

Operating Systems Assignment

COMPUTER ARCHITECTURE AND OPERATING SYSTEMS
ACCA4022

TOM OWEN (2520277)

Contents

Introduction	3
Task 1: Key Functions of General Operating Systems	3
Task 2: Windows CLI Commands	8
Task 3: Linux CLI Commands	14
Task 4: Windows OS Scripts	18
Task 5: Linux OS Scripts	20
Conclusion	22
References	22
Appendices	24

Figure 1 Task Manager Processes Page	4
Figure 2 File Explorer	5
Figure 3 Device Manager	6
Figure 4 Resource Monitor	7
Figure 5 Windows Security Dashboard	8
Figure 6 Firewall & Network Protection	8
Figure 7 - Output of the tasklist command in Windows CLI	9
Figure 8 Output of the taskkill command in Windows CLI	9
Figure 9 Output of the dir command in Windows CLI	10
Figure 10 Output of the cd command in Windows CLI	10
Figure 11 Output of the mkdir command in Windows CLI	11
Figure 12 Output of the copy command in Windows CLI	11
Figure 13 Output of the ipconfig command in Windows CLI	12
Figure 14 Output of the ping command in Windows CLI	12
Figure 15 Output of the net user command in Windows CLI	13
Figure 16 Output of the schtasks command in Windows CLI	13
Figure 17 Output of the ps command in Linux CLI	14
Figure 18 Output of the kill command in Linux CLI	15
Figure 19 Output of the ls command in Linux CLI	15
Figure 20 Output of the cd command in Linux CLI	16
Figure 21 Output of the mkdir command in Linux CLI	16
Figure 22 Output of the cp command in Linux CLI	16
Figure 23 Output of the ip a command in Linux CLI	17
Figure 24 Output of the ping command in Linux CLI	17
Figure 25 Output of the who command in Linux CLI	17

Figure 26 Output of the crontab -l command in Linux CLI	18
Figure 27 Output of the Batch variables script	18
Figure 28 Batch script showing the use of variables	18
Figure 29 Batch script showing user input, IF ELSE and goto	19
Figure 30 Output of the Batch user input script.....	19
Figure 31 Batch script showing a for loop	20
Figure 32 Output of the Batch for loop script	20
Figure 33 Bash script showing the use of variables	20
Figure 34 Output of the Bash variables script	21
Figure 35 Bash script showing user input and an IF statement	21
Figure 36 Output of the Bash user input script	21
Figure 37 Bash script showing a loop	22
Figure 38 Output of the Bash loop script	22
Figure 39 LabEx Quick Start with Linux course progress page	26
Figure 40 LabEx Linux terminal activity and completed lab badge	27
Figure 41 LabEx Basic Files Operations command verification.....	27
Figure 42 LabEx file management command activity.....	28
Figure 43 LabEx completed Basic Files Operations badge	28

Introduction

An operating system is a vital form of system software that acts as an interface between computer hardware and the user, enabling the effective operation and management of a computer system. Modern operating systems provide a range of important functions, including memory and process management, file and directory management, device management, system monitoring and security. This report examines these key operating system functions and demonstrates the practical use of CLI (command line interfaces) in both Windows and Linux environments. In addition, the report explores the use of Batch and Bash scripting to illustrate how administrative and automated tasks can be carried out within different operating systems.

Task 1: Key Functions of General Operating Systems

General purpose operating systems such as Windows, Linux and macOS perform a range of core functions that allow computer systems to run efficiently and securely. These functions help the operating system to manage hardware resources, support software execution, organise files and directories, control connected devices, monitor system performance and protect the system from threats. Although these tasks are carried out largely in the background, they are essential to the stable day-to-day operation of a computer system. [1]

1.1 Memory and Process Management

Memory and process management are part of the complex subsystems that run in the background to ensure smooth operation. Process management is the part of the OS that manages processes that are running simultaneously. Memory management is the

part of the operating system that allocates RAM to programmes. The OS can change the amount of allocation depending on the type of task or programme being performed on the PC and prevent a programme from overusing RAM and resources. The amount of memory being used and the processes running can be monitored using Task Manager on a Windows OS. [1] , [6]

Processes		26% CPU	85% Memory	3% Disk	0% Network
Name	Status				
Apps (7)					
Google Chrome (18)		0%	374.2 MB	0 MB/s	0 Mbps
Microsoft PowerPoint (2)		0.1%	195.4 MB	0 MB/s	0 Mbps
Microsoft Word (2)		0%	87.7 MB	0 MB/s	0 Mbps
Outlook (9)		0%	161.4 MB	0.1 MB/s	0 Mbps
Task Manager		3.3%	74.4 MB	0.1 MB/s	0 Mbps
Windows Explorer		0%	60.4 MB	0.1 MB/s	0 Mbps
Windows Explorer		6.7%	4.6 MB	0 MB/s	0 Mbps
Background processes (114)					
64-bit Synaptics Pointing Enh...		0%	0.4 MB	0 MB/s	0 Mbps
AMD Crash Defender Service		0%	0.4 MB	0 MB/s	0 Mbps
AMD External Events Client M...		0%	0.9 MB	0 MB/s	0 Mbps
AMD External Events Service ...		0%	0.5 MB	0 MB/s	0 Mbps
AMD Software (4)		0.1%	14.4 MB	0 MB/s	0 Mbps
Antimalware Core Service		0%	2.4 MB	0 MB/s	0 Mbps
Antimalware Service Executable		0.6%	131.3 MB	0.1 MB/s	0 Mbps
App Actions		0%	2.2 MB	0 MB/s	0 Mbps
AppHelperCap		0%	1.1 MB	0 MB/s	0 Mbps
Application Frame Host		0%	1.1 MB	0 MB/s	0 Mbps
Bonjour Service		0%	1.3 MB	0 MB/s	0 Mbps
CommRecovery		0%	2.3 MB	0 MB/s	0 Mbps
Credential Guard & VBS Key Is...		0%	0.5 MB	0 MB/s	0 Mbps
CTF Loader		0%	3.2 MB	0 MB/s	0 Mbps

Figure 1 Task Manager Processes Page

1.2 File and Directory Management

File and directory management is a process performed by the operating system to organise data into appropriate areas. It allows the user to create files, rename, move, copy and delete files. This enables information to be maintained in a structured way, meaning it's easy to find files when needed. In addition to this, operating systems can help to manage permissions and access to files. Files can be protected with a password or only be displayed and accessible to a certain user on the OS. An example of file and directory management software would be File Explorer, found in Windows. [1], [8]

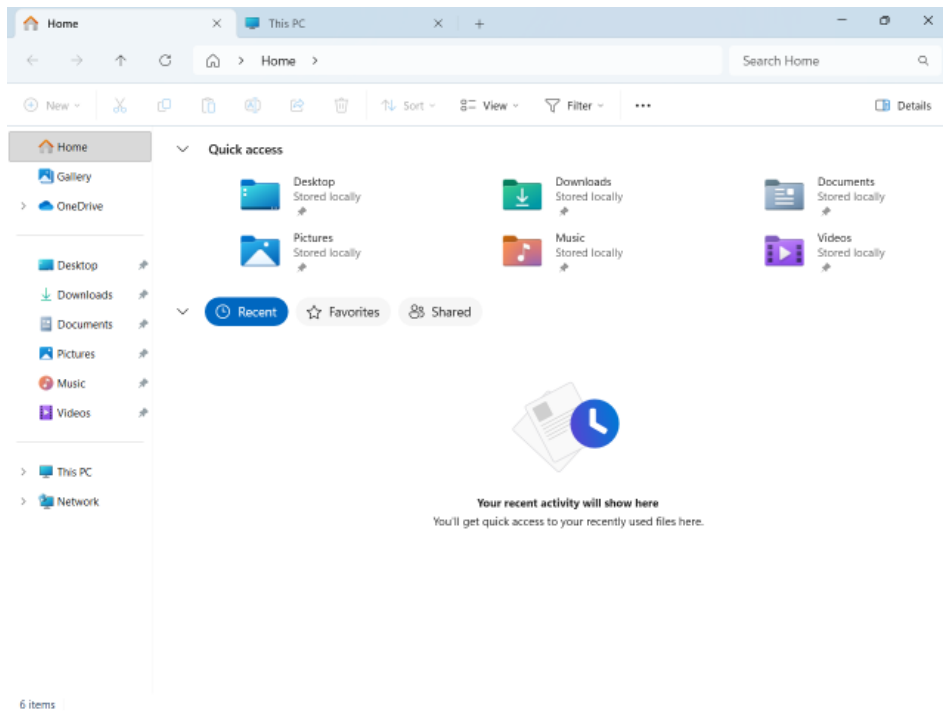


Figure 2 File Explorer

1.3 Device Management

Device management is an important function of the OS because it manages hardware devices that are connected to the computer. These can include devices like keyboards, mice, printers, monitors, storage drives and network adapters. For the OS to interpret connected devices, it requires the correct drivers to be installed for the peripherals. Most operating systems come with basic drivers, allowing for plug-and-play; however, dedicated drivers may be needed to take full advantage of an attached device's capability. [1], [6]

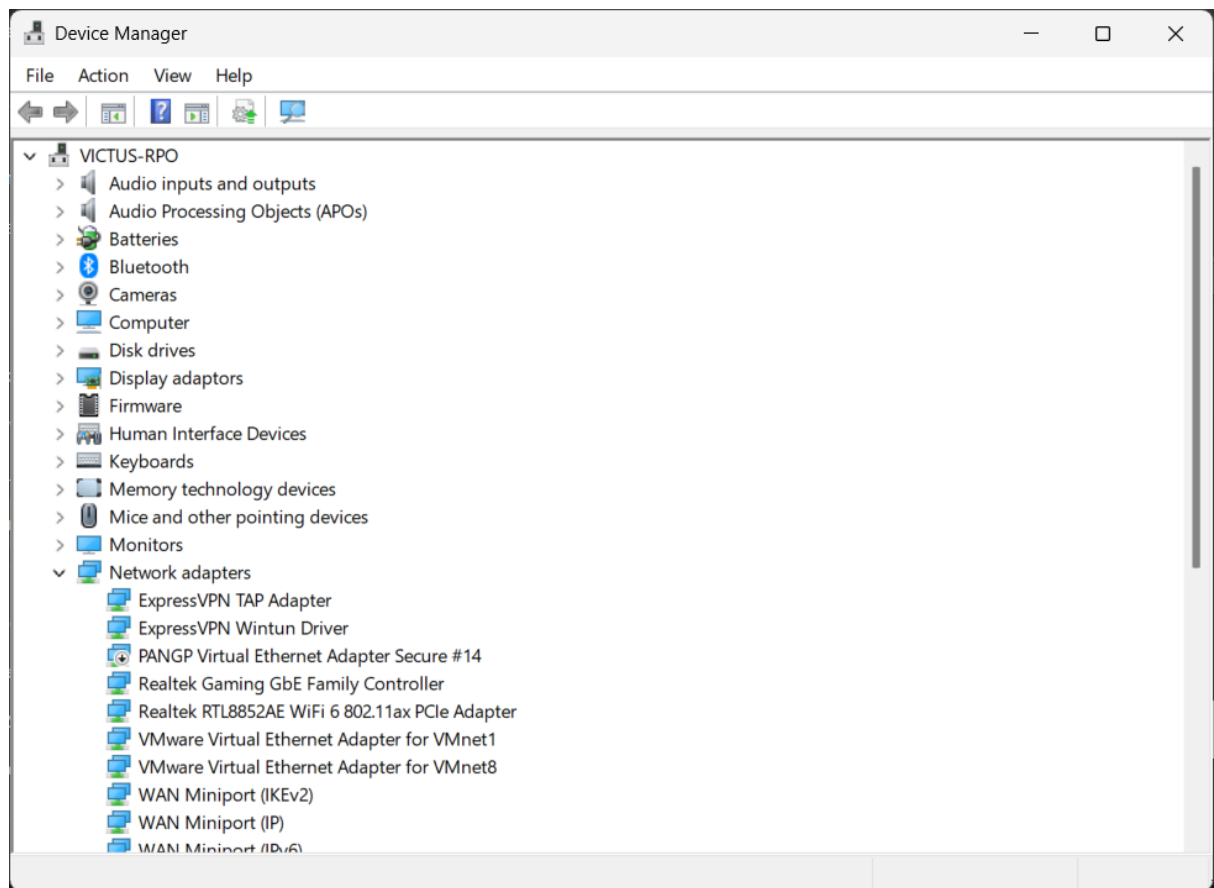


Figure 3 Device Manager

1.4 System Performance Monitoring

System performance monitoring is an important operating system function because it allows the system to track how efficiently its resources are being used. The operating system can monitor CPU performance, disk activity, memory usage and running processes. This helps identify slow performance, overloaded resources and processes that may not be responding correctly. Users can also use tools such as Resource Monitor and Task Manager to detect bottlenecks and terminate unresponsive programmes. [1], [6]

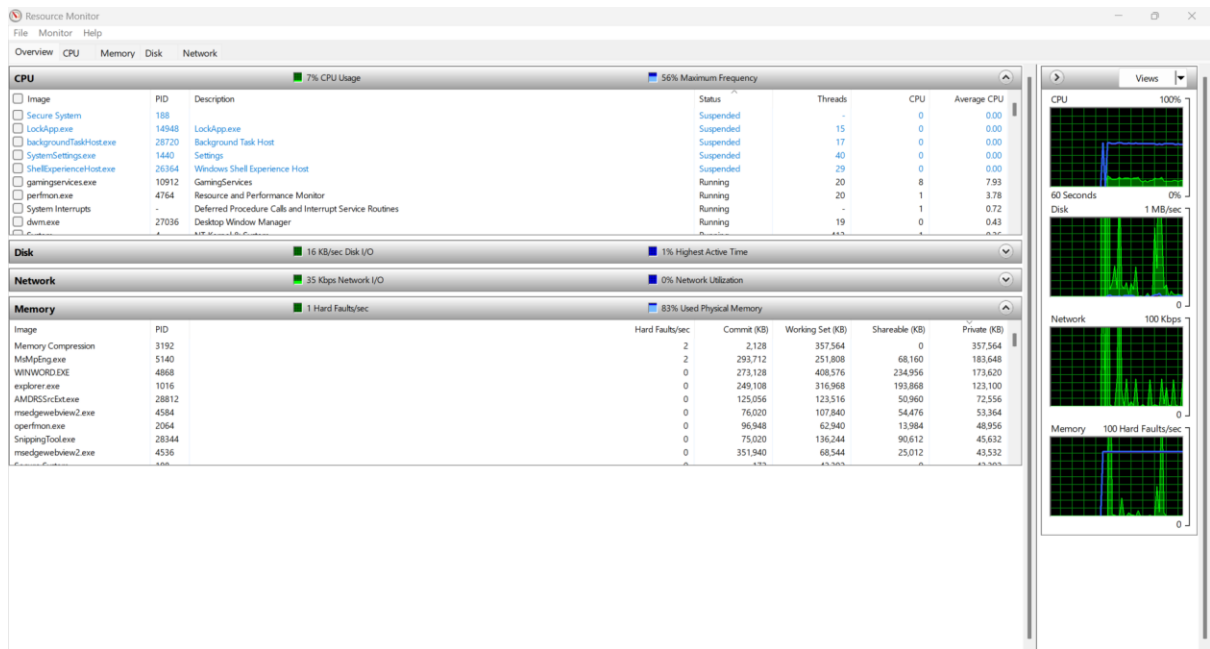


Figure 4 Resource Monitor

1.5 System Security

System Security helps the operating system protect the system from threats or unauthorised access. It does this through the use of user accounts, passwords, permissions (Group Policy Editor in Windows), firewalls, antivirus software such as Windows Defender, and with regular updates that patch vulnerabilities that may appear during the operating system's lifespan. Security is important to an operating system because it helps to protect files, programmes and personal data from being compromised. With these measures in place, it helps to reduce the risk of malware, remote attacks and potential misuse by a third party. In Windows, these functions can be viewed and managed through Windows Security and the firewall settings found on Windows 11/10 in the legacy Control Panel. [1], [9]

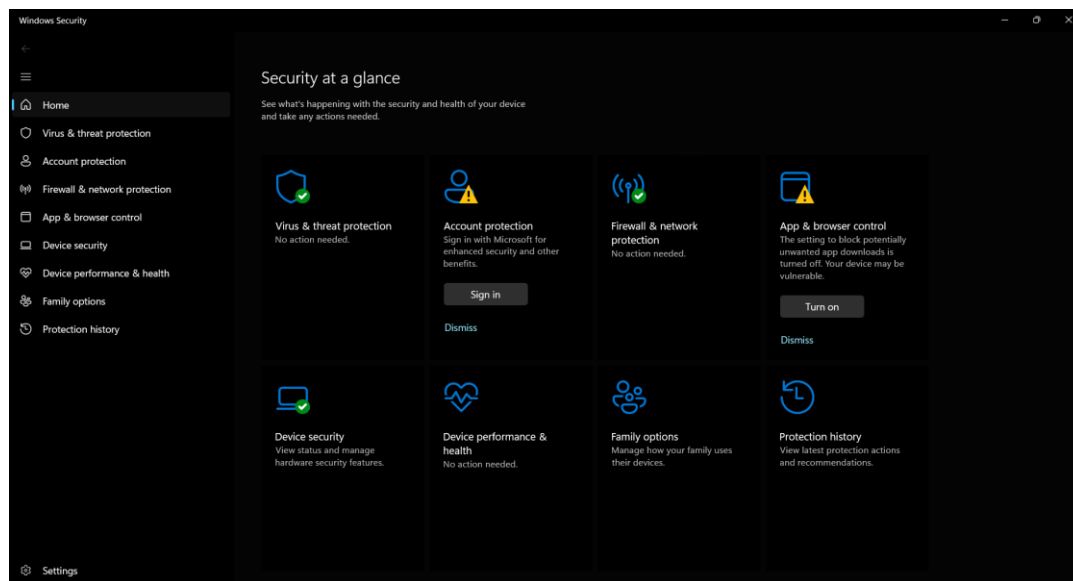


Figure 5 Windows Security Dashboard

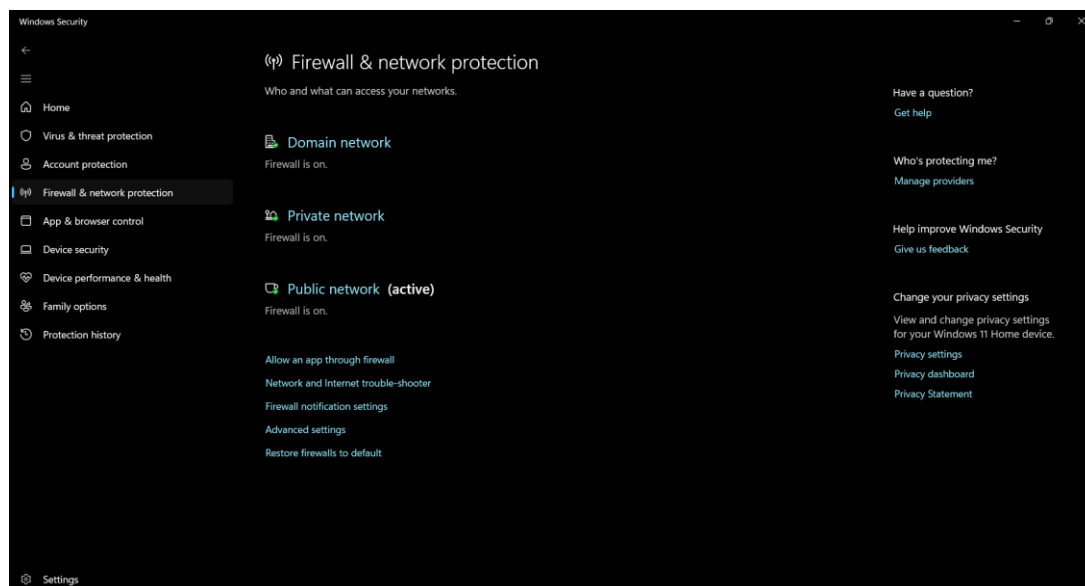


Figure 6 Firewall & Network Protection

Task 2: Windows CLI Commands

This section demonstrates the use of Windows CLI commands. These commands are often used by system administrators within organisations to carry out a range of management tasks. They can be used to monitor and manage areas such as process management, file and directory management, network testing and monitoring, user management, and task scheduling. This section will contain screenshots of these commands being run and showing their results. [2], [7]

2.1 tasklist

The tasklist command is used in Windows CLI to display the processes currently running on the system. It shows details such as the process name, process ID and memory usage. This is useful for checking active tasks and identifying processes using too many resources. [2], [8]

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	0 K
System	4	Services	0	584 K
Secure System	188	Services	0	43,380 K
Registry	232	Services	0	24,380 K
smss.exe	740	Services	0	8 K
csrss.exe	1204	Services	0	3,048 K
wininit.exe	1508	Services	0	540 K
services.exe	1508	Services	0	7,290 K
lsass.exe	1516	Services	0	32 K
svchost.exe	1536	Services	0	15,400 K
svchost.exe	1676	Services	0	25,280 K
fontdrvhost.exe	1708	Services	0	4 K
svchost.exe	1816	Services	0	14,544 K
svchost.exe	1856	Services	0	4,148 K
svchost.exe	2024	Services	0	684 K
svchost.exe	2064	Services	0	5,640 K
svchost.exe	2104	Services	0	2,528 K
svchost.exe	2112	Services	0	3,740 K
svchost.exe	2120	Services	0	6,304 K
svchost.exe	2128	Services	0	5,912 K
svchost.exe	2208	Services	0	2,836 K
svchost.exe	2276	Services	0	7,444 K
svchost.exe	2296	Services	0	1,056 K
svchost.exe	2388	Services	0	3,836 K
svchost.exe	2516	Services	0	10,132 K
svchost.exe	2756	Services	0	5,736 K
lsass.exe	2788	Services	0	56 K
symptehservice.exe	2868	Services	0	176 K
svchost.exe	2868	Services	0	4,376 K
svchost.exe	3048	Services	0	892 K
svchost.exe	3192	Services	0	4,936 K
WdDisplayContainer.exe	3880	Services	0	14,204 K
svchost.exe	3156	Services	0	3,128 K
ntls.exe	3220	Services	0	524 K
smfendrsr.exe	3228	Services	0	8 K
svchost.exe	3236	Services	0	7,368 K
svchost.exe	3324	Services	0	9,352 K
svchost.exe	3428	Services	0	4,880 K
svchost.exe	3512	Services	0	1,804 K
svchost.exe	3532	Services	0	3,116 K
svchost.exe	3736	Services	0	3,588 K
svchost.exe	3744	Services	0	9,680 K
svchost.exe	3752	Services	0	3,888 K
Memory Compression	3888	Services	0	260,768 K
svchost.exe	3928	Services	0	568 K
svchost.exe	3992	Services	0	4,040 K
svchost.exe	4000	Services	0	3,304 K
TouchpointAnalyticsClient	4000	Services	0	13,684 K
AppHelperCap.exe	4816	Services	0	3,964 K
lsass.exe	4824	Services	0	3,080 K
NetworkCap.exe	4832	Services	0	2,640 K
svchost.exe	4848	Services	0	8,764 K
svchost.exe	4380	Services	0	9,284 K
svchost.exe	4384	Services	0	5,964 K
svchost.exe	4556	Services	0	22,356 K
svchost.exe	4776	Services	0	13,488 K

Figure 7 - Output of the tasklist command in Windows CLI

2.2 taskkill

The taskkill command is used in Windows CLI to end a running process. It can stop a task by using its process ID (PID) or image name, for example “taskkill /IM notepad.exe /F”. This is useful for closing unresponsive programmes or tasks using too many resources. [2], [9]

```
Administrator: Command Prompt
Type "TASKKILL /?" for usage.

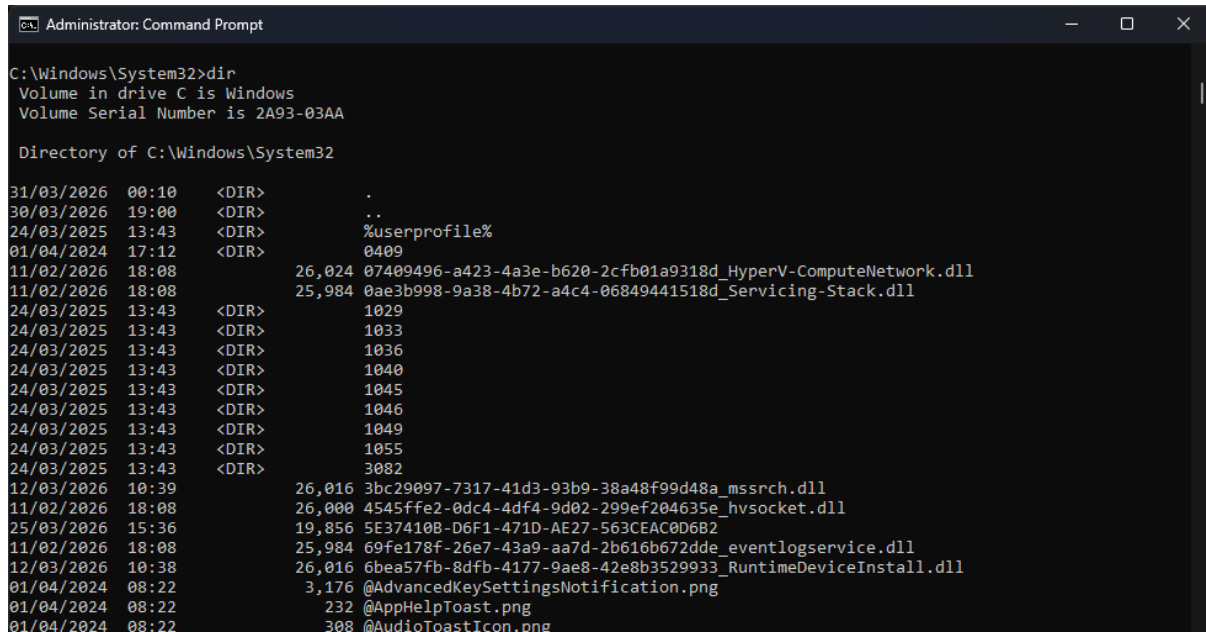
C:\Windows\System32>taskkill /IM notepad.exe /F
SUCCESS: The process "Notepad.exe" with PID 14844 has been terminated.

C:\Windows\System32>
```

Figure 8 Output of the taskkill command in Windows CLI

2.3 dir

The `dir` command is used in Windows CLI to display the files and folders in the current directory. It shows names, sizes and modified dates for items in that location. This is useful for checking directory contents without using File Explorer. [2], [7]



```
Administrator: Command Prompt
C:\Windows\System32>dir
Volume in drive C is Windows
Volume Serial Number is 2A93-03AA

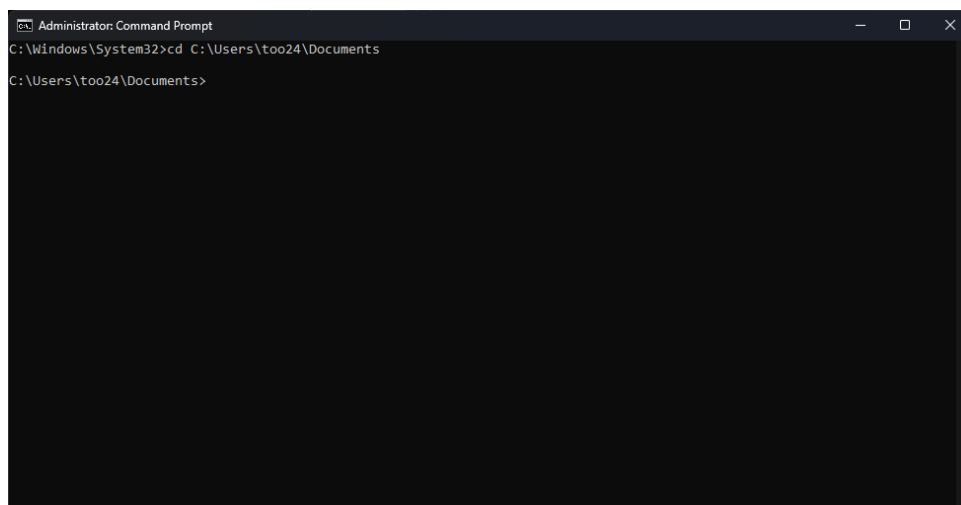
Directory of C:\Windows\System32

31/03/2026  00:10  <DIR>          .
30/03/2026  19:00  <DIR>          ..
24/03/2025  13:43  <DIR>          %userprofile%
01/04/2024  17:12  <DIR>          0409
11/02/2026  18:08                26,024  07409496-a423-4a3e-b620-2cfb01a9318d_HyperV-ComputeNetwork.dll
11/02/2026  18:08                25,984  0ae3b998-9a38-4b72-a4c4-06849441518d_Servicing-Stack.dll
24/03/2025  13:43  <DIR>          1029
24/03/2025  13:43  <DIR>          1033
24/03/2025  13:43  <DIR>          1036
24/03/2025  13:43  <DIR>          1040
24/03/2025  13:43  <DIR>          1045
24/03/2025  13:43  <DIR>          1046
24/03/2025  13:43  <DIR>          1049
24/03/2025  13:43  <DIR>          1055
24/03/2025  13:43  <DIR>          3082
12/03/2026  10:39                26,016  3bc29097-7317-41d3-93b9-38a48f99d48a_mssrch.dll
11/02/2026  18:08                26,000  4545ffe2-0dc4-4df4-9d02-299ef204635e_hvsocket.dll
25/03/2026  15:36                19,856  5E37410B-D6F1-471D-AE27-563CEAC0D682
11/02/2026  18:08                25,984  69fe178f-26e7-43a9-aa7d-2b616b672dde_eventlogservice.dll
12/03/2026  10:38                26,016  6bea57fb-8dfb-4177-9ae8-42e8b3529933_RuntimeDeviceInstall.dll
01/04/2024  08:22                 3,176  @AdvancedKeySettingsNotification.png
01/04/2024  08:22                 232  @AppHelpToast.png
01/04/2024  08:22                 308  @AudioToastIcon.png
```

Figure 9 Output of the `dir` command in Windows CLI

2.4 cd

The `cd` command is used in Windows CLI to change the current working directory. It allows the user to move between folders so commands can be run in the correct location. This is useful for file and folder navigation in the command line. [2], [7]

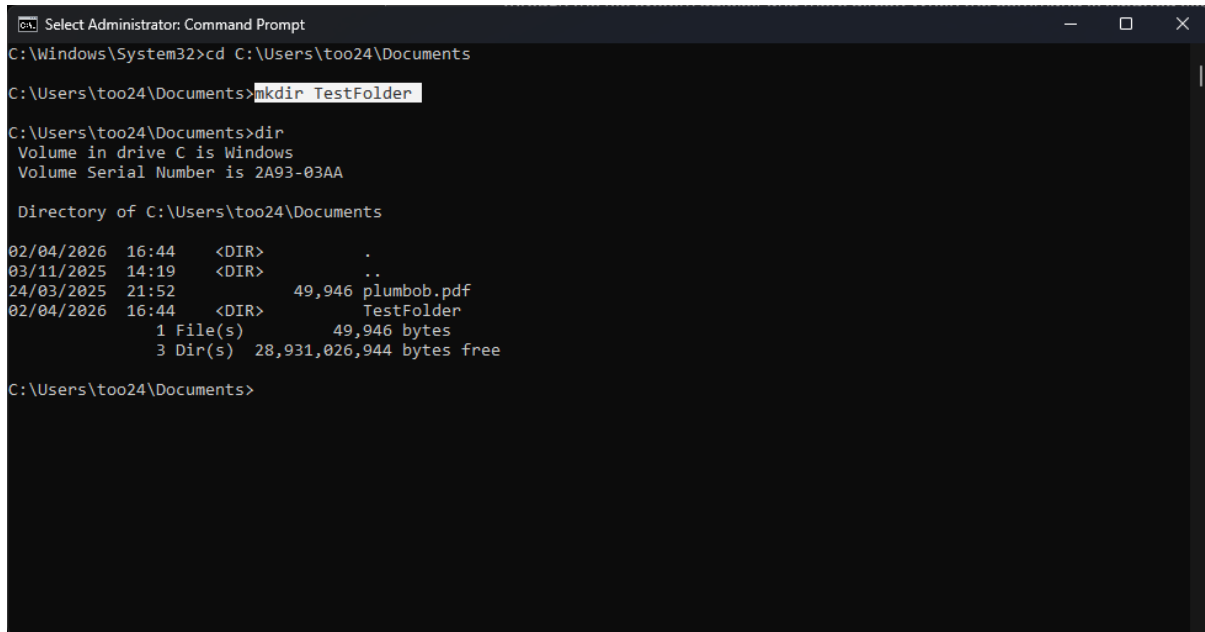


```
Administrator: Command Prompt
C:\Windows\System32>cd C:\Users\too24\Documents
C:\Users\too24\Documents>
```

Figure 10 Output of the `cd` command in Windows CLI

2.5 mkdir

The mkdir command is used in Windows CLI to create a new directory or folder. For example, “mkdir TestFolder” creates a folder with that name. This is useful for organising files and setting up new storage locations quickly. [2], [7]



```
Select Administrator: Command Prompt
C:\Windows\System32>cd C:\Users\too24\Documents
C:\Users\too24\Documents>mkdir TestFolder
C:\Users\too24\Documents>dir
Volume in drive C is Windows
Volume Serial Number is 2A93-03AA

Directory of C:\Users\too24\Documents

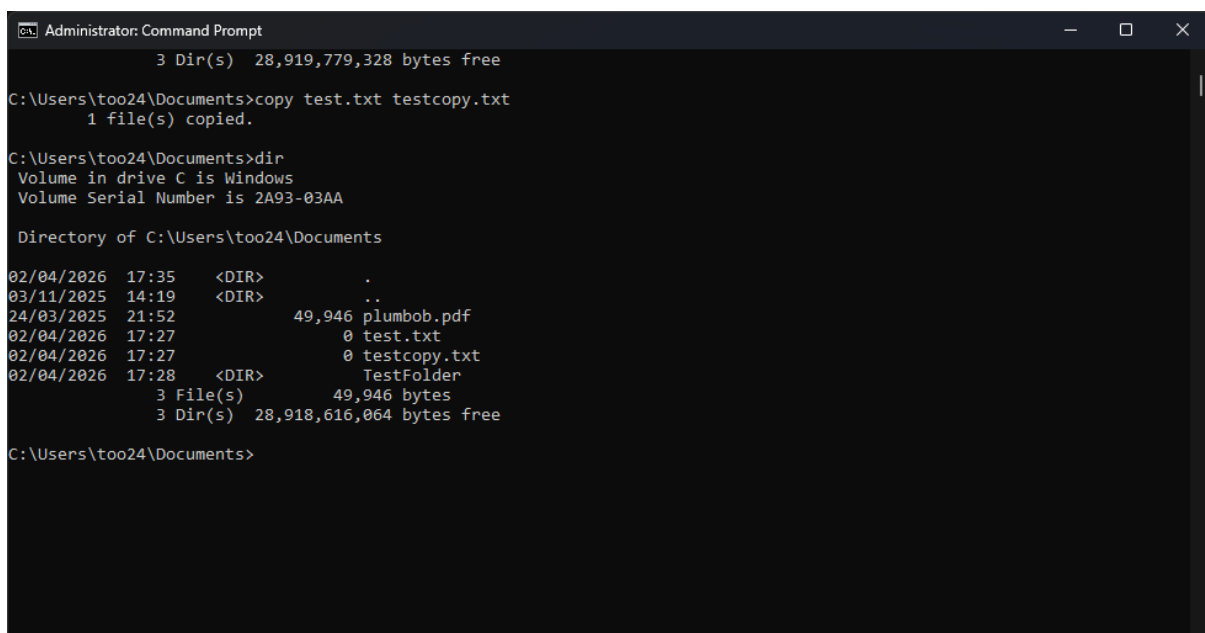
02/04/2026  16:44    <DIR>          .
03/11/2025  14:19    <DIR>          ..
24/03/2025  21:52             49,946 plumbob.pdf
02/04/2026  16:44    <DIR>          TestFolder
               1 File(s)              49,946 bytes
               3 Dir(s)  28,931,026,944 bytes free

C:\Users\too24\Documents>
```

Figure 11 Output of the mkdir command in Windows CLI

2.6 copy

The copy command is used in Windows CLI to duplicate a file from one location to another. For example, “copy test.txt testcopy.txt” creates a copy of the original file. This is useful for file management and simple backups. [2], [7]



```
Administrator: Command Prompt
               3 Dir(s)  28,919,779,328 bytes free
C:\Users\too24\Documents>copy test.txt testcopy.txt
1 file(s) copied.
C:\Users\too24\Documents>dir
Volume in drive C is Windows
Volume Serial Number is 2A93-03AA

Directory of C:\Users\too24\Documents

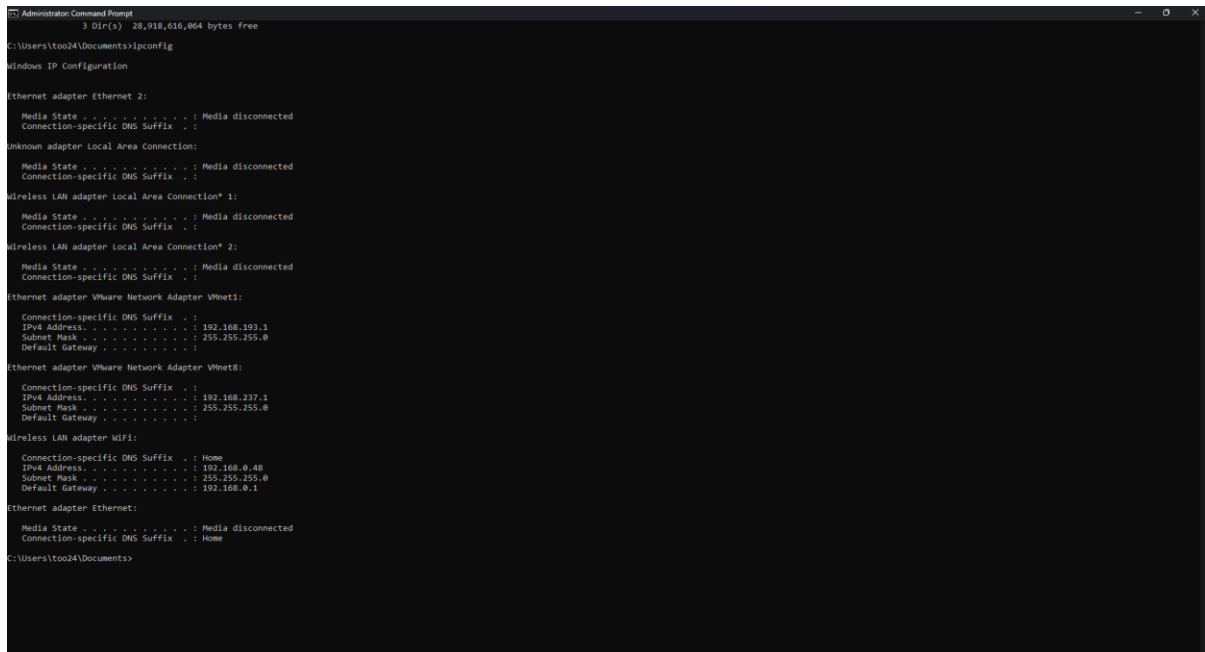
02/04/2026  17:35    <DIR>          .
03/11/2025  14:19    <DIR>          ..
24/03/2025  21:52             49,946 plumbob.pdf
02/04/2026  17:27              0 test.txt
02/04/2026  17:27              0 testcopy.txt
02/04/2026  17:28    <DIR>          TestFolder
               3 File(s)              49,946 bytes
               3 Dir(s)  28,918,616,064 bytes free

C:\Users\too24\Documents>
```

Figure 12 Output of the copy command in Windows CLI

2.7 ipconfig

The ipconfig command is used in Windows CLI to display the IP configuration of the computer. It can show details such as the IPv4 address, subnet mask and default gateway. This is useful for checking network settings and troubleshooting connectivity problems. [2], [10]



```
Administrator: Command Prompt
C:\Users\too24\Documents>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 2:

   Media State . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : 
Unknown adapter Local Area Connection:

   Media State . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : 
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  . : 
Ethernet adapter VMware Network Adapter VMnet1:

   Connection-specific DNS Suffix  . : 
   IPv4 Address. . . . . : 192.168.193.1
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : 
Ethernet adapter VMware Network Adapter VMnet8:

   Connection-specific DNS Suffix  . : 
   IPv4 Address. . . . . : 192.168.237.1
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : 
Wireless LAN adapter WiFi:

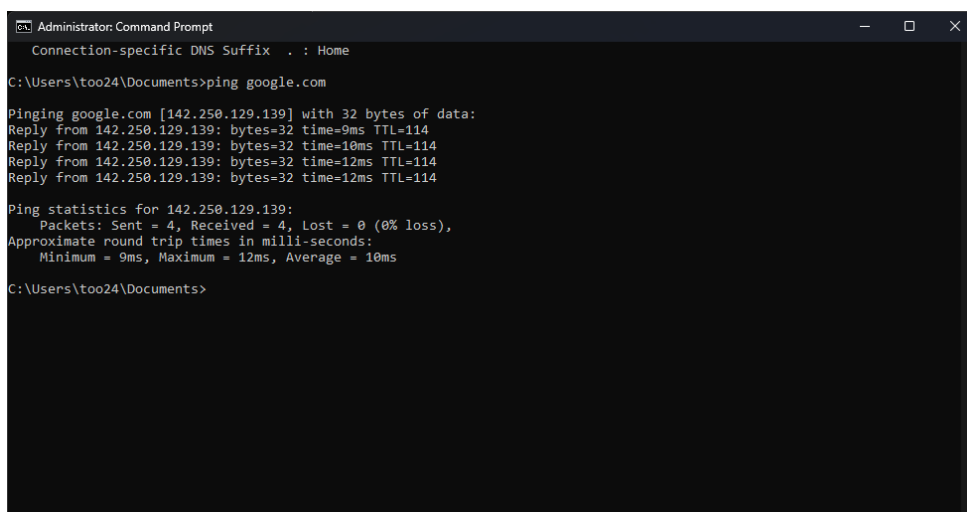
   Connection-specific DNS Suffix  . : Home
   IPv4 Address. . . . . : 192.168.0.48
   Subnet Mask . . . . . : 255.255.255.0
   Default Gateway . . . . . : 192.168.0.1
Ethernet adapter Ethernet:

   Media State . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : Home
C:\Users\too24\Documents>
```

Figure 13 Output of the ipconfig command in Windows CLI

2.8 Ping

The ping command is used in Windows CLI to test connectivity between the computer and another device or website. It sends packets and shows whether replies are received, along with response times. This is useful for checking whether a network connection is working correctly. [2], [7]



```
Administrator: Command Prompt
C:\Users\too24\Documents>ping google.com

Pinging google.com [142.250.129.139] with 32 bytes of data:
Reply from 142.250.129.139: bytes=32 time=9ms TTL=114
Reply from 142.250.129.139: bytes=32 time=10ms TTL=114
Reply from 142.250.129.139: bytes=32 time=12ms TTL=114
Reply from 142.250.129.139: bytes=32 time=12ms TTL=114

Ping statistics for 142.250.129.139:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 9ms, Maximum = 12ms, Average = 10ms
C:\Users\too24\Documents>
```

Figure 14 Output of the ping command in Windows CLI

2.9 net user

The net user command is used in Windows CLI to display information about user accounts on the computer. It can show the names of accounts and support basic account management tasks. This is useful for checking which users exist on the system. [2], [7]

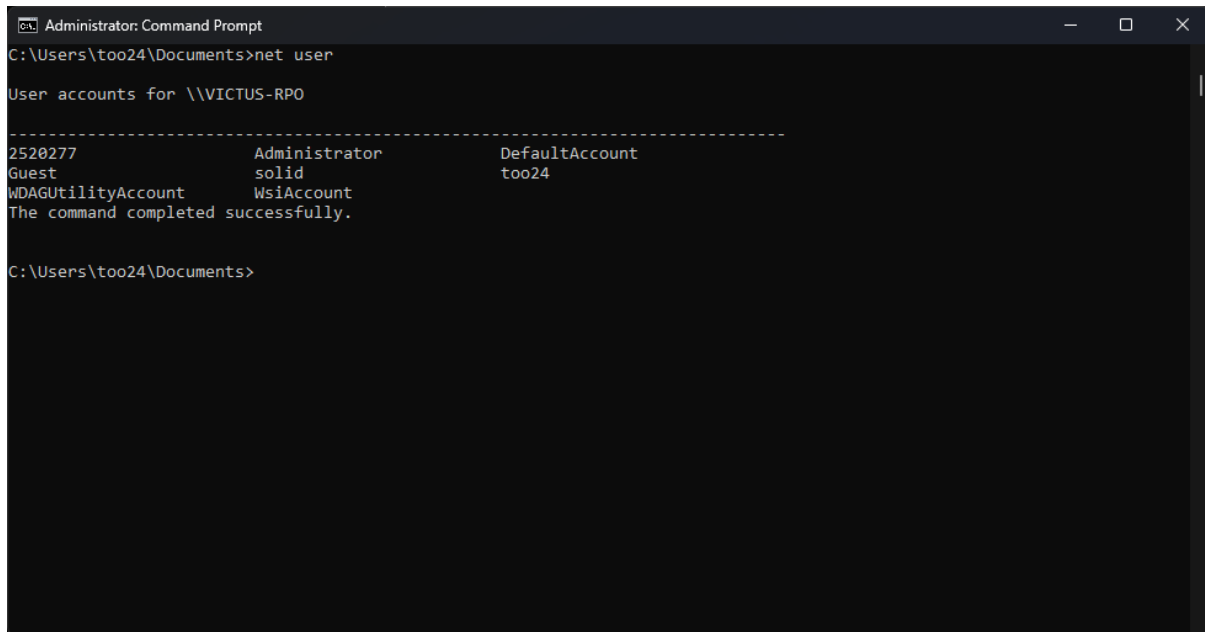


Figure 15 Output of the net user command in Windows CLI

2.10 schtasks

The `schtasks` command is used in Windows CLI to display and manage scheduled tasks on the system. It can show tasks set to run automatically at specific times or under certain conditions. This is useful for checking automation and scheduled maintenance tasks. [2], [11]

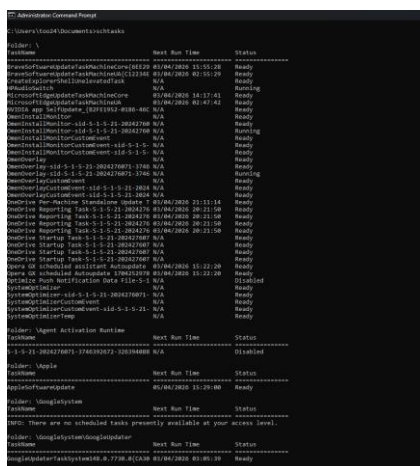


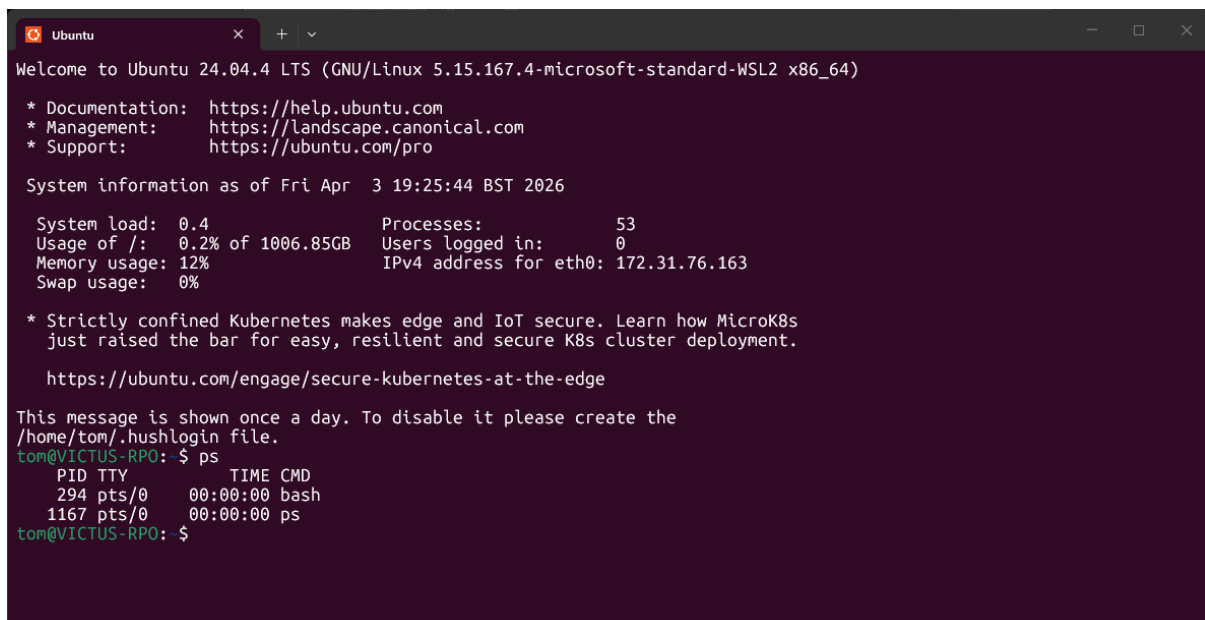
Figure 16 Output of the schtasks command in Windows CLI

Task 3: Linux CLI Commands

This section demonstrates the use of Linux CLI commands. These commands are used for system administration and can help with process management, file and directory management, network testing and monitoring, user management, and task scheduling. This section will contain screenshots of these commands being run and showing their results. [3], [12]

3.1 ps

The ps command in Linux CLI is used to display information about running processes. It can show details such as the process ID, terminal and command name. This is useful for checking which processes are active on the system. [3], [14]

A screenshot of a Linux terminal window titled 'Ubuntu'. The terminal shows the output of the 'ps' command. At the top, it says 'Welcome to Ubuntu 24.04.4 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)'. Below that are links for documentation, management, and support. Then, system information is displayed as of 'Fri Apr 3 19:25:44 BST 2026', including system load, usage of /, memory usage, swap usage, processes, users logged in, and IPv4 address for eth0. A message about MicroK8s is also shown. Finally, the output of the 'ps' command is displayed, showing a table with columns PID, TTY, TIME, and CMD. The table lists two processes: 'bash' with PID 294 and 'ps' with PID 1167.

```
Ubuntu
Welcome to Ubuntu 24.04.4 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Apr 3 19:25:44 BST 2026

System load: 0.4          Processes:            53
Usage of /:  0.2% of 1006.85GB Users logged in:       0
Memory usage: 12%        IPv4 address for eth0: 172.31.76.163
Swap usage:  0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/home/tom/.hushlogin file.
tom@VICTUS-RPO:~$ ps
  PID TTY          TIME CMD
   294 pts/0    00:00:00 bash
  1167 pts/0    00:00:00 ps
tom@VICTUS-RPO:~$
```

Figure 17 Output of the ps command in Linux CLI

3.2 kill

The kill command is used in Linux CLI to stop a running process on the system. It is normally used with the process ID (PID) of the process that needs to be ended. This is useful for stopping unresponsive processes or tasks using too many resources. [3], [14]

```

tom@VICTUS-RPO:~$ sleep 60 &
[2] 1172
tom@VICTUS-RPO:~$ ps
  PID TTY          TIME CMD
  294 pts/0    00:00:00 bash
 1170 pts/0    00:00:00 sleep
 1172 pts/0    00:00:00 sleep
 1173 pts/0    00:00:00 ps
tom@VICTUS-RPO:~$ kill 1172
tom@VICTUS-RPO:~$ ps
  PID TTY          TIME CMD
  294 pts/0    00:00:00 bash
 1170 pts/0    00:00:00 sleep
 1193 pts/0    00:00:00 ps
[2]+  Terminated                  sleep 60
tom@VICTUS-RPO:~$

```

Figure 18 Output of the kill command in Linux CLI

3.3 ls

The ls command is used in Linux CLI to display the files and directories in the current location. It shows the contents of the directory so the user can check what files and folders are stored there. This is useful for file and directory management without using a graphical interface. [3], [13]

```

tom@VICTUS-RPO:~$ ls
file.txt
[1]+  Done                  sleep 60
tom@VICTUS-RPO:~$ █

```

Figure 19 Output of the ls command in Linux CLI

3.4 cd

The cd command is used in Linux CLI to change the current working directory. It allows the user to move between folders so commands can be run in the correct location. This is useful for navigation and file management in the terminal. [3], [13]

```
tom@VICTUS-RPO: ~  
Welcome to Ubuntu 24.04.4 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/pro  
  
System information as of Sat Apr  4 15:28:29 BST 2026  
  
System load:  0.08      Processes:            33  
Usage of /:   0.2% of 1006.85GB  Users logged in:     1  
Memory usage: 12%      IPv4 address for eth0: 172.31.76.163  
Swap usage:   0%  
  
* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s  
just raised the bar for easy, resilient and secure K8s cluster deployment.  
  
https://ubuntu.com/engage/secure-kubernetes-at-the-edge  
  
This message is shown once a day. To disable it please create the  
/home/tom/.hushlogin file.  
tom@VICTUS-RPO:~$ cd ~  
tom@VICTUS-RPO:~$ pwd  
/home/tom  
tom@VICTUS-RPO:~$ |
```

Figure 20 Output of the cd command in Linux CLI

3.5 mkdir

The mkdir command is used in Linux to create a new folder or directory. This helps organise files and data into appropriate places. It is useful for quickly creating new directories without using a graphical interface. [3], [13]

```
This message is shown once a day. To disable it please create the  
/home/tom/.hushlogin file.  
tom@VICTUS-RPO:~$ cd ~  
tom@VICTUS-RPO:~$ pwd  
/home/tom  
tom@VICTUS-RPO:~$ cd ~  
tom@VICTUS-RPO:~$ mkdir TestFolder  
tom@VICTUS-RPO:~$ ls  
TestFolder  file.txt  
tom@VICTUS-RPO:~$
```

Figure 21 Output of the mkdir command in Linux CLI

3.6 cp

The cp command is used in Linux to copy a file from one location to another. It allows the user to duplicate a file without changing the original. This is useful for file management and creating simple backups. [3], [13]

```
tom@VICTUS-RPO:~$ cd ~  
tom@VICTUS-RPO:~$ echo "This is a test file" > testfile.txt  
tom@VICTUS-RPO:~$ cp testfile.txt testfilecopy.txt  
tom@VICTUS-RPO:~$ ls  
TestFolder  file.txt  testfile.txt  testfilecopy.txt  
tom@VICTUS-RPO:~$
```

Figure 22 Output of the cp command in Linux CLI

3.7 ip a

The ip a command is used in Linux CLI to display information about the network interfaces on the system. It can show details such as IP addresses and the status of network adapters. This is useful for checking network configuration and troubleshooting connectivity problems. [3], [14]

```
tom@VICTUS-RPO:~$ ip a
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
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:d0:b0:95 brd ff:ff:ff:ff:ff:ff
    inet 172.31.76.163/20 brd 172.31.79.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fed0:b095/64 scope link
        valid_lft forever preferred_lft forever
tom@VICTUS-RPO:~$ |
```

Figure 23 Output of the ip a command in Linux CLI

3.8 ping

The ping command is used in Linux CLI to test connectivity between the computer and another device or website on a network. It sends packets to the specified destination and shows whether replies are received. This is useful for checking whether a network connection is working correctly. [3], [15]

```
tom@VICTUS-RPO:~$ ping -c 4 google.com
PING google.com (192.178.223.102) 56(84) bytes of data.
64 bytes from yulhrs-in-f102.1e100.net (192.178.223.102): icmp_seq=1 ttl=108 time=11.4 ms
64 bytes from yulhrs-in-f102.1e100.net (192.178.223.102): icmp_seq=2 ttl=108 time=73.3 ms
64 bytes from yulhrs-in-f102.1e100.net (192.178.223.102): icmp_seq=3 ttl=108 time=62.9 ms
64 bytes from yulhrs-in-f102.1e100.net (192.178.223.102): icmp_seq=4 ttl=108 time=85.4 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 11.411/58.252/85.409/28.189 ms
tom@VICTUS-RPO:~$ |
```

Figure 24 Output of the ping command in Linux CLI

3.9 who

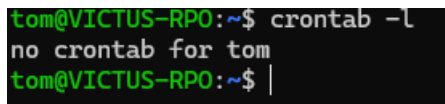
The who command is used in Linux CLI to display information about users who are currently logged into the system. It can show details such as the username, terminal and login time. This is useful for checking active users and sessions. [3], [16]

```
tom@VICTUS-RPO:~$ who
tom      pts/1      2026-04-03 19:25
tom@VICTUS-RPO:~$ |
```

Figure 25 Output of the who command in Linux CLI

3.10 crontab -l

The `crontab -l` command is used in Linux CLI to display the scheduled tasks that have been set for the current user. It shows any cron jobs configured to run automatically at specific times or intervals. This is useful for checking task scheduling and reviewing existing automation. [3], [17]



```
tom@VICTUS-RPO:~$ crontab -l
no crontab for tom
tom@VICTUS-RPO:~$ |
```

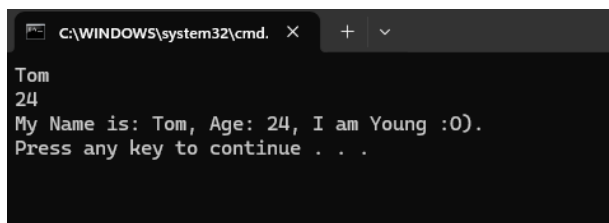
Figure 26 Output of the `crontab -l` command in Linux CLI

Task 4: Windows OS Scripts

This section demonstrates the use of Windows scripting through Batch commands and small script files. These scripts are used to automate simple tasks showing how they can be used for administrative actions without entering every command manually. Examples in this section will demonstrate features such as variables, user input, conditions and loops. Screenshots of the code and output are included below. [4]

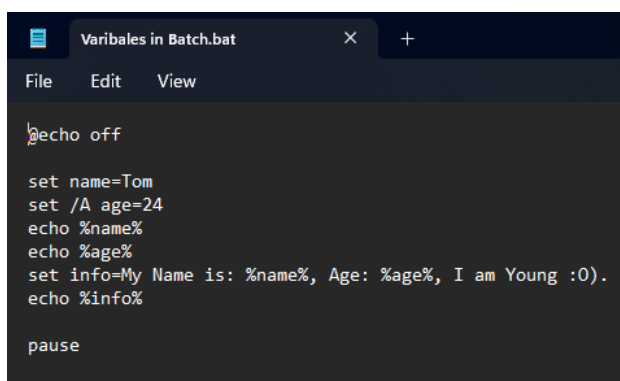
4.1 Batch Script Using Variables

This Batch script demonstrates the use of variables in Windows scripting. The `set` command is used to store text, while `set /A` is used to store a number. The script then uses `echo` to display the values on screen, showing how Batch scripts can store and output simple information. [4]



```
C:\WINDOWS\system32\cmd. X + v
Tom
24
My Name is: Tom, Age: 24, I am Young :0).
Press any key to continue . . .
```

Figure 27 Output of the Batch variables script



```
Varibales in Batch.bat X +
File Edit View
echo off

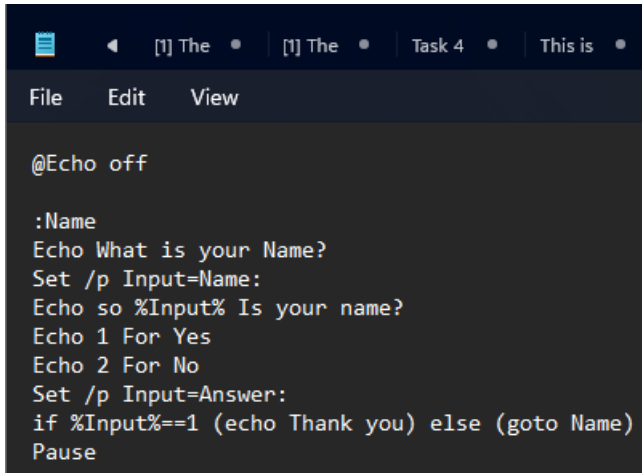
set name=Tom
set /A age=24
echo %name%
echo %age%
set info=My Name is: %name%, Age: %age%, I am Young :0).
echo %info%

pause
```

Figure 28 Batch script showing the use of variables

4.2 Batch Script Using User Input, IF ELSE and Goto

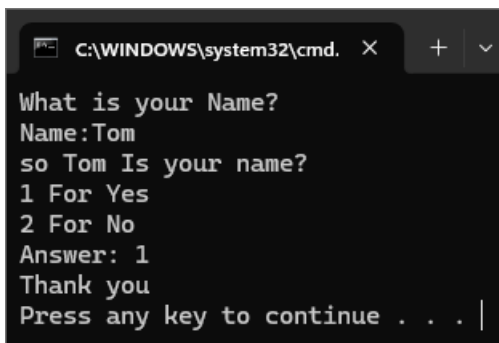
This Batch script demonstrates user input, decision making and branching in Windows scripting. The set /p command is used to collect input from the user, while the if else structure checks the response and the goto command moves the script to another labelled section. This shows how Batch scripts can react to user input and follow different paths. [4]

A screenshot of a Notepad++ window with a dark theme. The window has four tabs: '[1] The', '[1] The', 'Task 4', and 'This is'. The menu bar shows 'File', 'Edit', and 'View'. The script content is as follows:

```
@Echo off

:Name
Echo What is your Name?
Set /p Input=Name:
Echo so %Input% Is your name?
Echo 1 For Yes
Echo 2 For No
Set /p Input=Answer:
if %Input%==1 (echo Thank you) else (goto Name)
Pause
```

Figure 29 Batch script showing user input, IF ELSE and goto

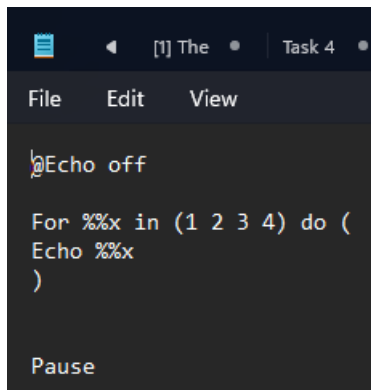
A screenshot of a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.'. The output of the batch script is displayed as follows:

```
What is your Name?
Name:Tom
so Tom Is your name?
1 For Yes
2 For No
Answer: 1
Thank you
Press any key to continue . . . |
```

Figure 30 Output of the Batch user input script

4.3 Batch Script Using a For Loop

This Batch script demonstrates the use of a for loop in Windows scripting. A for loop repeats a command a set number of times, allowing the script to carry out the same action automatically. This is useful because it shows how Batch scripts can automate repetitive tasks instead of requiring the user to enter the same command multiple times. [4]

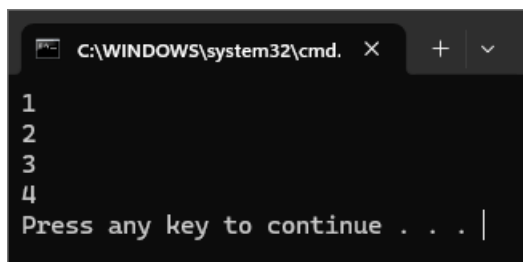
A screenshot of a text editor window titled '[1] The' and 'Task 4'. The editor has a menu bar with 'File', 'Edit', and 'View'. The script content is as follows:

```
Echo off

For %%x in (1 2 3 4) do (
Echo %%x
)

Pause
```

Figure 31 Batch script showing a for loop

A screenshot of a Windows command prompt window titled 'C:\WINDOWS\system32\cmd.'. The output of the batch script is shown as follows:

```
1
2
3
4
Press any key to continue . . . |
```

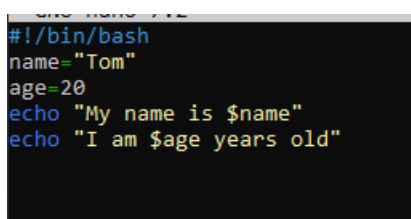
Figure 32 Output of the Batch for loop script

Task 5: Linux OS Scripts

This section demonstrates the use of Linux scripting through Bash commands and small script files. These scripts can be used to automate simple tasks and show how command line actions can be carried out without entering every command manually. The examples in this section will demonstrate features such as variables, user input, conditions and loops. [5], [12]

5.1 Bash Script Using Variables

This Bash script demonstrates the use of variables in Linux scripting. Variables are used to store values, and the echo command is used to display them on screen. This shows how Bash scripts can store and output simple information. [5], [12]

A screenshot of a Linux terminal window showing a Bash script. The script content is as follows:

```
#!/bin/bash
name="Tom"
age=20
echo "My name is $name"
echo "I am $age years old"
```

Figure 33 Bash script showing the use of variables

```

This message is shown once a day. To disable it please create the
/home/tom/.hushlogin file.
tom@VICTUS-RPO:~$ nano variables.sh
tom@VICTUS-RPO:~$ chmod +x variables.sh
tom@VICTUS-RPO:~$ ./variables.sh
My name is Tom
I am 20 years old
tom@VICTUS-RPO:~$

```

Figure 34 Output of the Bash variables script

5.2 Bash Script Using User Input and IF

This Bash script demonstrates user input and decision making in Linux scripting. The read command is used to collect input from the user, while the if statement checks the response and decides what message to display. This shows how Bash scripts can react to user input and follow different paths. [5], [12]

```

GNU nano 7.2
#!/bin/bash
echo "Enter your age:"
read age

if [ "$age" -ge 18 ]; then
    echo "You are an adult."
else
    echo "You are under 18."
fi

```

Figure 35 Bash script showing user input and an IF statement

```

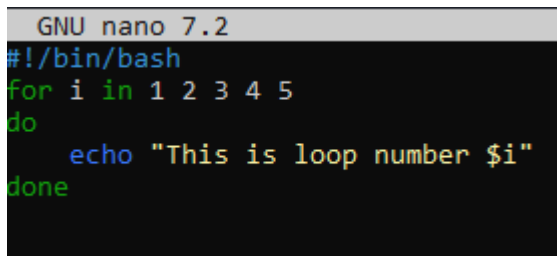
tom@VICTUS-RPO: ~
tom@VICTUS-RPO:~$ nano input_if.sh
tom@VICTUS-RPO:~$ chmod +x input_if.sh
tom@VICTUS-RPO:~$ ./input_if.sh
Enter your age:
20
You are an adult.
tom@VICTUS-RPO:~$

```

Figure 36 Output of the Bash user input script

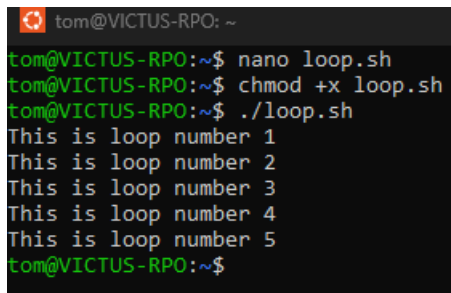
5.3 Bash Script Using a Loop

This Bash script demonstrates the use of a loop in Linux scripting. A loop repeats a command a set number of times, allowing the script to carry out the same action automatically. This is useful because it shows how Bash scripts can automate repetitive tasks. [5], [12]



```
GNU nano 7.2
#!/bin/bash
for i in 1 2 3 4 5
do
    echo "This is loop number $i"
done
```

Figure 37 Bash script showing a loop



```
tom@VICTUS-RPO: ~
tom@VICTUS-RPO:~$ nano loop.sh
tom@VICTUS-RPO:~$ chmod +x loop.sh
tom@VICTUS-RPO:~$ ./loop.sh
This is loop number 1
This is loop number 2
This is loop number 3
This is loop number 4
This is loop number 5
tom@VICTUS-RPO:~$
```

Figure 38 Output of the Bash loop script

Conclusion

In conclusion, this report has examined the key functions of general operating systems and has shown how these functions support the stable and secure running of modern computer systems. Functions such as memory management, process management, file management, device management, performance monitoring and security all help the operating system manage hardware, software and user activity.

This report has also demonstrated the practical use of Windows and Linux CLI commands, as well as Batch and Bash scripting. These examples show that command-line tools and scripts can be used for process management, file and directory management, networking, user management, task scheduling and simple automation. Overall, the examples in this report show that command-line tools and scripts remain important for operating system management and automation.

References

- [1] J. Williams, "OS 2 - Lecture - OS Functions," ACCA4022 Computer Architecture and Operating Systems, PowerPoint slides, 2026.
- [2] J. Williams, "OS 3 - Lecture - Sys Mon - File sys & CMD," ACCA4022 Computer Architecture and Operating Systems, PowerPoint slides, 2026.

[3] J. Williams, "OS 4 - Lecture - Linux CLI," ACCA4022 Computer Architecture and Operating Systems, PowerPoint slides, 2026.

[4] J. Williams, "OS 5 - Lecture - Intro to Bat Scripting," ACCA4022 Computer Architecture and Operating Systems, PowerPoint slides, 2026.

[5] J. Williams, "OS 6 - Lecture - Intro. to Bash Scripting," ACCA4022 Computer Architecture and Operating Systems, PowerPoint slides, 2026.

[6] Microsoft, "System Configuration Tools in Windows," Microsoft Support, 2026. Accessed: Apr. 6, 2026.

[7] Microsoft, "Windows Commands," Microsoft Learn, 2026. Accessed: Apr. 6, 2026.

[8] Microsoft, "tasklist," Microsoft Learn, 2023. Accessed: Apr. 6, 2026.

[9] Microsoft, "taskkill," Microsoft Learn, 2024. Accessed: Apr. 6, 2026.

[10] Microsoft, "ipconfig," Microsoft Learn, 2023. Accessed: Apr. 6, 2026.

[11] Microsoft, "schtasks commands," Microsoft Learn, 2023. Accessed: Apr. 6, 2026.

[12] GNU Project, "Bash Reference Manual," GNU, 2025. Accessed: Apr. 7, 2026.

[13] GNU Project, "GNU Coreutils 9.10," GNU, 2025. Accessed: Apr. 7, 2026.

[14] M. Kerrisk, "ip(8) - Linux manual page," man7.org. Accessed: Apr. 7, 2026.

[15] M. Kerrisk, “ping(8) - Linux manual page,” man7.org. Accessed: Apr. 7, 2026.

[16] M. Kerrisk, “who(1) - Linux manual page,” man7.org. Accessed: Apr. 7, 2026.

[17] M. Kerrisk, “crontab(1) - Linux manual page,” man7.org. Accessed: Apr. 7, 2026.

[18] J. Williams, “OS 1 - Tutorial - Intro History & Types,” ACCA4022 Computer Architecture and Operating Systems, PowerPoint slides, 2026.

[19] OpenStax, “What Is an Operating System?,” Introduction to Computer Science, 2024. [Online]. Available: <https://openstax.org/books/introduction-computer-science/pages/6-1-what-is-an-operating-system>

[20] TechTarget, “What is a Network Operating System?,” 2025. [Online]. Available: <https://www.techtarget.com/searchnetworking/definition/network-operating-system>

Appendices

Appendix 1 – OS Types Tutorial Activity

Operating systems can be grouped into different types depending on the purpose they serve. One of the most common types is the general-purpose operating system, which is designed for everyday computing tasks on devices such as desktop computers and laptops. Examples include Windows, Linux and macOS. These operating systems support a wide range of applications and are used by both home users and organisations. Mobile operating systems are designed for smartphones and tablets, with common examples including Android and Apple iOS. They are optimised for touchscreens, battery efficiency and mobile apps. [18], [19]

Another type is the network operating system, which allows users on a network to share resources such as files, email and media. Examples include Unix, FreeBSD, Ubuntu Server and Windows Server systems. Internetwork operating systems are used on devices such as routers and switches, for example Cisco IOS and Junos OS. These are designed for configuring and managing network hardware and are often controlled through command-line interfaces. [18], [20]

There are also more specialised operating systems. Real-time operating systems are used where fast and reliable responses are essential, such as in ATMs, aircraft systems

and industrial control systems. Embedded operating systems are designed for specific devices such as smart meters, drones, retail point-of-sale (POS) systems and smartwatches. Similar to these are IoT edge operating systems, which support devices connected at the edge of networks. Finally, AI and quantum operating systems are emerging types that aim to support adaptive AI-based interaction or advanced quantum processing capabilities. This shows that although operating systems share core functions, they can be developed for very different practical uses. [18]

Appendix 2 – Lecture Activities / Activity Log

Lecture 1 – Introduction, History and Types of Operating Systems

The lecture activity involved documenting what an operating system is and what an operating system does. I reviewed how an operating system acts as system software that manages the interaction between the user, software and hardware. This supported the introduction of the report and the OS Types tutorial activity.

Lecture 2 – Operating System Functions

The lecture activity involved researching and documenting common operating system functions. I reviewed key functions such as process management, memory management, file and directory management, device management, networking, security, and error handling. This supported Task 1 of the report.

Lecture 3 – System Monitoring, File Systems and Windows CMD

The lecture activity involved documenting system monitoring and management tools covered during the lecture. I reviewed Windows tools such as Task Manager, Resource Monitor, Device Manager, File Explorer, Windows Security and command-line tools. This supported the screenshots and explanations used in Task 1 and Task 2.

Lecture 4 – Linux CLI

The lecture activity involved making notes on Linux and the Linux command line. I practised Linux CLI commands in the Ubuntu terminal, including commands for process management, file and directory management, network testing, user information and task scheduling. This supported Task 3 of the report.

Lecture 5 – Introduction to Batch Scripting

The lecture activity involved documenting the key syntax areas of the Batch scripting language. I created and tested Batch scripts using variables, user input, if else, goto, and for loops. This supported Task 4 of the report.

Lecture 6 – Introduction to Bash Scripting

The lecture activity involved documenting the key syntax areas of Bash scripting. I created and tested Bash scripts using variables, user input, if statements and loops. This supported Task 5 of the report.

Appendix 3 – Online Course Evidence

The screenshot shows the LabEx user dashboard for Thomas O. The page has a light blue header with the LabEx logo, navigation links (LEARN, REVIEWS, PRICING, TEAMS), and an Upgrade button. Below the header, a welcome message "Welcome to LabEx, Thomas O!" is displayed. The main content area is divided into two sections. On the left, the "Next Milestone" section shows two progress bars: "Earn 50 skills (16/50)" and "Complete 5 streaks (1/5)". Below these are seven circular icons representing the days of the week (Sun, Mon, Tue, Wed, Thu, Fri, Sat), with Tuesday highlighted in green. On the right, a list of users is shown, each with a profile picture, a name, and a skill count. The users are: Tillmann F. (16 Skills), brandon (16 Skills), Thomas O (16 Skills, highlighted in yellow), tho n. (15 Skills), and Furkan \$. (15 Skills). The user Thomas O has a note that says "Earn a new skill to surpass 7050 learners!".

Continue Learning

The screenshot shows the "Continue Learning" section. It features a course card for "Quick Start with Linux" with a "COURSE" label and a "Quick Start" button. To the right of the course card is a "Back to Running Lab" button with a blue robot icon.

Figure 39 LabEx Quick Start with Linux course progress page

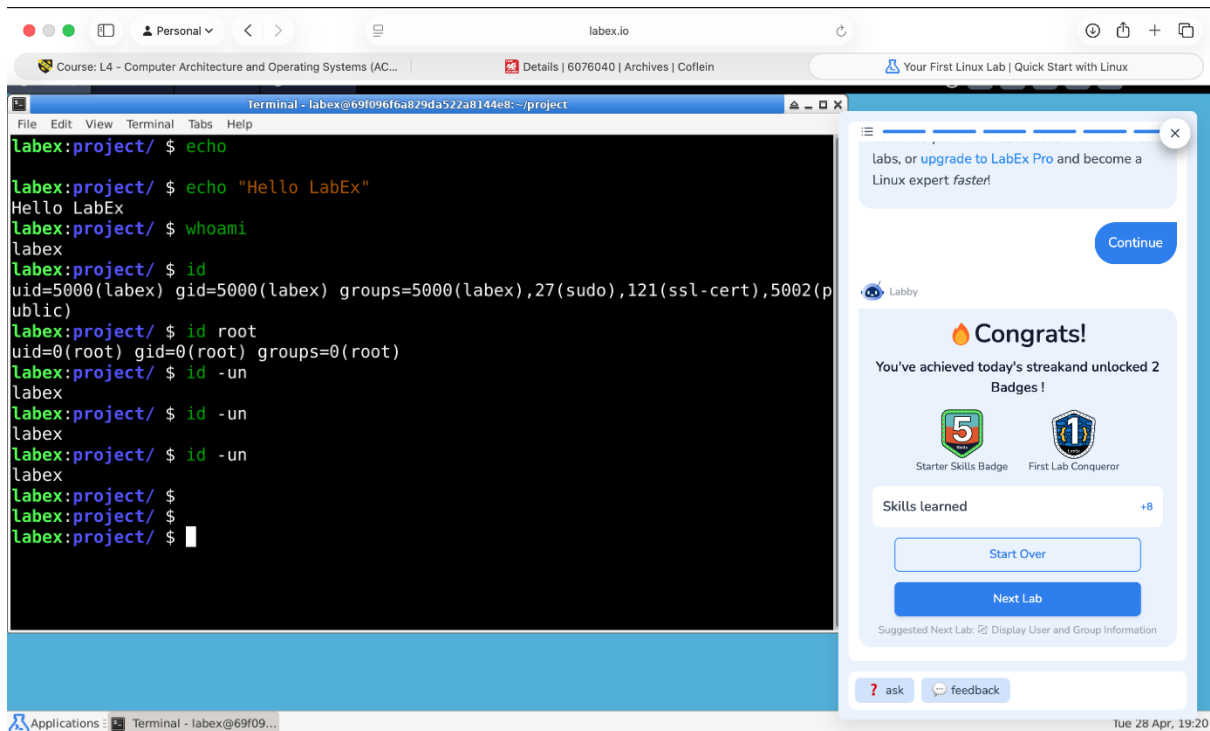


Figure 40 LabEx Linux terminal activity and completed lab badge

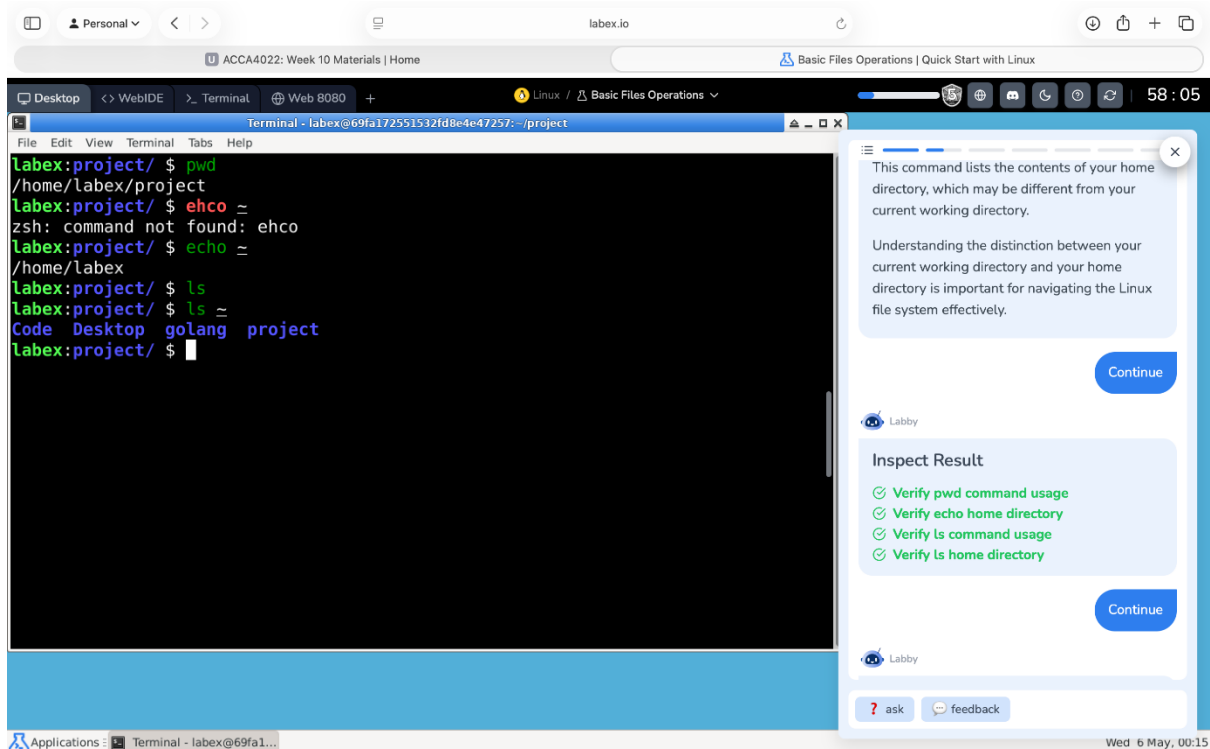


Figure 41 LabEx Basic Files Operations command verification

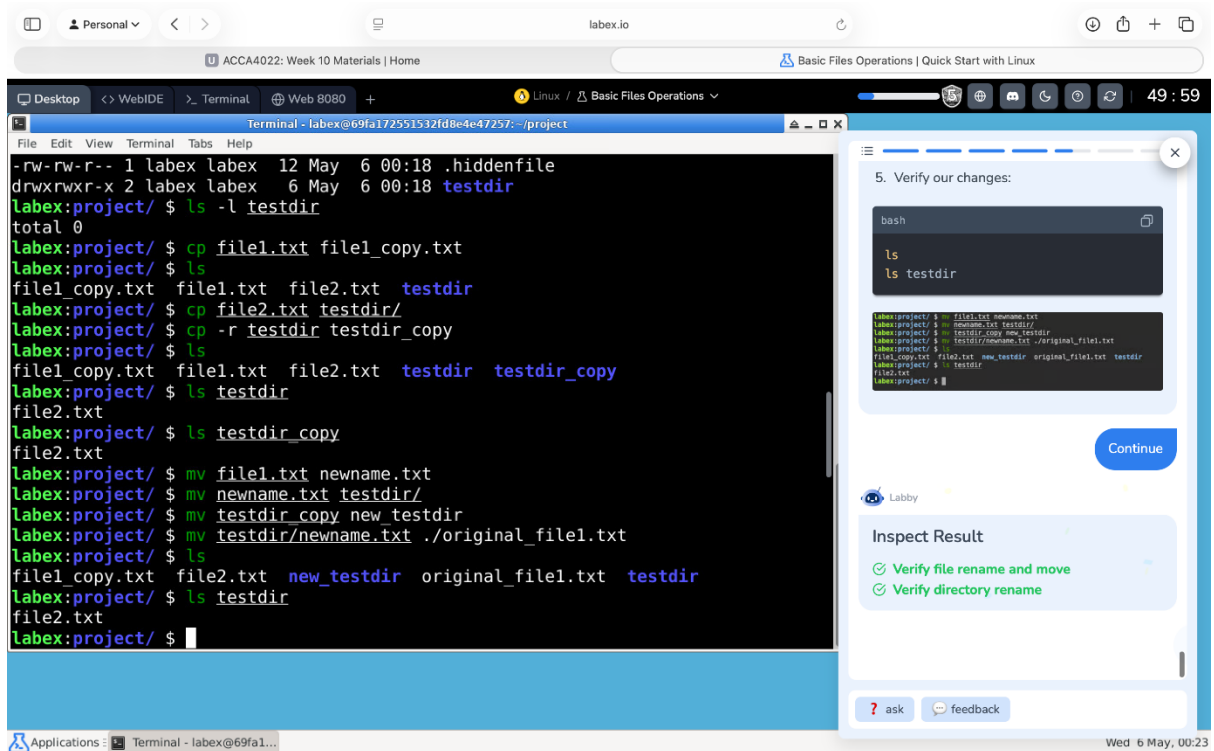


Figure 42 LabEx file management command activity

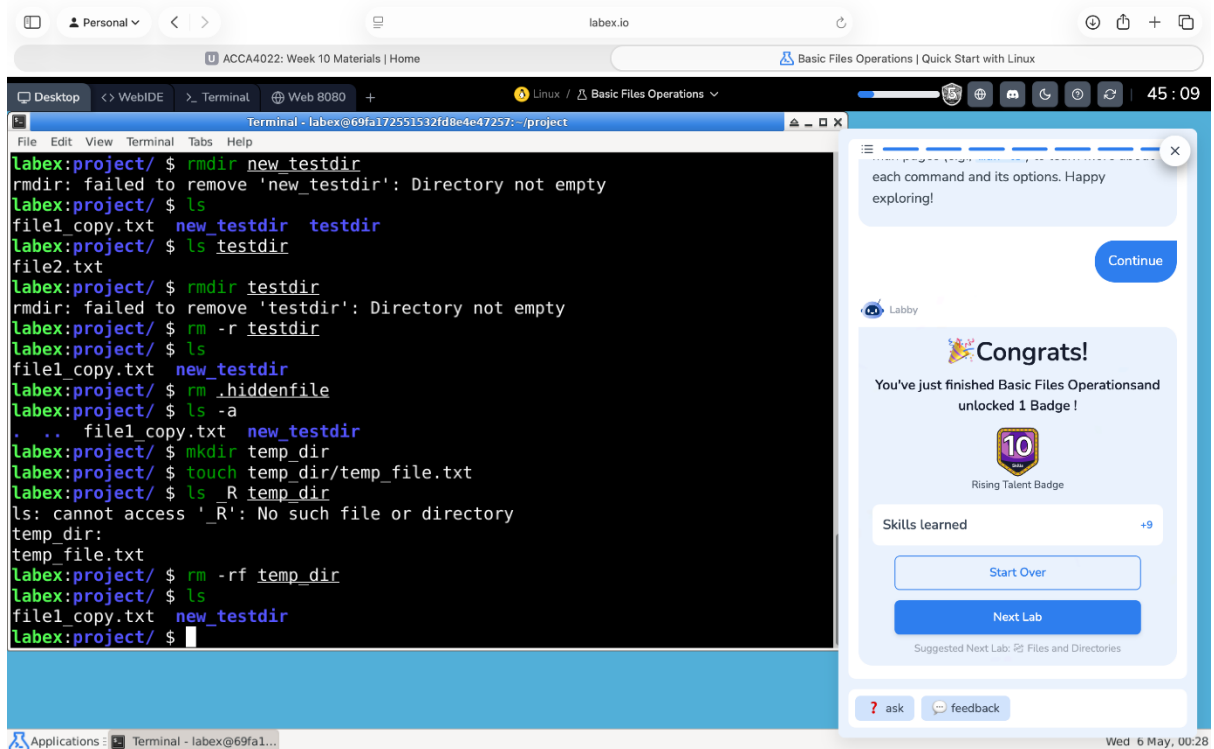


Figure 43 LabEx completed Basic Files Operations badge

