

OS Assignment

by Dan Field

Submission date: 11-Apr-2025 11:29AM (UTC+0100)

Submission ID: 255474604

File name: 61016_Dan_Field_OS_Assignment_644541_918844.docx (2.49M)

Word count: 1676

Character count: 8252

Assignment 1

Daniel Field

2430604

Computer Architecture and Operating Systems.

Assignment 1	1
Introduction	2
Task 1	2
Key functions of general Operating Systems	2
Task 2	4
Use of Windows OS CLI commands.	4
Task 3	8
Use of Linux OS CLI commands.	8
Task 4	10
Use of Windows OS scripts	10
Task 5	13
Use of Linux OS scripts.....	13
Conclusion.....	15
OS tutorial activity 1 Different types of OS's	15
Evidence of an online course:	16

Introduction

In this document I will be covering the key features of the Operating system as well as providing documentation of different commands used in Cmd & Linux as well as the corresponding Batch & Bash scripts that run with each with images of the input and the output of each command that is used.

Task 1

Key functions of general Operating Systems

The main function of an OS acts as a communication interface between the user and computer hardware. Its purpose is to provide a platform on which a user can execute programs conveniently and efficiently.[1]

There are multiple different key features that the OS provides and uses to protect and maintain computers Such as:

System performance monitoring:

With this, the system will monitor the health and Performance of the computer and will then provide visual diagnostic tools to the user that will allow them to see what is holding their computer back

Device management:

If it wasn't for this part of the OS, we would not be able to use mouses, keyboards, mics etc..., It is because of the OS that the computer can create a smooth

communication between the peripherals making it easier for the users to interact with the computer

Process Management:

If it wasn't for the process management capabilities of the OS, we wouldn't be able to run multiple different programs at the same time making sure each process doesn't interfere with one another, if this didn't exist computer would be crashing every second as there are multiple process that need to run to help the computer work

Examples of key features:

- **Task Manager**

It controls all the computer resources to show a more in-depth overview of processes of the system, I went into the Command Prompt and wrote tasklist. By doing this it allows me to have an overview of what is being used on the system and how many processors are being used on each application.

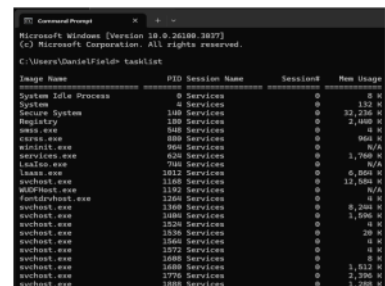
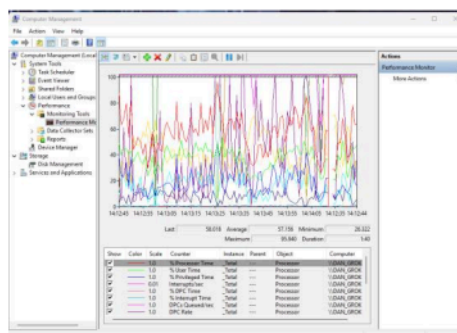


Image Name	PID	Session Name	Session	Mem Usage
System Idle Process	0	Services	0	0 K
System	4	Services	0	132 K
smss.exe	140	Services	0	22,236 K
svchost.exe	148	Services	0	2,400 K
csrss.exe	880	Services	0	4 K
wininit.exe	964	Services	0	96 K
services.exe	628	Services	0	1,700 K
lsass.exe	704	Services	0	1,700 K
svchost.exe	1168	Services	0	12,584 K
cmd.exe	1192	Services	0	0 K
cmd.exe	1264	Services	0	0 K
svchost.exe	1360	Services	0	0 K
svchost.exe	1440	Services	0	0 K
svchost.exe	1520	Services	0	0 K
svchost.exe	1536	Services	0	0 K
svchost.exe	1560	Services	0	0 K
svchost.exe	1572	Services	0	0 K
svchost.exe	1680	Services	0	1,812 K
svchost.exe	1776	Services	0	2,208 K
svchost.exe	1808	Services	0	2,208 K

- **Processor Management**



For the image on the left, I have used the computer management application which is a tool used to view the overall performance of the processing unit.

- **GUI**



The image to the right shows the desktop screen of a computer known as the GUI (Graphical User Interface), the OS is responsible for managing the GUI as seen on the image to the right.

Task 2

Use of Windows OS CLI commands.

For referencing = [50 Basic Windows Commands with Examples - Active Directory Pro](#)

(1) Tasklist code in Cmd (Input and Output)

The first command I used was the Tasklist command in CMD. This command shows me all the applications and processes within the PC. It shows the session name and the memory. This allows the user to know where most of the memory is being used which allows then to have more control over the amount of memory being used.

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	8 K
System	4	Services	0	132 K
Secure System	148	Services	0	32,236 K
Registry	188	Services	0	2,440 K
smss.exe	548	Services	0	4 K
csrss.exe	888	Services	0	964 K
wininit.exe	964	Services	0	N/A
services.exe	624	Services	0	1,768 K
lsass.exe	704	Services	0	N/A
lsass.exe	1012	Services	0	6,864 K
svchost.exe	1168	Services	0	12,584 K
WUDFHost.exe	1192	Services	0	N/A
Fontdrvhost.exe	1264	Services	0	4 K
svchost.exe	1360	Services	0	8,204 K
svchost.exe	1404	Services	0	1,596 K
svchost.exe	1524	Services	0	4 K
svchost.exe	1536	Services	0	20 K
svchost.exe	1564	Services	0	4 K
svchost.exe	1572	Services	0	4 K
svchost.exe	1608	Services	0	8 K
svchost.exe	1688	Services	0	1,512 K
svchost.exe	1776	Services	0	2,396 K
svchost.exe	1888	Services	0	1,288 K

(2) Perfmon code in Cmd (Input and Output)

[illegible]

```
C:\Users\DanielField>perfmon/rel
C:\Users\DanielField>
```

The screenshot shows the Windows Reliability Monitor interface. At the top, there's a navigation bar with tabs for 'Reliability Monitor', 'Contact Help', 'System and Security', 'Security and Maintenance', and 'Reliability Monitor'. Below the navigation bar, there's a heading 'Review your computer's reliability and problem history'. A paragraph explains that the tool tracks system stability on a scale from 1 to 10, with 10 being the best. It mentions that clicking on specific points in the graph can show specific hardware and software problems. The graph itself shows a blue line representing the reliability score over time, with various colored dots (blue, yellow, red) indicating events. Below the graph, there's a table titled 'Reliability details for 11/12/2012' with columns for 'Event', 'Summary', 'Date', and 'Action'. The table lists three events: a warning about Windows Updates, a critical error about Windows Updates, and a successful update of Windows Updates. To the right of the table, there's a link to 'View technical info...'. At the bottom, there are links for 'View reliability history...' and 'View all problem reports'.

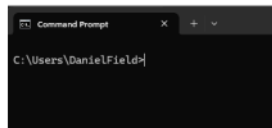
```
C:\Users\DanielField>ping 192.168.1.9

Pinging 192.168.1.9 with 32 bytes of data:
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

(5) CIs code in Cmd (Input and Output)

The CIs Command helps Clear the Command Line Interface for a Better Visual appearance



```
Command Prompt
-r count      Record route for count hops (IPv4-only).
-s count      Timestamp for count hops (IPv4-only).
-j host-list  Loose source route along host-list (IPv4-only).
-k host-list  Strict source route along host-list (IPv4-only).
-w timeout    Timeout in milliseconds to wait for each reply.
-R            Use routing header to test reverse route also (IPv6-only).
            Per RFC 5895 the use of this routing header has been
            deprecated. Some systems may drop echo requests if
            this header is used.
-S srcaddr    Source address to use.
-c compartment Routing compartment identifier.
-p            Ping a Hyper-V Network Virtualization provider address.
-4            Force using IPv4.
-6            Force using IPv6.

C:\Users\DanielField>ping 192.168.1.9

Pinging 192.168.1.9 with 32 bytes of data:
Reply from 192.168.1.9: bytes=32 time=1ms TTL=128
Reply from 192.168.1.9: bytes=32 time=1ms TTL=128
Reply from 192.168.1.9: bytes=32 time=1ms TTL=128
Reply from 192.168.1.9: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\DanielField>cls
```

(6) Ipconfig code in Cmd (Input and Output)

Displays all current TCP/IP network configuration values and refreshes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) settings. Used without parameters, **ipconfig** displays Internet Protocol version 4 (IPv4) and IPv6 addresses, subnet mask, and default gateway for all adapters.

```
Command Prompt
C:\Users\DanielField>ipconfig /all

Windows IP Configuration

Host Name . . . . . DanielField-Laptop
Primary Dns Suffix . . . . .
Node Type . . . . . Hybrid
IP Routing Enabled . . . . . No
WINS Proxy Enabled . . . . . No

Ethernet adapter Ethernet 2:

Media State . . . . . Media disconnected
Connection-specific DNS Suffix . . . . .
Description . . . . . Dell GigE Ethernet
Physical Address . . . . . 9C-EB-66-00-00-00
DHCP Enabled . . . . . Yes
Autonomous Configuration Enabled . . . . . Yes

Wireless LAN adapter Local Area Connection 1:

Media State . . . . . Media disconnected
Connection-specific DNS Suffix . . . . .
Description . . . . . Microsoft Wi-Fi Direct Virtual Adapter
Physical Address . . . . . 8B-46-8B-00-00-00
DHCP Enabled . . . . . No
Autonomous Configuration Enabled . . . . . Yes

Wireless LAN adapter Local Area Connection 2:

Media State . . . . . Media disconnected
Connection-specific DNS Suffix . . . . .
Description . . . . . Microsoft Wi-Fi Direct Virtual Adapter #2
Physical Address . . . . . 8B-46-8B-00-00-00
DHCP Enabled . . . . . No
Autonomous Configuration Enabled . . . . . Yes

Wireless LAN adapter WiFi:

Connection-specific DNS Suffix . . . . .
Description . . . . . Intel(R) Wi-Fi 6 AX200 160MHz
Physical Address . . . . . 28-90-5A-0B-00-00
DHCP Enabled . . . . . Yes
Autonomous Configuration Enabled . . . . . Yes
Link-local IPv6 Address . . . . . fe80::7603:2a7f:80ff:2a7603:1%{InterfaceName}
IPv4 Address . . . . . 192.168.1.100(Preferred)
Subnet Mask . . . . . 255.255.255.0
Lease Obtained . . . . . 22 February 2023 00:00:00
Lease Expires . . . . . 22 February 2023 00:00:00
Default Gateway . . . . . 192.168.1.1
```

(7) Dir code in Cmd (Input and Output)

Dir – Shows and lists all accessible directories within the directory

```
C:\Users\Daniel>dir
Volume in drive C has no label.
Volume Serial Number is 26A0-F397

Directory of C:\Users\Daniel

03/27/2025  11:27 PM    <DIR>          .
01/16/2025  09:54 PM    <DIR>          ..
03/13/2025  12:15 AM    <DIR>          .ftb
02/24/2025  09:07 PM    <DIR>          .mputils
01/16/2025  10:04 PM    <DIR>          Contacts
01/22/2025  08:48 PM    <DIR>          curseforge
03/25/2025  06:58 PM    <DIR>          Desktop
03/10/2025  09:12 PM    <DIR>          Documents
03/10/2025  09:15 PM    <DIR>          Downloads
01/16/2025  10:04 PM    <DIR>          Favorites
01/14/2025  08:13 PM    <DIR>          Intel
01/16/2025  10:04 PM    <DIR>          Links
01/16/2025  10:04 PM    <DIR>          Music
01/14/2025  07:26 PM    <DIR>          OneDrive
02/21/2025  09:25 PM    <DIR>          Pictures
01/16/2025  10:04 PM    <DIR>          Saved Games
01/16/2025  10:04 PM    <DIR>          Searches
01/16/2025  10:04 PM    <DIR>          Videos
               0 File(s)              0 bytes
               18 Dir(s)  235,234,512,896 bytes free

C:\Users\Daniel>|
```

(8) Echo code in Cmd (Input and Output)

Displays a message or text on the screen.

```
C:\Users\Daniel>Echo hello
hello

C:\Users\Daniel>|
```

(9) Exit code in Cmd (Input and Output)

Force closes the Cli

```
C:\Users\Daniel>exit|
```

(10) code in Cmd (Input and Output)

Scans the system directories and repairs any damaged or corrupted files within

```
C:\Windows\System32>sfc
Microsoft (R) Windows (R) Resource Checker Version 6.0
Copyright (C) Microsoft Corporation. All rights reserved.

Scans the integrity of all protected system files and replaces incorrect versions with
correct Microsoft versions.

sfc [/SCANNOW] [/VERIFYONLY] [/SCANFILE=<file>] [/VERIFYFILE=<file>]
[/OFFBOOTDIR=<offline windows directory>] [/OFFWINDIR=<offline boot directory>] [/OFFLOGFILE=<log file path>]]

/SCANNOW      Scans integrity of all protected system files and repairs files with
               problems when possible.
/VERIFYONLY   Scans integrity of all protected system files. No repair operation is
               performed.
/SCANFILE     Scans integrity of the referenced file, repairs file if problems are
               identified. Specify full path <file>.
/VERIFYFILE   Verifies the integrity of the file with full path <file>. No repair
               operation is performed.
/OFFBOOTDIR   For offline repair, specify the location of the offline boot directory
/OFFWINDIR    For offline repair, specify the location of the offline windows directory
/OFFLOGFILE   For offline repair, optionally enable logging by specifying a log file path

W-G.

sfc /SCANNOW
sfc /VERIFYFILE=c:\windows\system32\kernel32.dll
sfc /SCANFILE=d:\windows\system32\kernel32.dll /OFFBOOTDIR=d:\ /OFFWINDIR=d:\windows
sfc /SCANFILE=d:\windows\system32\kernel32.dll /OFFBOOTDIR=d:\ /OFFWINDIR=d:\windows /OFFLOGFILE=c:\log.txt
sfc /VERIFYONLY

C:\Windows\System32>sfc/SCANNOW
Beginning system scan. This process will take some time.
Beginning verification phase of system scan.
Verification 100% complete.
Windows Resource Protection did not find any integrity violations.
```

All the above is found at Reference list [2]

Task 3

Use of Linux OS CLI commands.

(1) Clear command in Linux (Input and Output)

Clears the terminal, removes all the previous commands from the window

```
localhost:~#
```

```
localhost:~# whoami
root
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~#
localhost:~# clear
```

(2) Date command in Linux (Input and Output)

Shows the date and time of your current time zone

```
localhost:~# date
Mon Feb 24 15:40:23 UTC 2025
localhost:~#
```

(3) Ls command in Linux (Input and Output)

```
localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~#
```

list and displays all files within the directory

Ps command shows the Running Processes of the device

<u>line</u>	<u>ID#</u>	<u>time</u>	<u>command</u>
1	USER	0.00	[root] /bin/sh /bin/init
2	root	0.00	[sheward]
3	root	0.00	[sheward]
4	root	0.00	[sheward@0-0]
5	root	0.00	[sheward@2-0]
6	root	0.00	[sheward@2-0]
7	root	0.00	[shurtint@0]
8	root	0.00	[shurtint@0]
9	root	0.00	[root_ranger]
10	root	0.00	[vertiblock]
11	root	0.00	[kconway]
12	root	0.00	[crypto]
13	root	0.00	[kconway]
14	root	0.00	[hiblocks]
15	root	0.00	[sheward@0-1]
16	root	0.00	[hiblocks]
17	root	0.00	[hibinet]
18	root	0.00	[hibinet]
19	root	0.00	[hibinet]
20	root	0.00	[hibinet]
21	root	0.00	[hibinet]
22	root	0.00	[hibinet]
23	root	0.00	[hibinet]
24	root	0.00	[hibinet]
25	root	0.00	[hibinet]
26	root	0.00	[hibinet]
27	root	0.00	[hibinet]
28	root	0.00	[hibinet]
29	root	0.00	[hibinet]
30	root	0.00	[hibinet]
31	root	0.00	[hibinet]
32	root	0.00	[hibinet]
33	root	0.00	[hibinet]
34	root	0.00	[hibinet]
35	root	0.00	[hibinet]
36	root	0.00	[hibinet]
37	root	0.00	[hibinet]
38	root	0.00	[hibinet]
39	root	0.00	[hibinet]
40	root	0.00	[hibinet]
41	root	0.00	[hibinet]
42	root	0.00	[hibinet]
43	root	0.00	[hibinet]
44	root	0.00	[settlam -d]
45	root	0.00	[shurd]
46	root	0.00	[shurd]
47	root	0.00	[shurd]
48	root	0.00	[shurd]
49	root	0.00	[shurd]
50	root	0.00	[shurd]
51	root	0.00	[shurd]
52	root	0.00	[shurd]
53	root	0.00	[shurd]
54	root	0.00	[shurd]
55	root	0.00	[shurd]
56	root	0.00	[shurd]
57	root	0.00	[shurd]
58	root	0.00	[shurd]
59	root	0.00	[shurd]
60	root	0.00	[shurd]
61	root	0.00	[shurd]
62	root	0.00	[shurd]
63	root	0.00	[shurd]
64	root	0.00	[shurd]
65	root	0.00	[shurd]
66	root	0.00	[shurd]
67	root	0.00	[shurd]
68	root	0.00	[shurd]
69	root	0.00	[shurd]
70	root	0.00	[shurd]
71	root	0.00	[shurd]
72	root	0.00	[shurd]
73	root	0.00	[shurd]
74	root	0.00	[shurd]
75	root	0.00	[shurd]
76	root	0.00	[shurd]
77	root	0.00	[shurd]
78	root	0.00	[shurd]
79	root	0.00	[shurd]
80	root	0.00	[shurd]
81	root	0.00	[shurd]
82	root	0.00	[shurd]
83	root	0.00	[shurd]
84	root	0.00	[shurd]
85	root	0.00	[shurd]
86	root	0.00	[shurd]
87	root	0.00	[shurd]
88	root	0.00	[shurd]
89	root	0.00	[shurd]
90	root	0.00	[shurd]
91	root	0.00	[shurd]
92	root	0.00	[shurd]
93	root	0.00	[shurd]
94	root	0.00	[shurd]
95	root	0.00	[shurd]
96	root	0.00	[shurd]
97	root	0.00	[shurd]
98	root	0.00	[shurd]
99	root	0.00	[shurd]
100	root	0.00	[shurd]

Top command Monitor System Processes in real-time

[illegible]

displays disk space usage in a human-readable format

```
localhost:~# df
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/root        5120000    2417696   2702304    47% /
devtmpfs         93464         0     93464     0% /dev
tmpfs            93628         4     93616     0% /run
none            93620         0     93620     0% /dev/shm
localhost:~#
```

```
localhost:~# free
```

	total	used	free	shared	buff/cache	available
Mem:	187244	4832	181584	4	828	179340
Swap:	0	0	0			

displays memory usage in a human-readable format.

(8) Uptime command in Linux (Input and Output)

shows the current system uptime.

```
localhost:~# uptime
12:53:37 up 1 min, load average: 0.01, 0.00, 0.00
localhost:~#
```

(9) netstat command in Linux (Input and Output)

netstat -tuln
shows all listening TCP and UDP connections.

```
localhost:~# netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
netstat: /proc/net/tcp6: No such file or directory
netstat: /proc/net/udp6: No such file or directory
netstat: /proc/net/raw6: No such file or directory
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State        I-Node Path
unix  2      [ ]         DGRAM      2961
```

(10) lscpu command in Linux (Input and Output)

lscpu
provides detailed CPU information.

```
localhost:~# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit
Byte Order:             Little Endian
Address sizes:          32 bits physical, 32 bits virtual
CPU(s):                 1
On-line CPU(s) list:   0
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):              1
Vendor ID:              AuthenticAMD
CPU family:             5
Model:                  4
Model name:             Ryzen 5 3600
Stepping:               3
CPU MHz:                3600.000
bogomips:               200.00
Virtualization:         AMD-V
Flags:                  fpu tsc mtr pae cx8 cmov clflush mmx fxsr sse sse2 rdtscl cpuid svm npt
```

All the above is found at Reference list [3]

Task 4

Use of Windows OS scripts

(1) Echo command in Batch script (Input and Output)

```
Echo Hi!
```

Echo Command Allows the output of the characters behind it, aka (Hi!)

```
C:\Users\Daniel\Downloads\OneDrive_2025-04-10\NotePad++ Bat Programs>"First test bat commands.bat"
Hi!
```

(2) Choice command in Batch script (Input and Output)

Choice Provides a list of single letter options to the user but doesn't allow duplicate choices

```
ECHO You want coffee?
ECHO Enter Y for yes
ECHO Enter N for no
CHOICE / You want coffee?
Enter Y for yes
Enter N for no
Yes or No [Y,N]?
```

(3) Date command in Batch script (Input and Output)

Displays the current date in day/month and year

```
Echo %date%
```

```
Thu 04/10/2025
```

(4) Array command in Batch script (Input and Output)

Arrays are used to store large amounts

```
Array[1]=Bob
Array[3]=Dave
Array[4]=Sam
Assigning new value
Array[3]=Zulu
Press any key to continue . . . |
```

```
@Echo off

Set Array[0]=Tony
Set Array[1]=Bob
Set Array[2]=Kerry
Set Array[3]=Dave
Set Array[4]=Sam

Echo Array[1]=%Array[1]%
Echo Array[3]=%Array[3]%
Echo Array[4]=%Array[4]%

set Array[3]=Zulu
Echo Assigning new value
Echo Array[3]=%Array[3]%

Pause
```

(5) Function command in Batch script (Input and Output)

Functions allow you to call a set of code at any time anywhere within the code

```
Hello
Function says Hi
```

```
Echo Hello
Call :Function_ID
Exit /B

:Function_ID
Echo Function says Hi

Exit /B

Pause
```

(6) Exit command in Batch script (Input and Output)

```
Echo Hi!  
exit
```

Closes the Cmd when it is run

(7) Help command in Batch script (Input and Output)

Shows a List of all commands that can be used in “.Bat Files”

```
For more information on a specific command, type HELP command-name  
ASSOC      Displays or modifies file extension associations.  
ATTRIB     Displays or changes file attributes.  
BREAK      Sets or clears extended CTRL+C checking.  
BCDEDIT    Sets properties in boot database to control boot loading.  
CACLS      Displays or modifies access control lists (ACLs) of files.  
CALL       Calls one batch program from another.  
CD         Displays the name of or changes the current directory.  
CHCP       Displays or sets the active code page number.  
CHDIR      Displays the name of or changes the current directory.  
CHKDSK     Checks a disk and displays a status report.  
CHKNTFS    Displays or modifies the checking of disk at boot time.  
CLS        Clears the screen.  
CMD        Starts a new instance of the Windows command interpreter.  
COLOR      Sets the default console foreground and background colors.  
COMP       Compares the contents of two files or sets of files.  
COMPACT    Displays or alters the compression of files on NTFS partitions.  
CONVERT    Converts FAT volumes to NTFS. You cannot convert the  
            current drive.  
COPY       Copies one or more files to another location.  
DATE       Displays or sets the date.  
DEL        Deletes one or more files.  
DIR        Displays a list of files and subdirectories in a directory.  
DISKPART   Displays or configures Disk Partition properties.  
DOSKEY     Edits command lines, recalls Windows commands, and  
            creates macros.  
DRIVERQUERY Displays current device driver status and properties.  
ECHO       Displays messages, or turns command echoing on or off.
```

(8) Variables command in Batch script (Input and Output)

To create a variable, use the set command write the variable name and then set what you want the variable to hold within

```
Set Name=Yourname  
Echo %Name%
```

(9) Operators command in Batch script (Input and Output)

Operators are used for Mathematical calculations within code

```
6  
6  
Sum = 12
```

```
set /A A=6  
set /A B=6  
  
Echo %A%  
Echo %B%  
  
set /A Sum=A+B  
Echo Sum = %Sum%  
  
pause
```

(10) Goto command in Batch script (Input and Output)

jumps to the function name that is called after the command

```
Jump Worked  
Press any key to continue . . . |
```

```
goto Jump  
Echo Hello  
:Jump  
Echo Jump Worked  
Pause
```

All the above is found at Reference list [4]

Task 5

Use of Linux OS scripts

(1) Bash Command Echo (Input and Output)

used to display sentences in the terminal

```
echo "Hello World";
```

```
Hello World
```

(2) Bash Variables (Input and Output)

Variables are used to store data such as names and ages

```
name="Alice"  
echo "Hello, $name!"
```

(3) Loops Bash Command (Input and Output)

For loops are used to repeatedly execute a piece of code

```
for element in 1 2 3 4 5  
do  
echo $element  
done
```

(4) Arrays Bash Command (Input and Output)

Arrays are used to store a collection of the same data type

```
nums=(1 3 12)  
for i in "${nums[@]}"  
do  
echo "$i"  
done
```

(5) Operators Bash Command (Input and Output)

Operators are used for any and mathematical calculations within code

```
A=3
B=7

Sum=$((A+B))
echo "A + B"
echo $Sum
```

(6) functions in Bash Command (Input and Output)

Functions are used to call sections of code from anywhere any time

```
add_numbers() {
    local A=$1
    local B=$2
    add 3 and 7
    The sum of 3 and 7 is: 10
}

# Call the function with arguments
echo "add 3 and 7"
add_numbers 3 7
```

(7) Case statement Bash Command (Input and Output)

Case statements are used to when you need multiple if statement as it is a smarter version of the if statement

```
Default Name.
```

```
name="john1"

case $name in
    "john")
        echo "John."
        ;;
    "others" | "others2")
        echo "Other names."
        ;;
    *)
        echo "Default Name."
        ;;
esac
```

(8) Read bash Command (Input and Output)

The read command allows for user input into the code

```
What is your name?
Josh
Hello, Josh!
```

```
echo "What is your name?"
read name

# Use the input
echo "Hello, $name!"
```

All the above is found at Reference list [5]

Conclusion

In conclusion i have found that if it wasn't for the OS's that we have today our computers would be much slower and much more unreliable than they are now, they allow us to use our computers in a way that help us on a daily basis and if a problem occurs it has the tools required to help us fix it, Moving on in using windows compared to Linux I find it easier to use windows as it allows for a more visual representation of what you are doing compared to Linux which is all single lined texted based

OS tutorial activity 1 Different types of OS's

There are many different types of OS's, and they are all used for different proposes such as:

General-purpose OS's:

like windows, Linux and MacOS

Mobile OS's:

like Android and IOS.

As well as other types of OS's like:

Network OS's:

This OS allows a single network to be shares across multiple users and shares the resources between them all

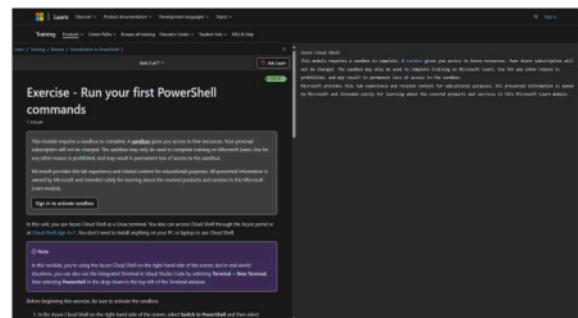
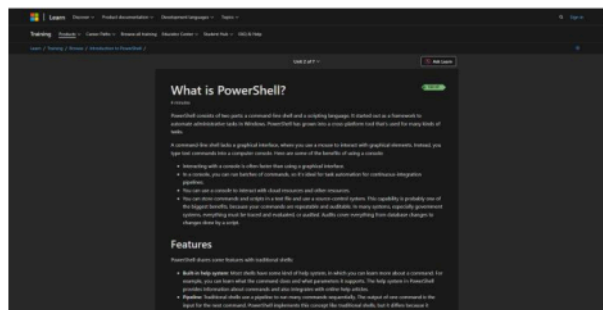
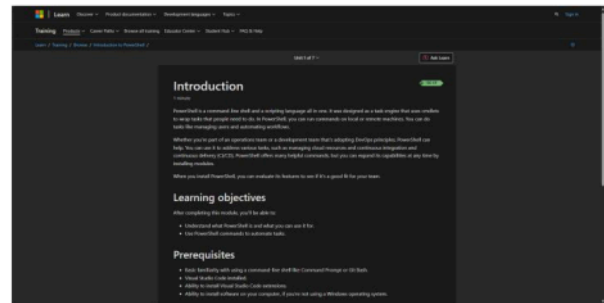
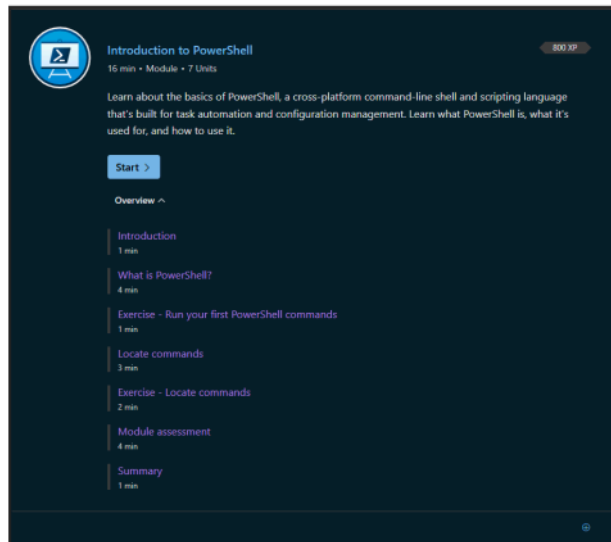
Multitasking OS's:

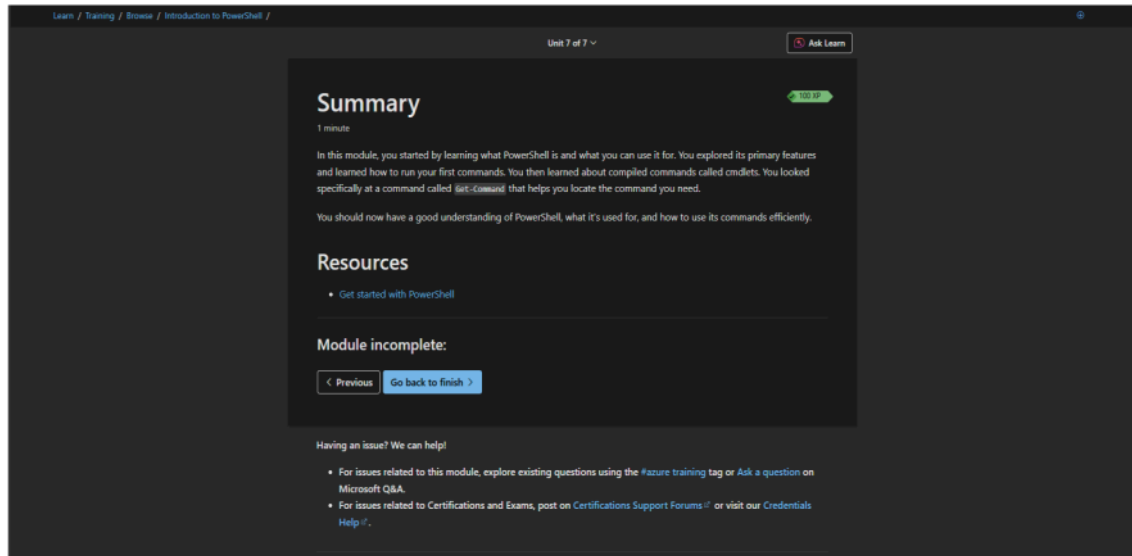
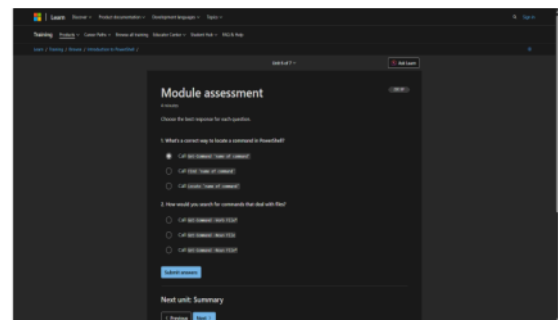
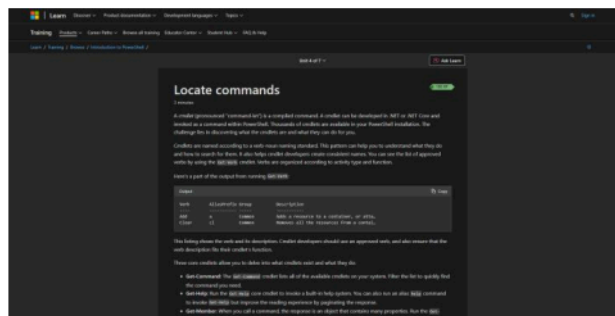
Also known as a Time-sharing OS, this type of OS is generally used in general purpose OS's such as windows or Linux, it shares the CPU time with other users or tasks that are being used by the system so that multiple process can run at a time

Mobile OS's

This OS is mainly used in mobile devices as it is developed and optimized for more touchscreen-based input like the Android phone or Apple Phones to help them to be more power efficient, this type of OS is also used in a lot of wireless devices making it easier to use or connect to other devices.

Evidence of an online course:





References

- [1] Anonymous (-12-11). *Functions of Operating System*. Available: <https://www.geeksforgeeks.org/functions-of-operating-system/>.
- [2] Microsoft. *Microsoft Batch Line Commands*. Available: [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-r2-and-2008/cc749154\(v%3dws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-r2-and-2008/cc749154(v%3dws.11)).
- [3] Anonymous (-07-14). *Linux Commands Cheat Sheet: Beginner to Advanced 2025*. Available: <https://www.geeksforgeeks.org/linux-commands-cheat-sheet/>.

[4] Anonymous Anonymous Batch file commands (A-Z) | explanation and examples. Batch file commands (A-Z) | explanation and examples. Available: <https://www.trytoprogram.com/batch-file-commands/>.

[5] Anonymous . *Bash Shell Scripting Tutorials and Examples* - w3schools. Available: <https://www.w3schools.io/terminal/bash-tutorials/>.

OS Assignment

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

76/100

Total:		76
1. Engagement with Literature Skills:		
+ Some good use of citations. + formatted reference list.	20	10
2. Knowledge and Understanding Skills:		
Key functions: + Good overview of key OS functions documented. - could be extended to include 1/2 additional areas. + Screenshots showing where some of the functions can be seen/monitored/managed are documented. Win Commands: + Good overview of relevant commands that demonstrates an understanding of the language. - 1/2 additional options could be described. Bash Commands: + V. good overview of relevant commands that demonstrates a sound understanding of the language. Win Scripts: + Good overview of the syntax that demonstrates an understanding of the scripting aspect of the language. - some scripts could be enhanced with some citations to demonstrate related research to differentiate and enhance use of the scripting aspect of the language.	20	17

Bash Scripts: + V. good overview of the syntax that demonstrates a clear understanding of the scripting aspect of the language.		
3. Cognitive and Intellectual Skills:		
+ Evidence of good analysis and understanding of OS functions. – Analysis could be extended to briefly include 1/2 additional areas. + Screenshots showing where some of the functions can be seen/monitored/managed are included in the analysis. + Conclusion.	10	8
4. Practical Application Skills:		
1. Win Commands: + Good use of relevant Win. language commands. – 1/2 additional options could be used. 2. Bash Commands: + V. good use of relevant Bash commands. 3. Win Scripts: + Good use of Win scripts to demonstrate applied knowledge of the core syntax. – some scripts could be further customised by using and citing some research to differentiate and enhance the use of the scripting aspect of the language. 4. Bash Scripts: + V. good use of Bash scripts to demonstrate applied knowledge of the core syntax.	40	36
5. Transferable Skills for Life and Professional Practice:		

+ Report format, structure, and presentation quite good. - Title page and Toc should be on separate pages. - Try to avoid using 'I' first person tense when writing a report. - Each report section could be numbered and be briefly introduced. + Tutorial Activity 1 on OS Types: relevant supplementary information on OS types is included in the Appendix of the report. – could be extended to include 1/2 additional OS types. + Evidence of engagement with a relevant CPD course documented in the appendix. - Lecture activity evidence not documented.	10	5
Plagiarism & On-time check:		/

PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7
PAGE 8
PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18
