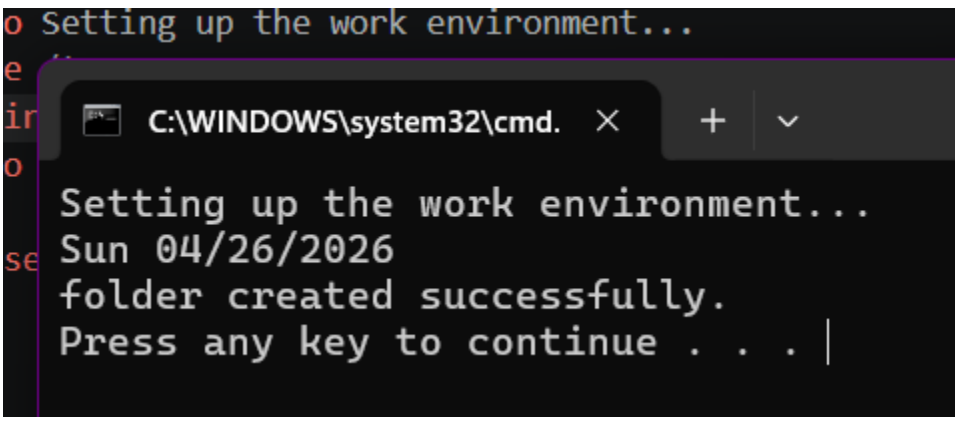


Figura11

#!/bin/bash: This is the shebang that tells the system to execute the file using the Bash interpreter.

echo: Prints text to the terminal to interact with the user.

$(whoami): This variable retrieves the username of the currently logged-in user.

ls -F: Lists files by adding a trailing symbol (like /) to distinguish folders from regular files.

Tip in Cloud Shell: Type `nano info.sh`, paste the code, save with Ctrl+O, and exit with Ctrl+X. Then run it with `bash info.sh`.

6. SCRIPTS AND AUTOMATION

Finally, I developed scripts, which act as automated "instruction lists" for the computer.

- ***Windows (Batch):*** *I created a .bat script to clear the console, display the date, and prepare a reports folder automatically. This type of automation is a cornerstone of modern IT efficiency.*
- ***Linux (Bash):*** *My info.sh script used the shebang #!/bin/bash to execute a sequence of commands that identify the user and list folder contents. This demonstrates how scripting can eliminate human error in repetitive tasks.*

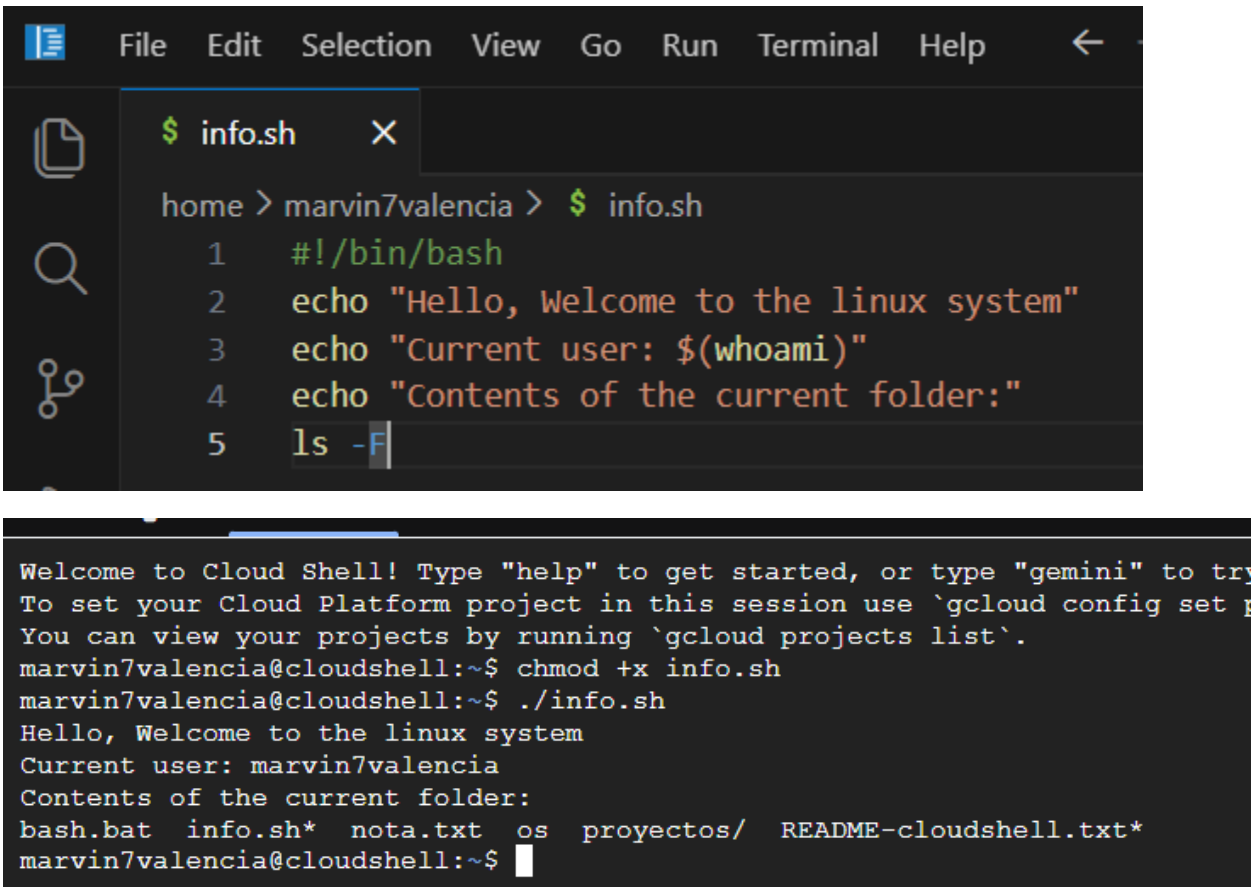


Figure 12: Successful execution of the script in the terminal.

#!/bin/bash: Known as the shebang, it tells the operating system to use the Bash interpreter to process commands.

echo: Used to print text to the console and communicate with the user.

$(whoami): A command substitution that allows you to dynamically insert the current user's name into the message.

ls -F: Lists the files in the current directory, adding a visual indicator to differentiate folders from files.

Conclusion

Writing this report helped me understand that an Operating System is much more than the interface we see on a computer screen. It acts as the central controller that manages hardware resources, coordinates processes, and maintains system stability.

Using tools such as Task Manager and Linux monitoring commands allowed me to observe how the operating system distributes CPU and memory resources between applications. This practical experience made the theory easier to understand because I could see process management happening in real time.

Working with command-line interfaces in both Windows and Linux was initially challenging, but it quickly became clear why system administrators rely heavily on these tools. Commands such as `ipconfig`, `tasklist`, and `top` provide fast and detailed access to system information that graphical interfaces sometimes hide.

The scripting section was the most interesting part of the assignment because it demonstrated how simple automation can improve productivity. Creating `.bat` and `.sh` scripts showed me that repetitive tasks can be completed more efficiently with only a few lines of code.

Overall, this assignment provided me with a stronger understanding of how operating systems function internally and how important they are in both personal and professional computing environments.

REFERENCES

[1] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, 10th ed. Hoboken, NJ: Wiley, 2018.

[2] W. Stallings, *Operating Systems: Internals and Design Principles*, 9th ed. Upper Saddle River, NJ: Pearson, 2017.

[3] Google's Gemini AI, private communication, April 2026.

[4] Microsoft Learn, "Windows Commands," Microsoft, 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/

[5] Red Hat, "What is Linux?", 2024. Available: https://www.redhat.com/en/topics/linux/what-is-linux

APPENDICES

Understanding OS Types: A Quick & Simple Guide

1. The "Old School" Batch OS

Back in the day, you couldn't just sit down and type on a computer. You had to prepare your work on cards and give them to an operator.

It’s like a professional laundry service. You drop off a "batch" of clothes, they wash everything together, and you pick it up later. There’s **zero interaction** while the machine is working.

2. The "Fair Sharer" (Time-Sharing OS)

This was a game changer. It allows many people to use the same powerful computer at the same time.

Imagine a fast-food chef jumping between three different burgers. He flips one, then moves to the next so fast that every customer feels like their burger is the only one being cooked. It uses **"time slices"** to keep everyone happy.

3. The "Team Players" (Distributed OS)

Instead of one big brain, this system uses many different computers connected together. To the user, it feels like just one super-fast machine. If one computer in the group crashes, the others pick up the slack. It’s all about **collaboration and reliability**.

4. The "Office Manager" (Network OS)

This one lives on a server. Its whole job is to manage a group of other computers (clients), making sure they can share files and stay secure.

It's like the administrator in an office who holds the keys to the file cabinet and decides who is allowed to enter which room.

5. The "No-Mistakes" RTOS (Real-Time OS)

This is the most "intense" one. These systems are used when timing is a matter of life or death literally.

- **Hard Real-Time:** Think of an **airbag** in a car. If it’s even a millisecond late, it’s useless.
- **Soft Real-Time:** Think of **streaming a movie**. If it lags for a second, it's annoying, but nobody gets hurt.

ONLINE CERTIFIED COURSE

Cisco
Networking
Academy

This certificate is awarded to

Marvin Alexander valencia

for successfully completing

Operating Systems Basics

offered by Networking Academy
through the Cisco Networking Academy program.

Lynn Bloomer
Director
Cisco Networking Academy

04 May 2026
Completion Date

Cert ID: f5587ad4-d4ef-471c-a958-7fb678e230fe