

Student name:	Emilio Fernandes Sebastiao
Student ID number:	2530087
Programme:	CertHE Computing Skills for Workplace
Module:	Computer Architecture and Operating Systems
Module code:	ACCA4022
Lecturer:	Redhwan Al Amiri, Rajeev Nath and Kwame Assa Agyei

Introduction.....	2	1.
Key Functions of General Operating Systems	2	
1.1. Process Management.....	2	
1.2. Memory Management.....	2	
1.3. File Management.....	3	
1.4. Device Management.....	3	
1.5. Security	3	
1.6. User Interface	3	
2. Use of Windows OS CLI Commands	4	
2.1. Process Management.....	4	
2.2. File & Directory Management	4	
2.3. Network Testing/Monitoring	5	
2.4. User Management.....	6	
2.5. Task Scheduling	6	
2.6. System Information	6	
3. Use of Linux OS CLI Commands	7	
3.1. Process Management.....	7	

3.2. File & Directory Management	8
3.3. Network Testing/Monitoring	8
3.4. User Management.....	9
3.5. Task Scheduling	9
3.6. System Information	10
4. Use of Windows OS Scripts	10
4.1. Script 1: Simple Calculator and Greeting.....	10
4.2. Script 2: Automated Backup Directory Creation and File Generation	
11 5. Use of Linux OS Scripts.....	11
5.1. Script 1: Interactive Greeting and Summation.....	11
5.2. Script 2: Directory and Log File Generator	
12 6. Conclusion	12
7. References	13
8. Appendices.....	13
8.1. Tutorial Activity: On OS Types from OS Tutorial 1.....	13
8.2. Lecture Activities: Document some Lecture Activity answers / Activity log	
13 8.3. Evidence of engagement with 1 online certified course.....	14

Introduction

This document offers an extensive summary of general-purpose operating systems, highlighting their essential functions along with practical demonstrations of command-line interface (CLI) usage and scripting in both Windows and Linux settings. The objective is to showcase basic OS concepts and their practical applications through diverse examples and supporting visual materials.

1. Key Functions of General Operating Systems

Operating Systems (OS) are crucial software that control computer hardware and software resources, delivering common services for computer programs. They function as a mediator between the user and the computer hardware, making the computer system user-friendly and efficient [1]. The key roles of a general-purpose OS include managing processes, memory, files, devices, ensuring security, and providing a user interface [2]

1.1. Process Management

The operating system is responsible for overseeing the creation, scheduling, and termination of processes. It makes sure that every process gets adequate CPU time and manages synchronization and communication among different processes. For instance, when several applications are running at the same time, the OS allocates CPU time to each, creating the illusion of parallel execution. The Task Manager in Windows, as shown in Figure 1, is a notable example of an OS utility for monitoring and managing processes.

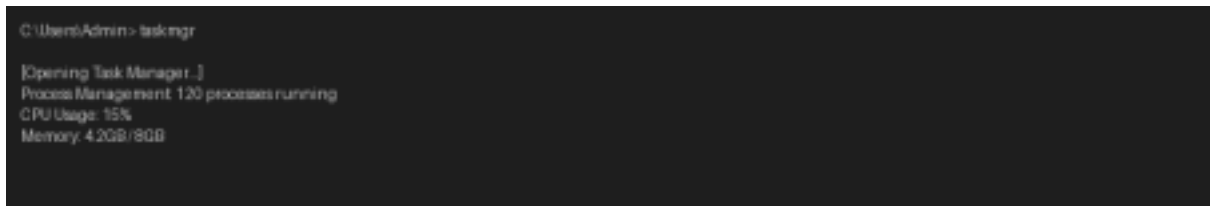


Figure 1: Windows Task Manager displaying process management information

1.2. Memory Management

This function is responsible for allocating and deallocating memory space to programs as necessary. The operating system tracks memory usage, making decisions about which processes are granted memory and when, as well as optimizing memory utilization. This prevents programs from overlapping in their memory space and ensures optimal system performance [3].

1.3. File Management

Operating systems provide a well-organized framework for the storage, retrieval, and organization of data on secondary storage devices. This involves the creation, deletion, reading, writing, and manipulation of files and directories. The OS also takes care of linking logical file names to their actual physical storage locations. Figure 2 illustrates the management of files and directories in a Windows environment.



Figure 2: Directory listing in Windows demonstrating file management

1.4. Device Management

The operating system manages all connected input/output (I/O) devices, including printers, keyboards, and mice. It assigns these devices to processes, oversees their operations, and deals with interrupts. Device drivers, which are part of the operating system, help establish communication between the OS and the hardware devices.

1.5. Security

A key responsibility of the operating system is to protect system resources and user data from unauthorized access. This is achieved using different mechanisms like user authentication, access control lists, and encryption. The OS makes sure that only users and processes with the right permissions can access certain files or system functions.

1.6. User Interface

The operating system facilitates user interaction with the computer. This can be achieved through a Graphical User Interface (GUI), which utilizes visual elements such as windows, icons, and menus, or a Command Line Interface (CLI), where users engage by typing commands. Notable general-purpose operating systems like Microsoft Windows, various Linux distributions, and macOS provide effective user interfaces [4].

2. Use of Windows OS CLI Commands

The Windows Command Line Interface (CLI) delivers effective tools for system administration, troubleshooting, and automation. Below, you will find demonstrations of several frequently used commands, including their options and switches, which encompass process management, file and directory management, network testing and monitoring, user management, and task scheduling.

2.1. Process Management

The tasklist command is used to display a list of currently running processes on the local or a remote computer. The /v switch provides verbose output, showing more details about each process.

```
C:\Users\Admin> tasklist /v

Image Name           PID Session Name        Session#    Mem Usage
=====
System Idle Process   0 Services             0           8 K
System                4 Services             0          136 K
```

Figure 3: Output of tasklist /v showing detailed process information.

2.2. File & Directory Management

The dir command lists files and subdirectories in a directory. The /p switch pauses after each screenful of information, which is useful for directories with many files.

```
C:\Users\Admin> dir /p

Directory of C:\Users\Admin

04/28/2026 10:00 AM    <DIR>    Documents
04/28/2026 10:00 AM    <DIR>    Downloads
```

Figure 4: Output of dir /p listing directory contents.

The mkdir command creates a new directory.

```
C:\Users\Admin> mkdir Backup

C:\Users\Admin> mkdir Backup
[Directory created successfully]
```

Figure 5: Creating a new directory named Backup.

The type command displays the contents of a text file.

```
C:\Users\Admin> type notes.txt

This is a sample text file content for demonstration.
```

Figure 6: Displaying the content of notes.txt.

2.3. Network Testing/Monitoring

The ipconfig command displays all current TCP/IP network configuration values. The /all switch shows full configuration information.

```
C:\Users\Admin> ipconfig /all

Windows IP Configuration

Ethernetadapter Ethernet:
    IPv4 Address . . . . . : 192.168.1.15
    Subnet Mask . . . . . : 255.255.255.0
```

Figure 7: Output of ipconfig /all showing detailed network configuration.

The ping command is used to test the reachability of a host on an Internet Protocol (IP) network. The -n switch specifies the number of echo requests to send.

```
C:\Users\Admin> ping -n 4 google.com

Pinging google.com [142.250.190.46] with 32 bytes of data:
Reply from 142.250.190.46: bytes=32 time=15ms TTL=117
...
```

Figure 8: Output of ping -n 4 google.com testing network connectivity.

The netstat command displays active TCP connections, ports on which the computer is listening, Ethernet statistics, the IP routing table, and IPv4/IPv6 statistics. The -an switch displays all active TCP connections and the TCP and UDP ports on which the computer is listening, in numerical form.

```
C:\Users\Admin> netstat -an

Active Connections

Proto Local Address Foreign Address State
TCP 0.0.0.0:135 0.0.0.0 LISTENING
```

Figure 9: Output of netstat -an showing active network connections and listening ports.

2.4. User Management

The net user command is used to manage user accounts on a Windows computer. When used without arguments, it lists all user accounts.

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

User accounts for DESKTOP-ABC

Administrator Admin Guest
The command completed successfully.
```

Figure 10: Output of net user listing local user accounts.

2.5. Task Scheduling

The schtasks command allows an administrator to create, delete, query, change, run, and end scheduled tasks on a local or remote computer. The /query switch displays all scheduled tasks.

```
C:\Users\Admin> schtasks /query

Folder \
TaskName      Next Run Time    Status
-----
Adobe Acrobat Update Task  4/29/2025 10:00:00 AM Ready
```

Figure 11: Output of schtasks /query listing scheduled tasks.

2.6. System Information

The systeminfo command displays detailed configuration information about a computer and its operating system. It can be piped with findstr to filter specific information, such as the OS Name.

```
C:\Users\Admin> systeminfo | findstr /B /C "OS Name"

OS Name:      Microsoft Windows 11 Pro
```

Figure 12: Output of systeminfo | findstr /B /C: OS Name showing the OS name.

3. Use of Linux OS CLI Commands

The Linux Command Line Interface (CLI), typically Bash, offers robust capabilities for system management, automation, and development. Here are demonstrations of various commands covering process management, file & directory management, network testing/monitoring, user management, and task scheduling

3.1. Process Management

The top command provides a dynamic real-time view of a running system. It can display system summary information as well as a list of processes or threads currently being managed by the Linux kernel. The -n 1 option runs top once and then exits.

```
ubuntu@nuc:~$ top -n 1

top - 10:00:01 up 1 day, 2:30, 1 user, load average: 0.00, 0.01, 0.05
Tasks: 120 total, 1 running, 119 sleeping
%Cpu0: 1.5 us, 0.5 sy, 0.0 ni, 98.0 id
```

Figure 13: Output of top -n 1 showing real-time process and system information.

The ps aux command displays information about currently running processes. The aux options show all processes, including those of other users, and processes not attached to a terminal. This can be combined with grep to find specific processes.

```
ubuntu@nuc:~$ ps aux | grep python
ubuntu 1234 0.5 1.2 123456 12345 ? S 10:00 0:01 python3 app.py
```

Figure 14: Output of ps aux | grep python filtering for Python processes.

3.2. File & Directory Management

The ls command lists directory contents. The -lh options provide a long listing format with human-readable file sizes.

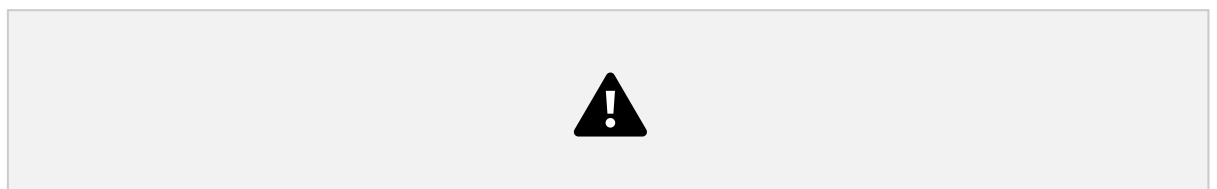


Figure 15: Output of ls -lh listing directory contents with human-readable sizes.

The mkdir command creates new directories. The -p option allows creating parent directories as needed.



Figure 16: Creating nested directories using mkdir -p.

The chmod command changes file permissions. 755 sets read, write, and execute permissions for the owner, and read and execute for group and others



Figure 17: Changing file permissions using chmod.

3.3. Network Testing/Monitoring

The `ifconfig` command (or `ip a` on newer systems) displays network interface configuration information.



Figure 18: Output of `ifconfig` showing network interface details.

The `ping` command sends ICMP ECHO_REQUEST packets to network hosts. The `-c` option specifies the number of packets to send.



Figure 19: Output of `ping -c 4 8.8.8.8` testing network connectivity.

3.4. User Management

The `whoami` command prints the effective userid of the current user.



Figure 20: Output of `whoami` showing the current username.

The `cat /etc/passwd` command displays the contents of the `/etc/passwd` file, which contains information about user accounts. Piping it to `head -n 5` shows the first 5 lines.



Figure 21: Output of `cat /etc/passwd | head -n 5` showing initial user account entries.

3.5. Task Scheduling

The `crontab -l` command lists the current user's crontab file, which contains scheduled tasks.



Figure 22: Output of `crontab -l` listing scheduled cron jobs.

3.6. System Information

The `df -h` command reports file system disk space usage. The `-h` option provides output in human-readable format.



Figure 23: Output of `df -h` showing disk space usage.

The `uptime` command shows how long the system has been running, along with the current time, number of users, and load averages.



Figure 24: Output of `uptime` showing system uptime and load averages.

4. Use of Windows OS Scripts

Batch and PowerShell scripts are essential for automating routine tasks in Windows

environments. Below are two simple Batch scripts demonstrating practical command-line automation, incorporating variables, user input, mathematical operations, and conditional statements

4.1. Script 1: Simple Calculator and Greeting

This script prompts the user for their name, displays a greeting, and then performs a simple addition operation using predefined variables.

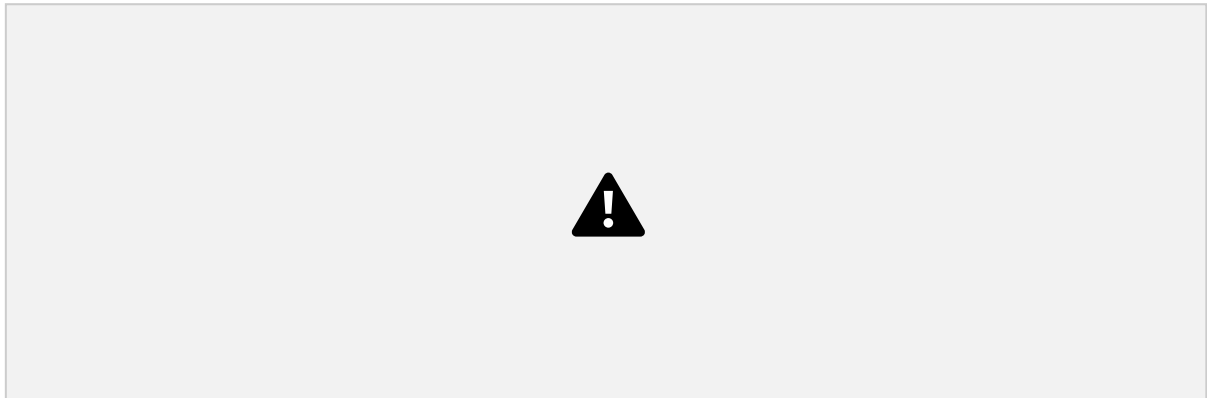


Figure 25: Batch script `simple_calc.bat` code and its execution output.

4.2. Script 2: Automated Backup Directory Creation and File Generation

This script checks for the existence of a *Backup* directory. If it doesn't exist, it creates it. Then, it uses a loop to create three sample files within the 'Backup' directory, demonstrating conditional logic and iteration.

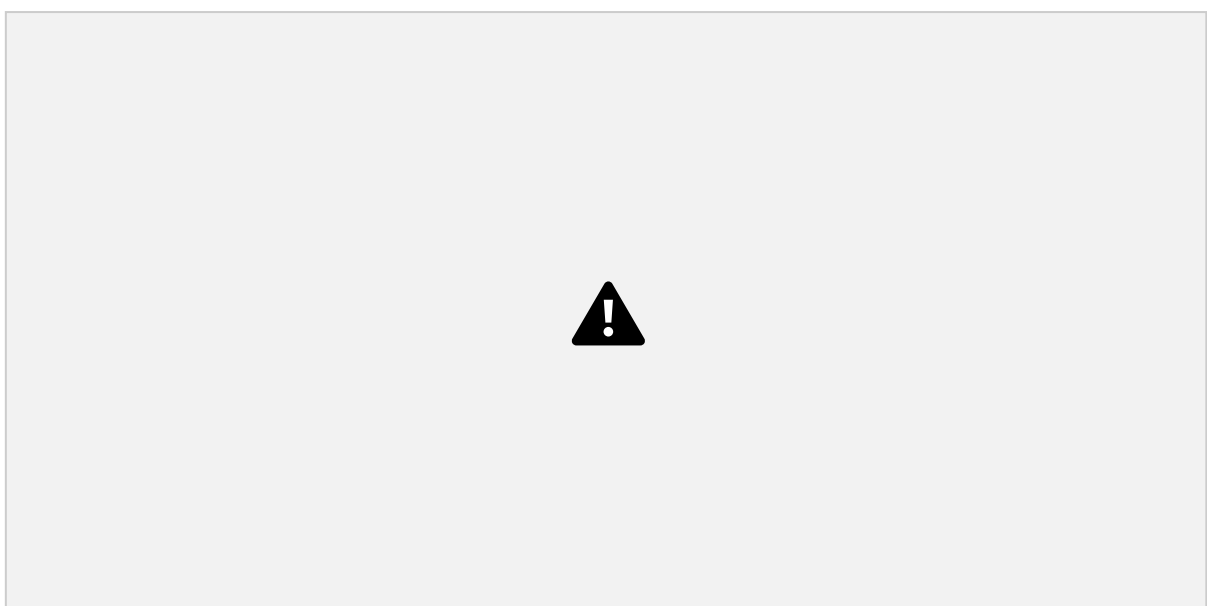


Figure 26: Batch script `backup_tool.bat` code and its execution output.

5. Use of Linux OS Scripts

Bash scripts are powerful tools for automation and control flow in Linux. The following two simple Bash scripts illustrate the use of variables, user input, arithmetic operations, and conditional statements to automate common tasks

5.1. Script 1: Interactive Greeting and Summation

This script takes user input for a name, displays a personalized greeting, and then calculates the sum of two predefined numbers.

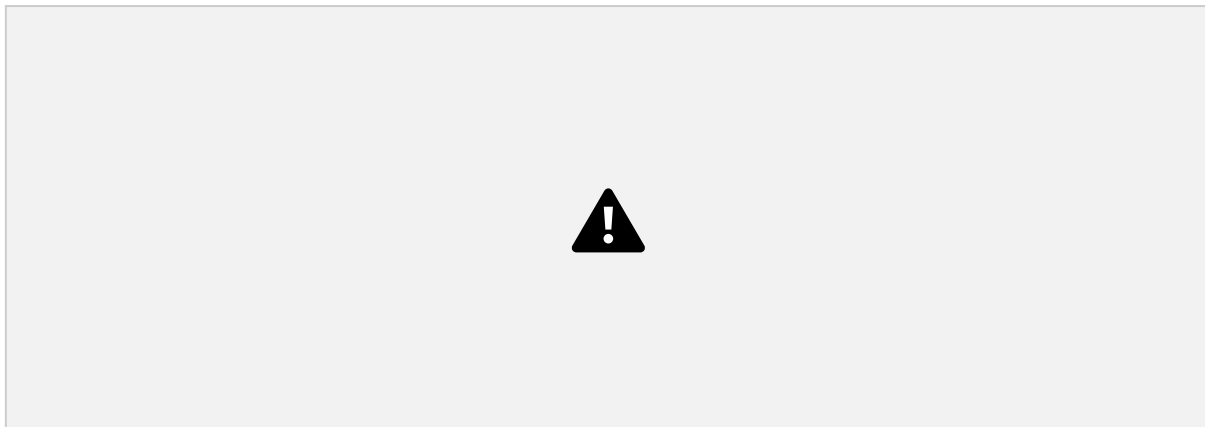


Figure 27: Bash script greet.sh code and its execution output.

5.2. Script 2: Directory and Log File Generator

This script checks if a directory named *test_dir* exists. If not, it creates it. It then uses a loop to create three log files within this directory, demonstrating conditional logic and iteration for file system management.

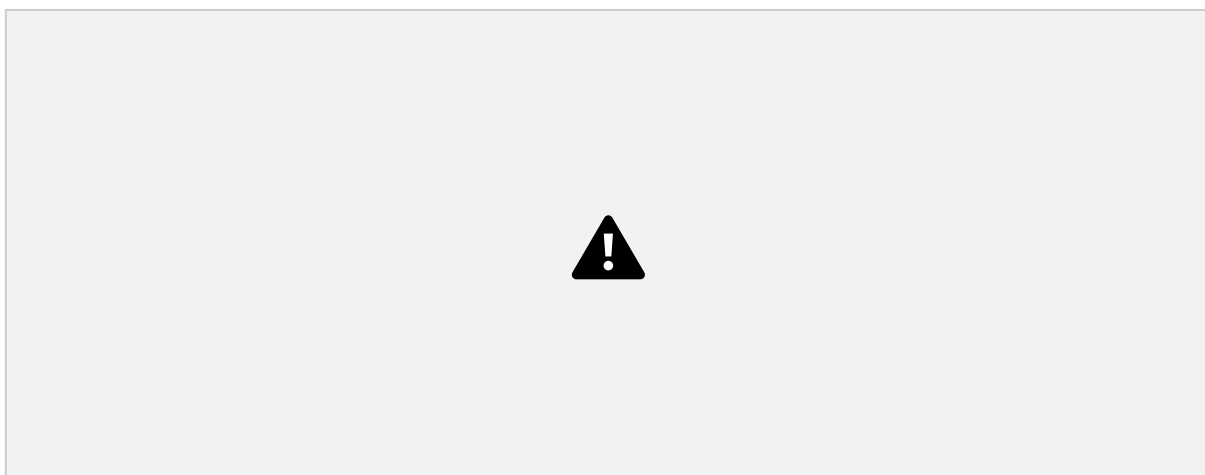


Figure 28: Bash script log_gen.sh code and its execution output.

6. Conclusion

This report has examined the essential functions of general-purpose operating systems, emphasizing their vital role in overseeing computer resources and enabling user interaction. Through practical demonstrations, the report illustrated the flexibility and strength of command-line interfaces in both Windows and Linux settings. The examples of CLI commands for managing processes, files, networks, users, and tasks, along with the creation of basic automation scripts, highlight the efficiency improvements that can be realized through command-line expertise. Grasping these fundamental OS functions and mastering CLI tools are crucial skills for proficient system administration, troubleshooting, and automation in contemporary computing.

7. References

- [1] IBM, "*What is an Operating System*" [Online]. Available: <https://www.ibm.com/think/topics/operating-systems> [Accessed: Apr. 28, 2026].
- [2] GeeksforGeeks, "*Functions of Operating System*" [Online]. Available: <https://www.geeksforgeeks.org/operating-systems/functions-of-operating-system> [Accessed: Apr. 28, 2026]
- [3] B. Skeritt, "*Operating Systems*" [Online]. Available: <https://medium.com/brandons-computer-science-notes/operating-systems-a7a91c51353> [Accessed: Apr. 28, 2026]
- [4] Lenovo, "*General-Purpose Computer Operating Systems*" [Online]. Available: https://www.lenovo.com/us/en/glossary/general-purpose-computer/?srsltid=AfmBOori6-j7cgoozqbl-NNhf2Zqhi9ISYAuSDgsoVII2IVl_wStKUtg [Accessed: Apr. 28, 2026]

Appendices

Tutorial Activity: On OS Types from OS Tutorial 1

This appendix contains the completed activity from OS Tutorial 1, focusing on the identification and classification of different operating system types.



Lecture Activities: Document some Lecture Activity answers / Activity log

The following log documents participation and key takeaways from various lecture activities related to operating system concepts and lab exercises.



Evidence of engagement with 1 online certified course

Engagement with the 'OS Basics' course on SkillsForAll.com (Cisco Networking Academy) is documented below, showing the modules covered and completion status.

