

OS Assignment 1

by Osian Murphy

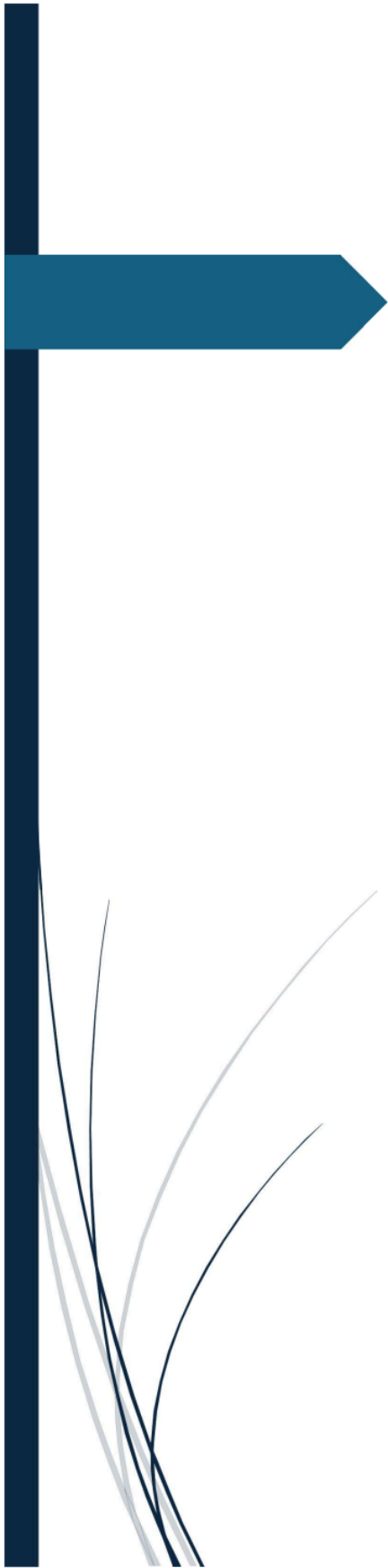
Submission date: 12-May-2026 04:57PM (UTC+0100)

Submission ID: 283268182

File name: 58793_Osian_Murphy_OS_Assignment_1_684823_268602044.docx (29.97M)

Word count: 3647

Character count: 30940



Assignment 1

Osian Murphy (2405124)

Contents

Task 1.....	1
Task 2.....	4
Task 3.....	10
Task 4.....	15
Screenshots.....	16
Script.....	18
Task 5.....	33
Screenshots.....	34
Script.....	35
Conclusion.....	37
References	38
Appendices	38
Different Types of OS	38
General Purpose OS's.....	38
Mobile Operating Systems.....	39
Network Operating Systems	39
Proof of Engagement with an Online Course	39

Task 1

General purpose operating systems (OS) for example Microsoft Windows, Linux and macOS give users a platform to manage computer hardware and software. They act as a layer between the user and the system, making sure of efficient execution of applications and system stability overall.

One of the primary functions of an operating system is process management. Process management involves creating, terminating and scheduling processes. The operating system uses scheduling to allocate CPU time among active processes fairly, to allow multitasking and keep responsiveness. Using task manager on windows can display this:

Name	CPU	Memory	GPU	Disk	Network	Status
Discord (8)	1.3%	1,507.9 MB	0.4%	0 MB/s	0.1 Mbps	Running
Opera GX Internet Browser (48)	0.8%	3,174.6 MB	0.8%	0.1 MB/s	0.1 Mbps	Running
Desktop Window Manager	0.8%	72.3 MB	5.5%	0 MB/s	0 Mbps	Running
RazerAppEngine (11)	0.5%	161.2 MB	0%	0 MB/s	0 Mbps	Running
EpicGamesLauncher	0.4%	29.2 MB	0%	0.1 MB/s	0 Mbps	Running
System	0.4%	0.1 MB	0%	0.1 MB/s	0 Mbps	Running
Windows Explorer	0.4%	115.6 MB	0.1%	0 MB/s	0 Mbps	Running
Conair CUE Component	0.3%	15.7 MB	0%	0 MB/s	0 Mbps	Running
Task Manager	0.3%	94.8 MB	0%	0.1 MB/s	0 Mbps	Running
Service Host: Local System (16)	0.3%	61.3 MB	0%	0 MB/s	0 Mbps	Running
Pasarc (2)	0.2%	31.6 MB	0%	0 MB/s	0 Mbps	Running
Microsoft Word (2)	0.1%	146.1 MB	0.1%	0 MB/s	0 Mbps	Running
Conair Device Control Service...	0.1%	0.5 MB	0%	0 MB/s	0 Mbps	Running
CUE	0.1%	14.9 MB	0%	0 MB/s	0 Mbps	Running
Microsoft Visual Studio	0.1%	1,500.7 MB	0.1%	0 MB/s	0 Mbps	Running
JetBrains / JetBrains.Profiles.D...	0.1%	48.8 MB	0%	0.1 MB/s	0 Mbps	Running
Services and Controller app	0.1%	5.7 MB	0%	0 MB/s	0 Mbps	Running
Steam Client WebHelper (7)	0.1%	86.4 MB	0%	0 MB/s	0 Mbps	Running
wallpaper2.exe (12 bit)	0.1%	6.4 MB	1.2%	0 MB/s	0 Mbps	Running
NordVPN	0.1%	257.3 MB	0%	0 MB/s	0.1 Mbps	Running

On Linux, tools like ps aux or top can display this also:

Ps aux

```

root@kali:~# ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1.0  0.0  0.0      0     0 ttyS0    Ss+  Mon04   0:00 systemd
root         2.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root         3.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root         4.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root         5.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root         6.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root         7.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root         8.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root         9.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        10.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        11.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        12.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        13.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        14.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        15.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        16.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        17.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        18.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        19.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        20.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        21.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        22.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        23.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        24.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        25.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        26.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        27.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        28.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        29.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        30.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        31.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        32.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        33.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        34.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        35.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        36.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        37.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        38.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        39.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        40.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        41.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        42.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        43.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        44.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        45.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        46.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        47.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        48.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        49.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        50.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        51.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        52.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        53.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        54.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        55.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        56.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        57.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        58.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        59.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        60.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        61.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        62.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        63.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        64.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        65.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        66.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        67.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        68.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        69.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        70.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        71.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        72.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root
root        73.0  0.0  0.0      0     0        *    S    Mon04   0:00 /usr/lib/systemd/systemd --root

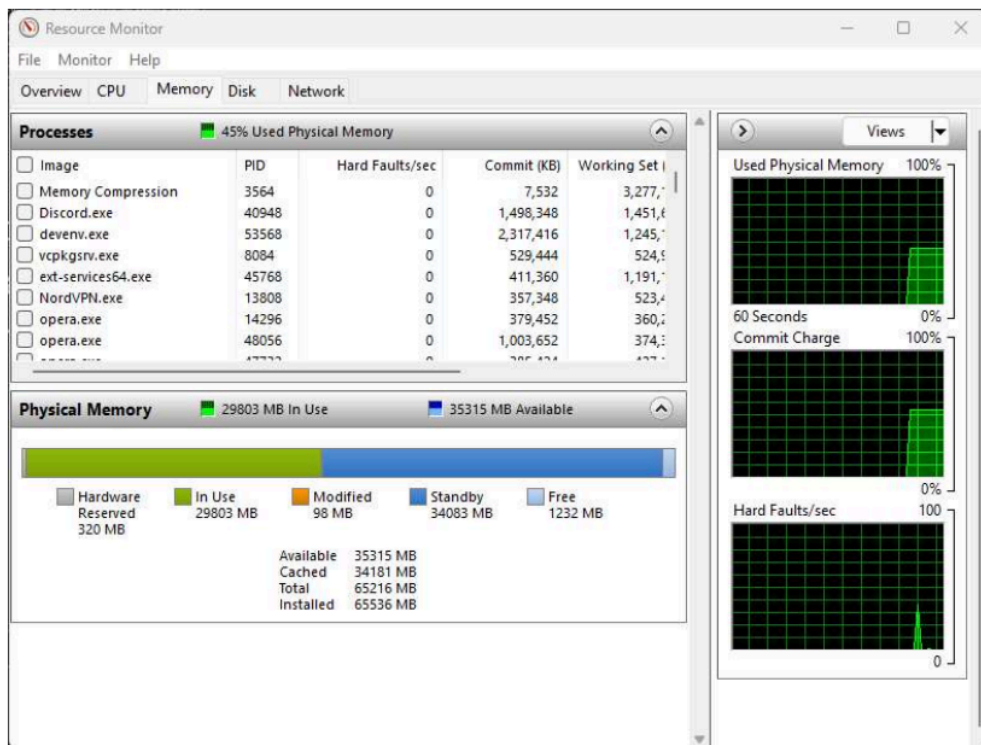
```

top

```
Apr 23 1:02 PM
root@ubuntu:~# top
top - 11:00:00 up 1 min, 1 user, load average: 0.00, 0.01, 0.00
task: 387 total, 1 running, 386 sleep, 0 prio, 0 sleep, 0 zombie
procs: 2.0 us, 0.0 sy, 0.0 id, 99.0 wa, 0.0 hi, 0.0 si, 0.0 st
Mem: 7886.3 total, 1378.7 free, 1312.5 used, 1884.5 buff/cache
Mem: 7886.3 total, 1378.7 free, 0.0 used, 1884.5 avail Mem

#PID#PPID#P#UID#VIRT#RES#SHR#S#C#ST#D#COMMAND
100 root 20 0 0 720108 41204 10240 1 20 0 0.1 0.0 0.00.00 gpmc-wmcli
200 root 20 0 0 0 0 0 0 0 0 0.0 0.0 0.00.00 xserver[611:27-xserver_cobound
2000 xserver 20 0 0 3421088 282272 134128 4 4 0 0.0 0.0 0.00.00 xorg
3376 xserver 20 0 0 14464 7056 1208 0 0 0 0.1 0.0 0.00.00 xorg
1 root 20 0 0 20480 1112 0 0 0 0.0 0.0 0.00.00 systemd
2 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
4 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
6 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
8 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
10 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
12 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
14 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
16 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
18 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
20 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
22 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
24 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
26 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
28 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
30 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
32 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
34 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
36 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
38 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
40 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
42 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
44 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
46 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
48 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
50 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
52 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
54 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
56 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
58 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
60 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
62 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
64 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
66 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
68 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
70 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
72 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
74 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
76 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
78 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
80 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
82 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
84 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
86 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
88 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
90 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
92 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
94 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
96 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
98 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
100 root 20 0 0 0 0 0 0 0 0.0 0.0 0.00.00 systemd
```

Another important function of an OS is memory management. Memory management is what controls the allocation and deallocation of system memory. Newer operating systems implement virtual memory, which allow processes to use more memory than what is physically available on the system, it does this by using a file called `pagefile.sys` to offload data onto it. This file is stored on the disk.



An operating system is also responsible for file system management. This is the process of organising, storing and retrieving storage devices. Different operating systems use different filesystems, for example Windows uses NTFS, while Linux uses ext4 (just examples there are other file system formats).

Another function of an operating system is security and management of users. OS' use authentication mechanisms and control access to system resources by using permissions and user accounts. On windows, there are tools such as Local Users and Groups, however on Linux it is managed through configuration files such as `/etc/passwd` and using commands like `chown` or `chmod`

Task 2

Windows

`dir` - `dir` lists files/directories in whatever directory you want

```
C:\Windows>dir
Volume in drive C has no label.
Volume Serial Number is

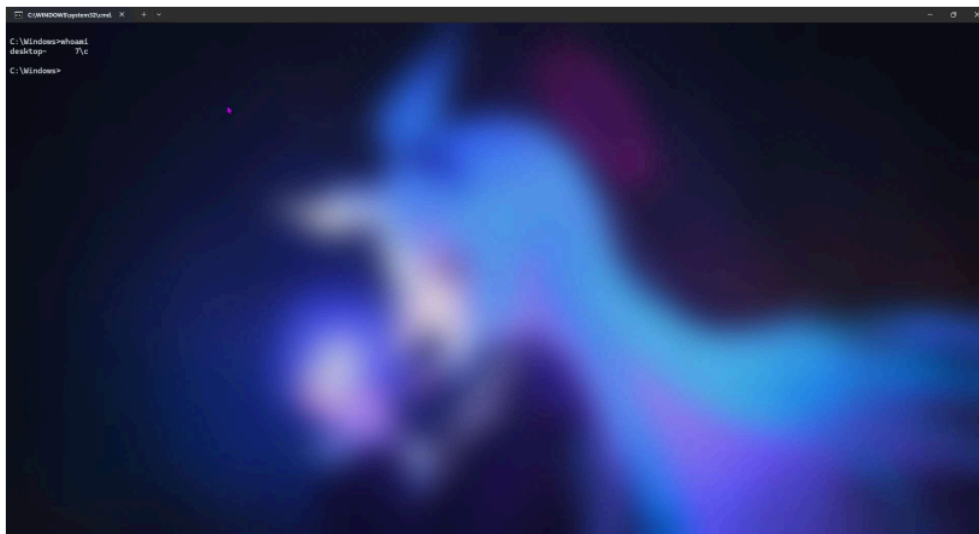
Directory of C:\Windows

13/02/2025  22:17    <DIR>
16/06/2022  16:23             13,076 acinetof.dll
18/01/2025  22:28    <DIR>
14/02/2025  13:44    <DIR>
16/02/2025  14:31    <DIR>
23/01/2025  20:28    <DIR>
13/02/2025  22:17    <DIR>
13/02/2025  20:25             126,978 bfave.exe
01/04/2024  07:34    <DIR>
01/04/2024  07:26    <DIR>
18/01/2025  21:24    <DIR>
18/02/2025  14:53             762 chesup.log
18/01/2025  21:09    <DIR>
29/01/2025  15:53    <DIR>
01/04/2024  07:22             23,085 Core.exe
29/01/2025  15:55    <DIR>
01/04/2024  07:26    <DIR>
13/02/2025  20:38    <DIR>
01/04/2024  07:34    <DIR>
01/04/2024  07:24    <DIR>
01/04/2024  16:12    <DIR>
13/12/2024  14:08             27,331 DirectX.log
13/01/2025  14:55             94,866 GPRINT.LDS
18/01/2025  21:24             1,156 OciInstall.log
18/01/2025  21:24    <DIR>
18/01/2025  21:24    <DIR>
13/02/2025  20:25             2,811,264 explorer.exe
01/04/2024  07:26    <DIR>
01/04/2024  16:14    <DIR>
01/04/2024  16:12    <DIR>
13/02/2025  20:25             1,885,008 Help.exe
13/02/2025  20:25             409,968 hh.exe
01/04/2024  07:34    <DIR>
18/01/2025  21:24    <DIR>
13/02/2025  22:17    <DIR>
13/02/2025  22:17    <DIR>
14/02/2025  11:44    <DIR>
22/04/2024  06:37             7,001 Info.exe
01/04/2024  07:34    <DIR>
18/01/2025  21:24    <DIR>
01/04/2024  07:26    <DIR>
18/01/2025  21:06    <DIR>
20/01/2025  02:07             43,131 mib.bin
01/04/2024  07:22
```

cd - allows you to traverse between directories

```
C:\Windows\System32>cd ..
C:\Windows>
```

echo – prints the string you enter example: "echo hello" this would print "hello"



chkdsk - scans for errors on the disk, then fixes them

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>chkdsk
The type of the file system is NTFS.

WARNING! /F parameter not specified.
Running CHKDSK in read-only mode.

Stage 1: Examining basic file system structure ...
 294144 file records processed.
File verification completed.
Phase duration (File record verification): 1.62 seconds.
 16506 large file records processed.
Phase duration (Orphan file record recovery): 6.60 milliseconds.
 0 bad file records processed.
Phase duration (Bad file record checking): 0.74 milliseconds.

Stage 2: Examining file name linkage ...
 7628 reparse records processed.
 414898 index entries processed.
Index verification completed.
Phase duration (Index verification): 4.34 seconds.
 0 unindexed files scanned.
Phase duration (Orphan reconnection): 116.98 milliseconds.
 0 unindexed files recovered to lost and found.
Phase duration (Orphan recovery to lost and found): 0.80 milliseconds.
 7628 reparse records processed.
Phase duration (Reparse point and Object ID verification): 15.12 milliseconds.

Stage 3: Examining security descriptors ...
Security descriptor verification completed.
Phase duration (Security descriptor verification): 18.44 milliseconds.
 60378 data files processed.
Phase duration (Data attribute verification): 0.81 milliseconds.
CHKDSK is verifying Usn Journal...
 36137288 USN bytes processed.
Usn Journal verification completed.
Phase duration (USN journal verification): 106.23 milliseconds.

Windows has scanned the file system and found no problems.
No further action is required.

 82977791 KB total disk space.
 25912232 KB in 119516 files.
 122160 KB in 60379 indexes.
 0 KB in bad sectors.
 406739 KB in use by the system.
 65536 KB occupied by the log file.
 56536660 KB available on disk.

 4096 bytes in each allocation unit.
 20744447 total allocation units on disk.
 14134165 allocation units available on disk.
Total duration: 6.23 seconds (6234 ms).

C:\Windows\System32>
```

sfc /scannow - scans for damaged/corrupted system files, then fixes them

```

Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

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 found corrupt files and successfully repaired them.
For online repairs, details are included in the CBS log file located at
windir\Logs\CBS\CBS.log. For example C:\Windows\Logs\CBS\CBS.log. For offline
repairs, details are included in the log file provided by the /OFFLOGFILE flag.

C:\Windows\System32>

```

systeminfo - displays system info

```

C:\Windows\System32>systeminfo

Host Name:                               Microsoft Windows 11 Pro
OS Name:                                 Microsoft Windows 11 Pro
OS Version:                             10.0.22H2 N/A Build 22H2
OS Manufacturer:                         Microsoft Corporation
OS Configuration:                       Standalone Workstation
OS Build Type:                            Multiprocessor Free
Registered Owner:
Registered Organization:
Product ID:                               06330-00000-00000-AA131
Original Install Date:                   18/01/2025, 21:51:09
System Boot Time:                        18/01/2025, 11:04:01
System Manufacturer:                     Micro-Star International Co., Ltd.
System Model:                             MS-7C26
System Type:                             x64-based PC
Processor(s):                             1 Processor(s) Installed.
[01]: AMD Family 17 Model 97 Stepping 2 AuthenticAMD ~4791 Mhz
      American Megatrends International, LLC 1.00, 27/03/2024
BIOS Version:                             C:\WINDOWS
Windows Directory:                       C:\WINDOWS
System Directory:                         C:\WINDOWS\System32
Boot Device:                             \Device\HarddiskVolume1
System Locale:                             en-gb;English (United Kingdom)
Input Locale:                             en-gb;English (United Kingdom)
Time Zone:                               (UTC+00:00) Dublin, Edinburgh, Lisbon, London
Total Physical Memory:                    65,217 MB
Available Physical Memory:                 45,510 MB
Virtual Memory: Max Size:                 65,213 MB
Virtual Memory: Available:                 44,128 MB
Virtual Memory: In Use:                    21,085 MB
Page File Location(s):                    C:\pagefile.sys
Domain:                                   WORKGROUP
Logon Server:
Hotfix(s):                                4 Hotfix(s) Installed.
[01]: KB5009622
[02]: KB5009197
[03]: KB5009387
[04]: KB5009388
5 NIC(s) Installed.
[01]: VMware Virtual Ethernet Adapter for VMnet1
      Connection Name: VMware Network Adapter VMnet1
      DHCP Enabled:   Yes
      DHCP Server:
      IP address(es):
      [01]:
      [02]:
[02]: VMware Virtual Ethernet Adapter for VMnet8
      Connection Name: VMware Network Adapter VMnet8
      DHCP Enabled:   Yes
      DHCP Server:

```

Task 3

Linux

```
naomi@DESKTOP-DKL6L17: ~  
naomi@DESKTOP-DKL6L17:/$ cd  
naomi@DESKTOP-DKL6L17:~$ cd /  
naomi@DESKTOP-DKL6L17:/$ cd  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh  test123  
naomi@DESKTOP-DKL6L17:~$
```

Ls – lists files/directories in the active directory

```
naomi@DESKTOP-DKL6L17: /  
naomi@DESKTOP-DKL6L17:/$ cd  
naomi@DESKTOP-DKL6L17:~$ cd /  
naomi@DESKTOP-DKL6L17:/$ |
```

Cd – allows you to traverse between directorie

```
naomi@DESKTOP-DKL6L17: ~  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh  
naomi@DESKTOP-DKL6L17:~$ touch test123  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh  test123  
naomi@DESKTOP-DKL6L17:~$
```

Touch – Update the access and modification times of each file to the current time. (can also create a file if it doesn't exist yet)

```
naomi@DESKTOP-DKL6L17:~$ cat test.sh  
#!/bin/bash  
set -m pipefail  
shopt -s nocasematch  
  
print_usage() {  
    cat << "EOF"  
  
    [ ASCII art of the letter 'A' made of dashes ]  
  
    EOF  
}  
  
ask_question() {  
    local correct="$1"  
    local question="$2"  
    local answer_a="$3"  
    local answer_b="$4"  
    local answer_c="$5"  
    local answer_d="$6"  
  
    clear  
    print_usage  
    echo "Question"  
    echo "%s" $question  
    echo "%s" $answer_a  
    echo "%s" $answer_b  
    echo "%s" $answer_c  
    echo "%s" $answer_d  
  
    read -p "Your answer: " given_answer  
    if [[ "$given_answer" == "$correct" ]]; then  
        echo "Don't go down that path, trust me. Been there, done that. Just stick to the quiz, eh?"  
    fi  
    if [[ "$given_answer" == "$correct" ]]; then  
        echo "Correct!"  
        sleep 2  
    else  
        echo "Wrong! Try again."  
        exit 0  
    fi  
}  
  
ask_question "a" "How tall is an among us character?" "3'6\"" "4'2\"" "5'6\"" "4'1\"" "  
ask_question "a" "How many colors are there in among us?" "8" "10" "12" "14" "  
ask_question "a" "When was among us released?" "2019" "2018" "2020" "2021" "  
ask_question "a" "What is the name of the company that made among us?" "Innerloth" "Valve" "riot games" "epic games" "  
ask_question "a" "Which role is among us not playable?" "crewmate" "impster" "engineer" "scientist" "  
ask_question "a" "How many tails does a crewmate have to go to win? (trick question)" "4" "10" "15" "20" "  
ask_question "a" "What task involves connecting wires?" "admin" "electrical" "download/upload" "empty garbage" "  
ask_question "a" "What is the default map on among us?" "The Skeld" "Mira HQ" "Polus" "Mirage" "  
ask_question "a" "What color is usually sus?" "red" "blue" "green" "yellow" "  
ask_question "a" "Which role can temporarily take another player's appearance?" "crewmate" "impster" "shapeshifter" "scientist" "  
ask_question "a" "What is the cooldown before an impster can eliminate again called?" "kill cooldown" "elimination timer" "impster cooldown" "kill timer"
```

Cat – prints a files contents

```
naomi@DESKTOP-DKL6L17: ~  
naomi@DESKTOP-DKL6L17:~$ cd  
naomi@DESKTOP-DKL6L17:~$ cd /  
naomi@DESKTOP-DKL6L17:/$ cd  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh  test123  
naomi@DESKTOP-DKL6L17:~$ rm test123  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh  
naomi@DESKTOP-DKL6L17:~$
```

Rm – removes file

```
naomi@DESKTOP-DKL6L17: ~  
naomi@DESKTOP-DKL6L17:~$ cp test.sh test2.sh  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh  test2.sh  
naomi@DESKTOP-DKL6L17:~$
```

Cp – copies files

```
naomi@DESKTOP-DKL6L17: ~  
naomi@DESKTOP-DKL6L17:~$ cp test.sh test2.sh  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh  test2.sh  
naomi@DESKTOP-DKL6L17:~$ mkdir test3  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh  test2.sh  test3  
naomi@DESKTOP-DKL6L17:~$ |
```

Mkdir – creates a directory

```
MAN(1) Manual pager utils MAN(1)  
NAME  
man - an interface to the system reference manuals  
SYNOPSIS  
man [man options] [[section] page ...] ...  
man -k [apropos options] regexp ...  
man -K [man options] [section] term ...  
man -f [whatis options] page ...  
man -l [man options] file ...  
man -w|-W [man options] page ...  
DESCRIPTION  
man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.  
The table below shows the section numbers of the manual followed by the types of pages they contain.  
1 Executable programs or shell commands  
2 System calls (functions provided by the kernel)  
3 Library calls (functions within program libraries)  
4 Special files (usually found in /dev)  
5 File formats and conventions, e.g. /etc/passwd  
6 Games  
7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7), man-pages(7)  
Manual page man(1) line 1 (press h for help or q to quit)
```

Man – gives documentation of a specific command

```
naomi@DESKTOP-DKL6L17: ~  
naomi@DESKTOP-DKL6L17:~$ cp test.sh test2.sh  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh test2.sh  
naomi@DESKTOP-DKL6L17:~$ mkdir test3  
naomi@DESKTOP-DKL6L17:~$ ls  
test.sh test2.sh test3  
naomi@DESKTOP-DKL6L17:~$ man mkdir  
naomi@DESKTOP-DKL6L17:~$ man man  
man: can't resolve man7/groff_man.7  
naomi@DESKTOP-DKL6L17:~$ date  
Tue Apr 28 23:46:10 BST 2026  
naomi@DESKTOP-DKL6L17:~$
```

Date – shows the current date and time

```
naomi@DESKTOP-DKL6L17:~$ man mount  
naomi@DESKTOP-DKL6L17:~$ mount  
none on /usr/lib/modules/6.6.87.2-microsoft-standard-WSL2 type overlay (rw,nosuid,nodev,noatime,lowerdir=/modules,upperdir=/lib/modules/6.6.87.2-microsoft-standard-WSL2/rw/upper,workdir=/lib/modules/6.6.87.2-microsoft-standard-WSL2/rw/work,uuid=on)  
none on /mnt/wsl type tmpfs (rw,relatime)  
drivers on /usr/lib/wsl/drivers type 9p (ro,nosuid,nodev,noatime,aname=drivers,fmask=222,dmask=222,cache=5,access=client,msize=65536,trans=fd,rfd=8,wfd=8)  
/dev/sdd on / type ext4 (rw,relatime,discard,errors=remount-ro,data=ordered)  
none on /mnt/wslg type tmpfs (rw,relatime)  
/dev/sdd on /mnt/wslg/distro type ext4 (ro,relatime,discard,errors=remount-ro,data=ordered)  
none on /usr/lib/wsl/lib type overlay (rw,nosuid,nodev,noatime,lowerdir=/gpu_lib_packaged:/gpu_lib_inbox,upperdir=/gpu_lib/wsl/upper,workdir=/gpu_lib/wsl/work,uuid=on)  
rootfs on /init type rootfs (ro,size=16346296k,nr_inodes=4086574)  
none on /dev type devtmpfs (rw,nosuid,relatime,size=16346296k,nr_inodes=4086574,mode=755)  
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,noatime)  
proc on /proc type proc (rw,nosuid,nodev,noexec,noatime)  
devpts on /dev/pts type devpts (rw,nosuid,noexec,noatime,gid=5,mode=620,ptmxmode=000)  
none on /run type tmpfs (rw,nosuid,nodev,mode=755)  
none on /run/lock type tmpfs (rw,nosuid,nodev,noexec,noatime)  
none on /run/shm type tmpfs (rw,nosuid,nodev,noatime)  
none on /dev/shm type tmpfs (rw,nosuid,nodev,noatime)  
none on /run/user type tmpfs (rw,nosuid,nodev,noexec,noatime,mode=755)  
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc (rw,relatime)  
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)  
none on /mnt/wslg/versions.txt type overlay (rw,relatime,lowerdir=/systemvhd,upperdir=/system/rw/upper,workdir=/system/rw/work,uuid=on)  
none on /mnt/wslg/doc type overlay (rw,relatime,lowerdir=/systemvhd,upperdir=/system/rw/upper,workdir=/system/rw/work,uuid=on)  
none on /tmp/.X11-unix type tmpfs (ro,relatime)
```

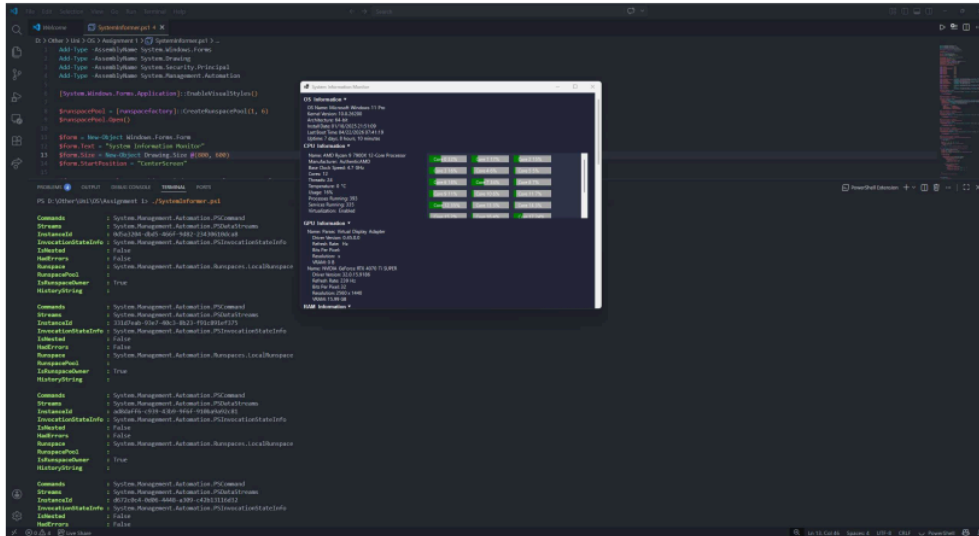
Mount – lets you mount a usb or filesystem

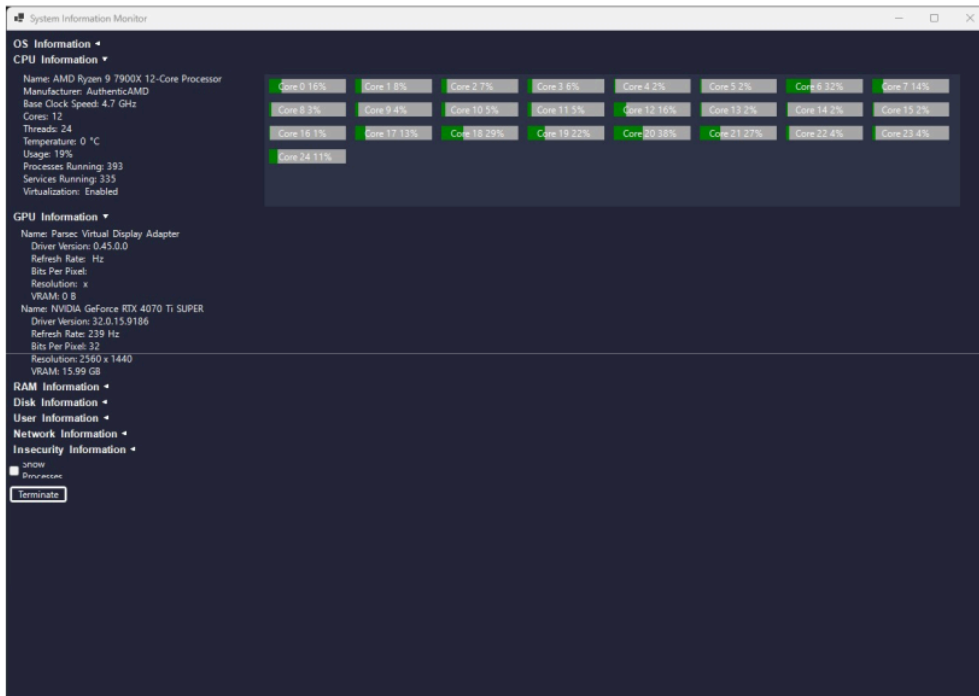
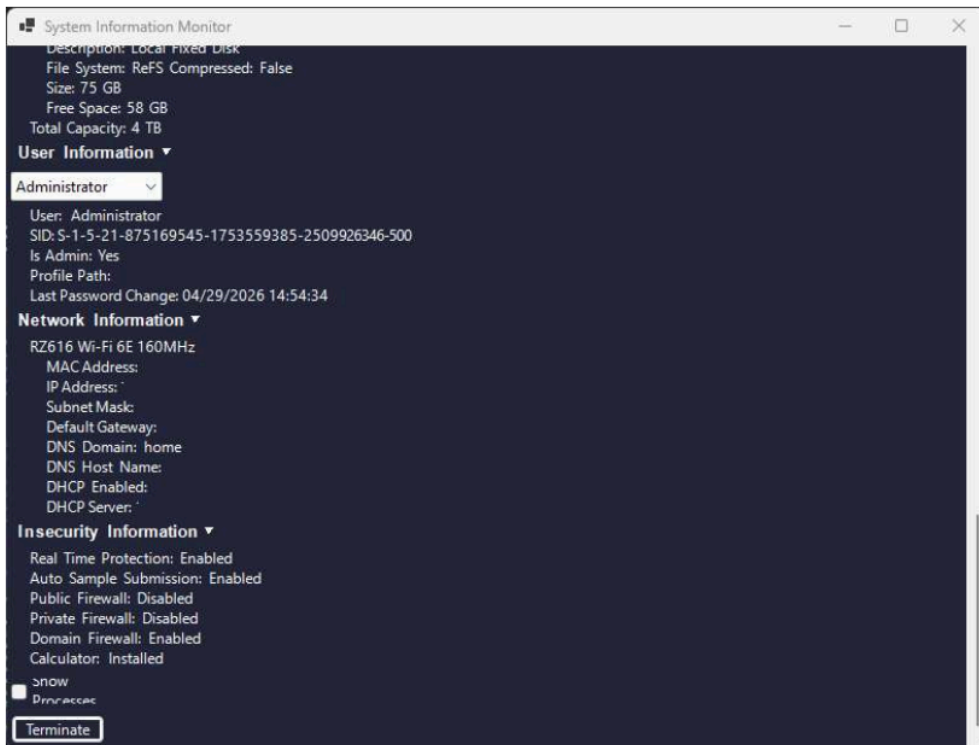
Task 4

Powershell uses object-oriented syntax, which allows interaction with .NET libraries. The script I have created imports GUI related assemblies using `Add-Type``, which allows the creation of Windows Forms interfaces. In the script, I initialize objects using

`New-Object`, and their properties are configured by using dot notation, for example `\$form.Text`. I also used Global variables (`\$Global:`) to store UI elements and system data across the entire script. I retrieved system information using `Get-WmiObject`, which queries Windows Management Instrumentation (WMI) classes for example `Win32\_OperatingSystem`, this retrieves the operating system. In the script, I also used runspace, which allowed the script to use multithreading, which was to improve the performance of the script, as well as responsiveness of the script.

Screenshots





Script

```
Add-Type -AssemblyName System.Windows.Forms
Add-Type -AssemblyName System.Drawing
Add-Type -AssemblyName System.Security.Principal
Add-Type -AssemblyName System.Management.Automation

[System.Windows.Forms.Application]::EnableVisualStyles()

$runspacePool = [runspacefactory]::CreateRunspacePool(1, 6)
$runspacePool.Open()

$form = New-Object Windows.Forms.Form
$form.Text = "System Information Monitor"
$form.Size = New-Object Drawing.Size @(800, 600)
$form.StartPosition = "CenterScreen"

$formContentPanel = New-Object Windows.Forms.FlowLayoutPanel
$formContentPanel.Dock = "Fill"
$formContentPanel.AutoScroll = $true
$formContentPanel.FlowDirection = "TopDown"
$formContentPanel.WrapContents = $false
$formContentPanel.BackColor = [Drawing.Color]::FromArgb(36, 39, 58)
$formContentPanel.ForeColor = [Drawing.Color]::White
$topPaddingControl = New-Object Windows.Forms.Label
$topPaddingControl.Height = 10
$formContentPanel.Controls.Add($topPaddingControl)
$form.Controls.Add($formContentPanel)

$Global:osInfoLbl = $null
$Global:cpuInfoLbl = $null
$Global:cpuInfoSplitContainerPanel = $null
$Global:cpuInfoBarContainerPanel = $null
$Global:cpuInfoGfx = $null
$Global:gpuInfoLbl = $null
$Global:ramInfoLbl = $null
$Global:diskInfoLbl = $null
$Global:userInfoLbl = $null
$Global:ntwkInfoLbl = $null
$Global:secInfoLbl = $null

$Global:processListView = $null
$Global:processListCheckbox = $null
$Global:termBtn = $null

$Global:osInformation = Get-WmiObject -Class
Win32_OperatingSystem | Select-Object -Property Caption, Version,
OSArchitecture, BuildNumber, InstallDate, LastBootUpTime, LocalDateTime
```

```

$Global:cpuUsagePerCoreInformation = Get-WmiObject -Class
Win32_PerfFormattedData_PerfOS_Processor | Select-Object PercentIdleTime
$cpuLogicalThreadCount = $Global:cpuUsagePerCoreInformation.length
$Global:cpuInformation = Get-WmiObject -Class Win32_Processor |
Select-Object VirtualizationFirmwareEnabled, LoadPercentage, NumberOfCores,
ThreadCount, Name, Manufacturer, CurrentClockSpeed
$Global:cpuTempInformation = Get-WmiObject
MSAcpi_ThermalZoneTemperature -Namespace "root/wmi" | Select-Object
CurrentTemperature
$Global:cpuProcessInformation = (Get-WmiObject -Class
Win32_Process).Count
$Global:cpuServiceInformation = (Get-WmiObject -Class
Win32_Service).Count

$Global:gpuInformation = Get-WmiObject -Class
Win32_VideoController | Select-Object -Property Name, DriverVersion,
CurrentRefreshRate, CurrentBitsPerPixel, CurrentHorizontalResolution,
CurrentVerticalResolution, AdapterRAM
$Global:gpuVRAMInformation = & nvidia-smi --query-
gpu=memory.total,memory.free,memory.used --format=csv,noheader,nounits
$Global:isNvidiaGPU = $false

$Global:ramInformation = Get-WmiObject -Class Win32_PhysicalMemory
| Select-Object -Property Capacity, Speed, Manufacturer, PartNumber, Tag
$Global:diskInformation = Get-CimInstance -ClassName
Win32_LogicalDisk | Select-Object VolumeName, Size, FreeSpace, FileSystem,
Compressed, Description, DriveType, MediaType

$osInfoThread = [powershell]::Create()
$osInfoThread.AddScript({
    $osInformation = Get-WmiObject -Class Win32_OperatingSystem | Select-
Object -Property Caption, Version, OSArchitecture, BuildNumber, InstallDate,
LastBootUpTime, LocalDateTime

    return [PSCustomObject] @{
        osIn = $osInformation
    }
})

$osInfoThread.RunspacePool = $runspacePool
$Global:osInfoThreadHandle = $osInfoThread.BeginInvoke()

$cpuInfoThread = [powershell]::Create()
$cpuInfoThread.AddScript({
    $cpuUsages = Get-WmiObject -Class Win32_PerfFormattedData_PerfOS_Processor
| Select-Object PercentIdleTime
    $cpuInformation = Get-WmiObject -Class Win32_Processor | Select-Object
VirtualizationFirmwareEnabled, LoadPercentage, NumberOfCores, ThreadCount,
Name, Manufacturer, CurrentClockSpeed

```

```

    $cpuTempInformation = Get-WmiObject MSAcpi_ThermalZoneTemperature -
Namespace "root/wmi" | Select-Object CurrentTemperature
    $cpuProcessInformation = (Get-WmiObject -Class Win32_Process).Count
    $cpuServiceInformation = (Get-WmiObject -Class Win32_Service).Count

    return [PSCustomObject] @{
        cpuUsages          = $cpuUsages
        cpuInformation      = $cpuInformation
        cpuTempInformation  = $cpuTempInformation
        cpuProcessInformation = $cpuProcessInformation
        cpuServiceInformation = $cpuServiceInformation
    }
})
$cpuInfoThread.RunspacePool = $runspacePool
$Global:cpuInfoThreadHandle = $cpuInfoThread.BeginInvoke()

$gpuInfoThread = [powershell]::Create()
$gpuInfoThread.AddScript({
    $gpuInformation = Get-WmiObject -Class Win32_VideoController | Select-
Object -Property Name, DriverVersion, CurrentRefreshRate, CurrentBitsPerPixel,
CurrentHorizontalResolution, CurrentVerticalResolution, AdapterRAM

    return [PSCustomObject] @{
        gpuInformation      = $gpuInformation
    }
})
$gpuInfoThread.RunspacePool = $runspacePool
$Global:gpuInfoThreadHandle = $gpuInfoThread.BeginInvoke()

$ramInfoThread = [powershell]::Create()
$ramInfoThread.AddScript({
    $ramInformation = Get-WmiObject -Class Win32_PhysicalMemory | Select-
Object -Property Capacity, Speed, Manufacturer, PartNumber, Tag

    return [PSCustomObject] @{
        ramInformation      = $ramInformation
    }
})
$ramInfoThread.RunspacePool = $runspacePool
$Global:ramInfoThreadHandle = $ramInfoThread.BeginInvoke()

$diskInformationThread = [powershell]::Create()
$diskInformationThread.AddScript({
    $diskInformation = Get-CimInstance -ClassName Win32_LogicalDisk | Select-
Object VolumeName, Size, FreeSpace, FileSystem, Compressed, Description,
DriveType, MediaType

    return [PSCustomObject] @{

```

```

        diskInformation      = $diskInformation
    }
})
$diskInformationThread.RunspacePool = $runspacePool
$Global:diskInformationThreadHandle =
$Global:diskInformationThread.BeginInvoke()

function Format-Bytes([int64] $Bytes) {
    if ($null -eq $Bytes) {
        return "N/A"
    }
    $suffixes = "B", "KB", "MB", "GB", "TB", "PB"
    $index = 0
    while ($Bytes -ge 1024 -and $index -lt $suffixes.Length) {
        $Bytes = $Bytes / 1024
        $index++
    }
    return "{0} {1}" -f ([math]::Floor($Bytes*100)/100), $suffixes[$index]
}

function Add-TitleLabel([string] $NameContentControl, [string] $Title) {
    $titleLabel = New-Object Windows.Forms.Label
    $titleLabel.Text = " $Title ▼"
    $titleLabel.Name = "$($NameContentControl)_Header"
    $titleLabel.Font = New-Object Drawing.Font("Arial", 10,
[Drawing.FontStyle]::Bold)
    $titleLabel.AutoSize = $true

    $titleLabel.add_Click({
        param ($titleLabel)

        $headerControlName = $titleLabel.Name -replace "_Header", ""
        $formContentPanel.Controls | ForEach-Object -Process {
            if ($_.Name -eq "$($headerControlName)_Combo") {
                $contentControl = $_
                if ($contentControl.Visible) {
                    $contentControl.Visible = $false
                } else {
                    $contentControl.Visible = $true
                }
            }
        }
        if ($_.Name -eq $headerControlName) {
            $contentControl = $_
            if ($contentControl.Visible) {
                $contentControl.Visible = $false
                $titleLabel.Text = $titleLabel.Text -replace " ▼", " ▲"
            } else {
                $contentControl.Visible = $true
            }
        }
    })
}

```

```

        $titleLbl.Text = $titleLbl.Text -replace "◀", "▼"
    }
}
})

$formContentPanel.Controls.Add($titleLbl)
}

function Get-OSInfo {
    if ($null -eq $osInfoLbl) {
        Add-TitleLabel -NameContentControl "osInfoLbl" -Title "OS Information"

        $Global:osInfoLbl = New-Object Windows.Forms.Label
        $Global:osInfoLbl.Name = "osInfoLbl"
        $formContentPanel.Controls.Add($Global:osInfoLbl)
        $Global:osInfoLbl.AutoSize = $true
    }

    $currentTime = Get-Date
    $osUptime = $currentTime -
[Management.ManagementDateTimeConverter]::ToDateTime($Global:osInformation.LastBootUpTime)
    $Global:osInfoLbl.Text =
        "    OS Name: $($Global:osInformation.Caption)`n" +
        "    Kernel Version: $($Global:osInformation.Version)`n" +
        "    Architecture:
$($Global:osInformation.OSArchitecture)`n" +
        "    Install Date:
$([Management.ManagementDateTimeConverter]::ToDateTime($Global:osInformation.InstallDate))`n" +
        "    Last Boot Time:
$([Management.ManagementDateTimeConverter]::ToDateTime($Global:osInformation.LastBootUpTime))`n" +
        "    Uptime: $($osUptime.Days) days, $($osUptime.Hours)
hours, $($osUptime.Minutes) minutes"
}

function Get-CPUInfo {
    if ($null -eq $Global:cpuInfoLbl) {
        Add-TitleLabel -NameContentControl "cpuInfoLbl" -Title "CPU
Information"

        $Global:cpuInfoSplitContainerPanel = New-Object
Windows.Forms.FlowLayoutPanel
        $Global:cpuInfoSplitContainerPanel.Name = "cpuInfoLbl"
        $Global:cpuInfoSplitContainerPanel.AutoScroll = $true
    }
}

```

```

        $Global:cpuInfoSplitContainerPanel.FlowDirection =
[System.Windows.Forms.FlowDirection]::LeftToRight
        $Global:cpuInfoSplitContainerPanel.WrapContents = $false
        $Global:cpuInfoSplitContainerPanel.Size = New-Object Drawing.Size
@(800, 2000)
        $formContentPanel.Controls.Add($Global:cpuInfoSplitContainerPanel)

        $Global:cpuInfoLbl = New-Object Windows.Forms.Label
        $Global:cpuInfoSplitContainerPanel.Controls.Add($Global:cpuInfoLbl)
        $Global:cpuInfoLbl.AutoSize = $true

        $Global:cpuInfoBarContainerPanel = New-Object
Windows.Forms.FlowLayoutPanel
        $Global:cpuInfoBarContainerPanel.AutoScroll = $true
        $Global:cpuInfoBarContainerPanel.FlowDirection =
[System.Windows.Forms.FlowDirection]::LeftToRight
        $Global:cpuInfoBarContainerPanel.WrapContents = $true
        $Global:cpuInfoBarContainerPanel.BackColor =
[Drawing.Color]::FromArgb(47, 51, 71)
        $Global:cpuInfoSplitContainerPanel.Controls.Add($Global:cpuInfoBarCont
ainerPanel)

        for ($i = 0; $i -lt $cpuLogicalThreadCount; $i++) {
            $cpuInfoGfx = New-Object Windows.Forms.Label
            $cpuInfoGfx.Name = "cpuInfoGfx_$i"
            $cpuInfoGfx.Enabled = $false
            $cpuInfoGfx.Margin = New-Object Windows.Forms.Padding @(5, 5, 5,
5)
            $cpuInfoGfx.Text = "$(100 -
$Global:cpuUsagePerCoreInformation[$i].PercentIdleTime)"
            $cpuInfoGfx.Width = 100
            $cpuInfoGfx.Height = 20
            $cpuInfoGfx.Add_Paint({
                param ($sender, $e)
                $graphics = $e.Graphics
                $rect = [System.Drawing.Rectangle]::new(1, 1, $sender.Width-2,
$sender.Height-2)
                $graphics.FillRectangle([System.Drawing.Brushes]::DarkGray,
$rect)
                $rect.Width = $rect.Width * ([int64]$sender.Text / 100)
                $graphics.FillRectangle([System.Drawing.Brushes]::Green,
$rect)
                $graphics.DrawString("Core $($sender.Name -replace
'cpuInfoGfx_', '') $($sender.Text)%", $sender.Font,
[System.Drawing.Brushes]::White, $rect.X + 10, $rect.Y + 1)
            })
            $Global:cpuInfoBarContainerPanel.Controls.Add($cpuInfoGfx)
        }

```

```

    }

    $Global:cpuInfoLbl.Text =
        "    Name: $($Global:cpuInformation.Name)`n" +
        "    Manufacturer:
$($Global:cpuInformation.Manufacturer)`n" +
        "    Base Clock Speed:
$([math]::Round($Global:cpuInformation.CurrentClockSpeed / 10)/100) GHz`n" +
        "    Cores: $($Global:cpuInformation.NumberOfCores)`n" +
        "    Threads: $($Global:cpuInformation.ThreadCount)`n" +
        "    Temperature:
$($Global:cpuTempInformation.CurrentTemperature /10) °C`n" +
        "    Usage: $($Global:cpuInformation.LoadPercentage)%`n" +
        "    Processes Running:
$($Global:cpuProcessInformation)`n" +
        "    Services Running: $($Global:cpuServiceInformation)`n"
+
        "    Virtualization:
$($Global:cpuInformation.VirtualizationFirmwareEnabled ? ('Enabled') :
('Disabled'))`n"
    $Global:cpuInfoSplitContainerPanel.Width = $formContentPanel.Width - 32
    $Global:cpuInfoSplitContainerPanel.Height = $Global:cpuInfoLbl.Height + 10
    $Global:cpuInfoBarContainerPanel.Width = ($formContentPanel.Width -
$Global:cpuInfoLbl.Width - 48)
    $Global:cpuInfoBarContainerPanel.Height =
$Global:cpuInfoSplitContainerPanel.Height-6

    for ($i = 0; $i -lt $cpuLogicalThreadCount; $i++) {
        $cpuInfoGfx = $Global:cpuInfoBarContainerPanel.Controls[$i]
        $cpuInfoGfx.Text = "$(100 -
$Global:cpuUsagePerCoreInformation[$i].PercentIdleTime)"
    }
}

function Get-GPUInfo {
    if ($null -eq $Global:gpuInfoLbl) {
        Add-TitleLabel -NameContentControl "gpuInfoLbl" -Title "GPU
Information"

        $Global:gpuInfoLbl = New-Object Windows.Forms.Label
        $Global:gpuInfoLbl.Name = "gpuInfoLbl"
        $formContentPanel.Controls.Add($gpuInfoLbl)
        $Global:gpuInfoLbl.AutoSize = $true
    }

    $text = ""
    $Global:gpuInformation | ForEach-Object -Process {
        $gpuName = $_.Name

```

```

$gpuVRAM = ""

if ($gpuName.Contains("NVIDIA")) {
    $nvidiaSmiOutput = $Global:gpuVRAMInformation
    $gpuVRAM = "VRAM: $( [math]::Round($nvidiaSmiOutput.Split(",")[0]
/ 1024, 2) ) GB"
} else {
    $gpuVRAM = "VRAM: $(Format-Bytes -Bytes $_.AdapterRAM)"
}

$text = $text +
    "    Name: $($_.Name)`n" +
    "    Driver Version: $($_.DriverVersion)`n" +
    "    Refresh Rate: $($_.CurrentRefreshRate) Hz`n" +
    "    Bits Per Pixel: $($_.CurrentBitsPerPixel)`n" +
    "    Resolution: $($_.CurrentHorizontalResolution) x
$($_.CurrentVerticalResolution)`n" +
    "    $gpuVRAM`n"
}
$Global:gpuInfoLbl.Text = $text
}

function Get-RAMInfo {
    if ($null -eq $Global:ramInfoLbl) {
        Add-TitleLabel -NameContentControl "ramInfoLbl" -Title "RAM
Information"

        $Global:ramInfoLbl = New-Object Windows.Forms.Label
        $Global:ramInfoLbl.Name = "ramInfoLbl"
        $formContentPanel.Controls.Add($ramInfoLbl)
        $Global:ramInfoLbl.AutoSize = $true
    }
    $totalRam = 0

    $text = ""
    $Global:ramInformation | ForEach-Object -Process {
        $totalRam += $_.Capacity
        $text = $text +
            "    $_.Tag`n" +
            "    Part Number: $_.PartNumber`n" +
            "    Manufacturer: $_.Manufacturer`n" +
            "    Speed: $_.Speed MHz`n" +
            "    Capacity: $(Format-Bytes -Bytes $_.Capacity)`n"
    }
    $text = $text + "    Total Capacity: $(Format-Bytes -Bytes $totalRam)`n"
    $Global:ramInfoLbl.Text = $text
}

```

```

function Get-DiskInfo {
    if ($null -eq $Global:diskInfoLbl) {
        Add-TitleLabel -NameContentControl "diskInfoLbl" -Title "Disk
Information"

        $Global:diskInfoLbl = New-Object Windows.Forms.Label
        $Global:diskInfoLbl.Name = "diskInfoLbl"
        $formContentPanel.Controls.Add($diskInfoLbl)
        $Global:diskInfoLbl.AutoSize = $true
    }

    $text = ""
    $Global:diskInformation | ForEach-Object -Process {
        if ($null -eq $_.VolumeName) {
            return
        }
        $totalDiskCapacity += $_.Size
        $text = $text +
            "    $(($_.VolumeName -eq '' ? ('Local Disk') : ($_.VolumeName)))
            $(($_.Name))`n" +
            "        Description: $(($_.Description))`n" +
            "        File System: $(($_.FileSystem)) Compressed: $(($_.Compressed))`n"
+
            "        Size: $(Format-Bytes -Bytes $_.Size)`n" +
            "        Free Space: $(Format-Bytes -Bytes $_.FreeSpace)`n"
        }
    $text = $text + "    Total Capacity: $(Format-Bytes -Bytes
$totalDiskCapacity)`n"
    $Global:diskInfoLbl.Text = $text
}

function UserInfoComboBox {
    $users = Get-AllUsers
    Add-TitleLabel -NameContentControl "userInfoBox" -Title "User Information"
    $Global:userInfoBox = New-Object Windows.Forms.ComboBox
    $userInfoBox.AutoSize = $true
    $userInfoBox.DropDownStyle = "DropDownList"
    $userInfoBox.Items.AddRange($users)
    $userInfoBox.Name = "userInfoBox_Combo"
    $formContentPanel.Controls.Add($userInfoBox)

    $userInfoBox.add_SelectedIndexChanged({
        param ($sender, $e)
        Get-UserInfo -selectedUser $sender.SelectedItem
    })
}

```

```

    $currentUser =
[System.Security.Principal.WindowsIdentity]::GetCurrent().Name.Split("\")[-1]
    $defaultIndex = $userInfoBox.Items.IndexOf($currentUser)
    if ($defaultIndex -ge 0) {
        $userInfoBox.SelectedIndex = $defaultIndex
    }
}

function Get-AllUsers {
    $users = @()
    $context = New-Object
System.DirectoryServices.AccountManagement.PrincipalContext([System.DirectoryS
ervices.AccountManagement.ContextType]::Machine)
    $searcher = New-Object
System.DirectoryServices.AccountManagement.UserPrincipal($context)
    $query = New-Object
System.DirectoryServices.AccountManagement.PrincipalSearcher($searcher)

    foreach ($user in $query.FindAll()) {
        $users += $user.SamAccountName
    }
    return $users
}

function Get-UserInfo {
    param (
        [string]$selectedUser
    )

    if ($null -eq $userInfoLbl) {
        $Global:userInfoLbl = New-Object Windows.Forms.label
        $userInfoLbl.AutoSize = $true
        $userInfoLbl.Name = "userInfoBox"
        $formContentPanel.Controls.Add($userInfoLbl)
    }
    if (-not $selectedUser) {return}
    $context = New-Object
System.DirectoryServices.AccountManagement.PrincipalContext([System.DirectoryS
ervices.AccountManagement.ContextType]::Machine)
    $user =
[System.DirectoryServices.AccountManagement.UserPrincipal]::FindByIdentity($co
ntext, $selectedUser)
    if ($null -eq $user) {
        $userInfoLbl.Text = "Unknown."
        return
    }
    $userSID = $user.Sid.Value

```

```

# check if admin
$group =
[System.DirectoryServices.AccountManagement.GroupPrincipal]::FindByIdentity($context, "Administrators")
$isAdmin = $group.Members | Where-Object {
    $_.SamAccountName -eq $selectedUser
} | ForEach-Object {
    $true
}
$isAdmin = $isAdmin -eq $true ? ("Yes") : ("No")

$userProfilePath = Get-WmiObject Win32_UserProfile | Where-Object {
    $_.SID -eq $userSID
} | Select-Object -ExpandProperty LocalPath

$lastPwChange = $user.LastPasswordSet

$userInfoLbl.Text =
    "    User: $selectedUser`n" +
    "    SID: $userSID`n" +
    "    Is Admin: $isAdmin`n" +
    "    Profile Path: $userProfilePath`n" +
    "    Last Password Change: $lastPwChange`n"
}

function Get-NetworkInfo {
    if ($null -eq $ntwkInfoLbl) {
        Add-TitleLabel -NameContentControl "ntwkInfoLbl" -Title "Network
Information"

        $Global:ntwkInfoLbl = New-Object Windows.Forms.label
        $Global:ntwkInfoLbl.Name = "ntwkInfoLbl"
        $formContentPanel.Controls.Add($ntwkInfoLbl)
        $ntwkInfoLbl.AutoSize = $true
    }

    $ntwkInfo = Get-WmiObject -Class Win32_NetworkAdapterConfiguration |
Select-Object -Property Description, MACAddress, IPAddress, IPSubnet,
DefaultIPGateway, DNSDomain, DNSHostName, DHCPEnabled, DHCPServer
    $ntwkInfo | ForEach-Object -Process {
        if ($null -eq $_.DNSDomain) {
            return
        }
        $ntwkInfoLbl.Text = $ntwkInfoLbl.Text +
            "    $(($_.Description))`n" +
            "        MAC Address: $(($_.MACAddress))`n" +
            "        IP Address: $(($_.IPAddress[0]))`n" +
            "        Subnet Mask: $(($_.IPSubnet))`n" +

```

```

        "        Default Gateway: $($_.DefaultIPGateway)`n" +
        "        DNS Domain: $($_.DNSDomain)`n" +
        "        DNS Host Name: $($_.DNSHostName)`n" +
        "        DHCP Enabled: $($_.DHCPEnabled)`n" +
        "        DHCP Server: $($_.DHCPServer)`n"
    }
}

function IsPublicFirewallEnabled {
    $firewall = New-Object -ComObject HNetCfg.FwMgr
    $profile = $firewall.LocalPolicy.CurrentProfile
    return ($profile.Type -eq 4) -and $profile.FirewallEnabled
}

function IsPrivateFirewallEnabled {
    $firewall = New-Object -ComObject HNetCfg.FwMgr
    $profile = $firewall.LocalPolicy.CurrentProfile
    return ($profile.Type -eq 2) -and $profile.FirewallEnabled
}

function IsDomainFirewallEnabled {
    $firewall = New-Object -ComObject HNetCfg.FwMgr
    $profile = $firewall.LocalPolicy.CurrentProfile
    return ($profile.Type -eq 1) -and $profile.FirewallEnabled
}

function Get-SecurityInfo {
    if ($null -eq $secInfoLbl) {
        Add-TitleLabel -NameContentControl "secInfoLbl" -Title "Insecurity
Information"

        $Global:secInfoLbl = New-Object Windows.Forms.Label
        $Global:secInfoLbl.Name = "secInfoLbl"
        $secInfoLbl.AutoSize = $true
        $formContentPanel.Controls.Add($secInfoLbl)
    }

    $isRealTimeProtEnabled = ((Get-MpPreference).DisableRealtimeMonitoring -eq
>false) ? ("Enabled") : ("Disabled")
    $isAutoSampleSubmissionEnabled = ((Get-MpPreference).SubmitSamplesConsent)
? ("Enabled") : ("Disabled")
    $isPublicFirewallEnabled = (IsPublicFirewallEnabled) ? ("Enabled") :
("Disabled")
    $isPrivateFirewallEnabled = (IsPrivateFirewallEnabled) ? ("Enabled") :
("Disabled")
    $isDomainFirewallEnabled = (IsDomainFirewallEnabled) ? ("Enabled") :
("Disabled")

```

```

    $isCalculatorInstalled = (Test-Path -Path
"$($env:SystemRoot)\System32\calc.exe") ? ("Installed") : ("Not Installed")

    $secInfoLbl.Text =
        "    Real Time Protection: $isRealTimeProtEnabled`n" +
        "    Auto Sample Submission:
$isAutoSampleSubmissionEnabled`n" +
        "    Public Firewall: $isPublicFirewallEnabled`n" +
        "    Private Firewall: $isPrivateFirewallEnabled`n" +
        "    Domain Firewall: $isDomainFirewallEnabled`n" +
        "    Calculator: $isCalculatorInstalled`n"
}

function Get-ProcessList() {
    return Get-Process | Select-Object Id, ProcessName, CPU, Path
}

function Show-ProcessListCheckbox() {
    if ($processListCheckbox.Checked) {
        $formContentPanel.Visible = $false
        $processListView.Visible = $true
        Show-ProcessList
    }
}

function Show-ProcessList() {
    $processListView.Items.Clear()
    $procs = Get-ProcessList

    foreach ($proc in $procs) {
        $itm = New-Object System.Windows.Forms.ListViewItem($proc.Id)
        $itm.SubItems.Add($proc.ProcessName)
        $itm.SubItems.Add($proc.CPU)
        $itm.SubItems.Add($proc.Path)
        $processListView.Items.Add($itm)
    }
}

function TerminateProcess() {
    if ($processListView.SelectedItems.Count -gt 0) {
        $procId = $processListView.SelectedItems[0].Text
        Stop-Process -Id $procId -Force -ErrorAction SilentlyContinue
        Show-ProcessList
    }
}

function Render-ProcessList() {
    $Global:processListCheckbox = New-Object Windows.Forms.CheckBox

```

```

$processListCheckbox.Text = "Show Processes"
$processListCheckbox.ForeColor = [Drawing.Color]::White
$processListCheckbox.BackColor = [Drawing.Color]::FromArgb(36, 39, 58)
$processListCheckbox.Add_CheckStateChanged({
    Show-ProcessListCheckbox
})
$formContentPanel.Controls.Add($processListCheckbox)

$Global:processListView = New-Object Windows.Forms.ListView
$processListView.View = "Details"
$processListView.FullRowSelect = $true
$processListView.GridLines = $true
$processListView.Visible = $false
$processListView.Columns.Add("PID", 50)
$processListView.Columns.Add("Name", 100)
$processListView.Columns.Add("CPU", 50)
$processListView.Columns.Add("Path", 200)
$form.Controls.Add($processListView)

$Global:termBtn = New-Object Windows.Forms.Button
$termBtn.Text = "Terminate Process"
$termBtn.ForeColor = [Drawing.Color]::White
$termBtn.BackColor = [Drawing.Color]::FromArgb(36, 39, 58)
$termBtn.Add_Click({
    TerminateProcess
})
$formContentPanel.Controls.Add($termBtn)
}

function Refresh-Info-From-Threads {
    if ($true -eq $osInfoThreadHandle.IsCompleted) {
        $osInfo = $osInfoThread.EndInvoke($osInfoThreadHandle)
        $Global:osInformation = $osInfo.osIn
        Get-OSInfo
        $Global:osInfoThreadHandle = $osInfoThread.BeginInvoke()
    }
    if ($true -eq $cpuInfoThreadHandle.IsCompleted) {
        $cpuInfo = $cpuInfoThread.EndInvoke($cpuInfoThreadHandle)
        $Global:cpuInformation = $cpuInfo.cpuInformation
        $Global:cpuTempInformation = $cpuInfo.cpuTempInformation
        $Global:cpuUsagePerCoreInformation = $cpuInfo.cpuUsages
        $Global:cpuServiceInformation = $cpuInfo.cpuServiceInformation
        $Global:cpuProcessInformation = $cpuInfo.cpuProcessInformation
        Get-CPUInfo
        $Global:cpuInfoThreadHandle = $cpuInfoThread.BeginInvoke()
    }
    if ($true -eq $gpuInfoThreadHandle.IsCompleted) {
        $gpuInfo = $gpuInfoThread.EndInvoke($gpuInfoThreadHandle)
    }
}

```

```

        $Global:gpuInformation = $gpuInfo.gpuInformation
        Get-GPUInfo
        $Global:gpuInfoThreadHandle = $gpuInfoThread.BeginInvoke()
    }
    if ($true -eq $ramInfoThreadHandle.IsCompleted) {
        $ramInfo = $ramInfoThread.EndInvoke($ramInfoThreadHandle)
        $Global:ramInformation = $ramInfo.ramInformation
        Get-RAMInfo
        $Global:ramInfoThreadHandle = $ramInfoThread.BeginInvoke()
    }
    if ($true -eq $diskInformationThreadHandle.IsCompleted) {
        $ramInfo =
$diskInformationThread.EndInvoke($diskInformationThreadHandle)
        $Global:diskInformation = $ramInfo.diskInformation
        Get-DiskInfo
        $Global:diskInformationThreadHandle =
$diskInformationThread.BeginInvoke()
    }
}

$form.Add_KeyDown({
    if ($_.KeyCode -eq "Escape") {
        $processListCheckbox.Checked = $false
        $formContentPanel.Visible = $true
    }
})

Get-OSInfo
Get-CPUInfo
Get-GPUInfo
Get-RAMInfo
Get-DiskInfo
UserInfoComboBox
Get-NetworkInfo
Get-SecurityInfo
Render-ProcessList

$refreshInfoTimer = New-Object System.Windows.Forms.Timer
$refreshInfoTimer.Interval = 1000
$refreshInfoTimer.Add_Tick({
    Refresh-Info-From-Threads
})
$refreshInfoTimer.Start()

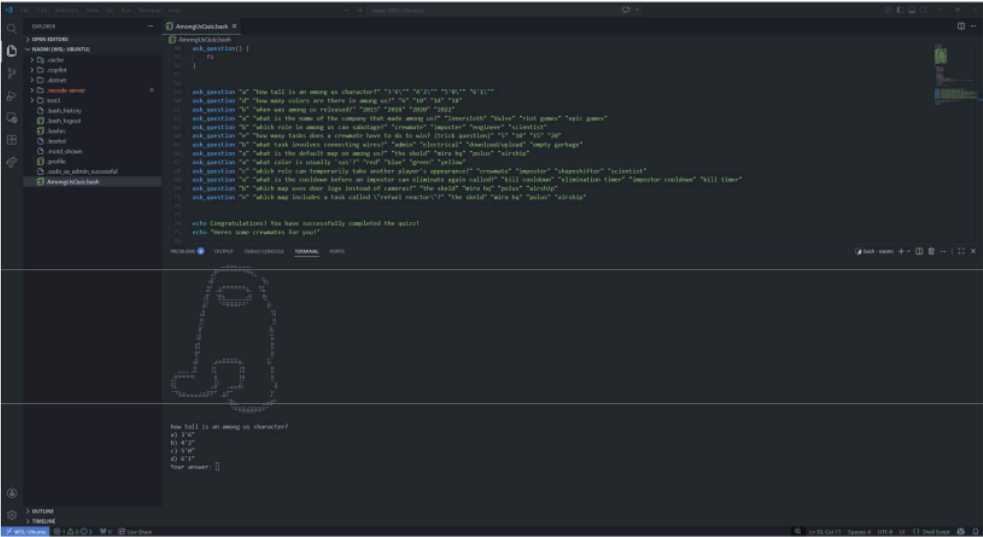
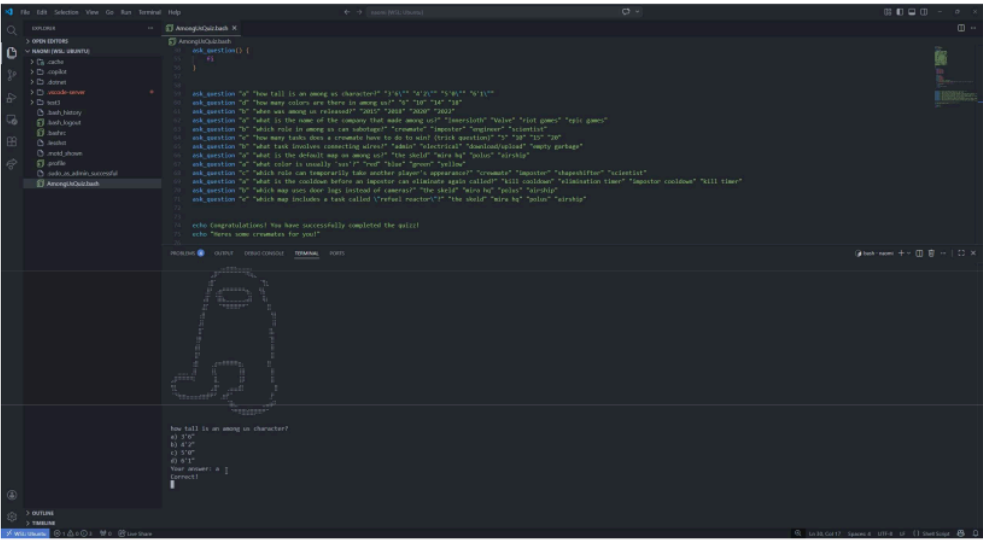
[void]$form.ShowDialog()

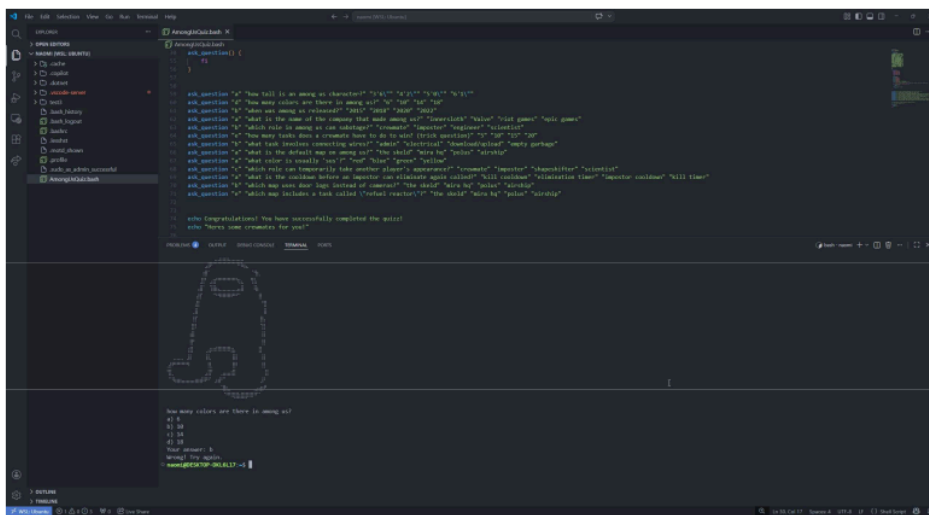
```

Task 5

The bash script I have created begins with shebang (`#!/bin/bash`) which specifies the interpreter. I enabled strict error handling using `set -euo pipefail`, which improves my scripts reliability. The `shopt -s nocasematch` option allows my script to be case insensitive input comparison, which is needed if I did not want to worry about the user entering `A` instead of `a` for the right answer in the quiz. I defined functions using standard syntax (`ask_question(){}`), which enabled me to reuse code blocks. I used a `cat << 'EOF'` to output a multi line ASCII art to add to the script. I assigned variables without spaces and accessed the variables via the `$` prefix. I also used control flow structures, for example conditional statements and loops, to manage the progression of the quiz aswell as handling if the answer from the user is correct or incorrect.

Screenshots





Script

```
#!/bin/bash
set -euo pipefail
shopt -s nocasematch
```

```
print_amongus() {  
    cat << 'EOF'
```

EOF

}

```
ask_question() {
```

```

local correct="$1"
local question="$2"
local answer_a="$3"
local answer_b="$4"
local answer_c="$5"
local answer_d="$6"

clear
print_amongus
echo "$question"
echo "a) $answer_a"
echo "b) $answer_b"
echo "c) $answer_c"
echo "d) $answer_d"
read -p "Your answer: " given_answer
if [[ "$given_answer" == "fortnite" ]]; then
    echo "Don't go down that path, trust me. Been there, done that. just
stick to the quiz, ok?"
fi
if [[ "$given_answer" == "$correct" ]]; then
    echo "Correct!"
    sleep 2
else
    echo "Wrong! Try again."
    exit 0
fi
}

ask_question "a" "how tall is an among us character?" "3'6\" "4'2\" "5'0\" "
6'1\"
ask_question "d" "how many colors are there in among us?" "6" "10" "14" "18"
ask_question "b" "when was among us released?" "2015" "2018" "2020" "2022"
ask_question "a" "what is the name of the company that made among us?"
"Innersloth" "Valve" "riot games" "epic games"
ask_question "b" "which role in among us can sabotage?" "crewmate" "imposter"
"engineer" "scientist"
ask_question "e" "how many tasks does a crewmate have to do to win? (trick
question)" "5" "10" "15" "20"
ask_question "b" "what task involves connecting wires?" "admin" "electrical"
"download/upload" "empty garbage"
ask_question "a" "what is the default map on among us?" "the skeld" "mira hq"
"polus" "airship"
ask_question "a" "what color is usually 'sus'?" "red" "blue" "green" "yellow"
ask_question "c" "which role can temporarily take another player's
appearance?" "crewmate" "imposter" "shapeshifter" "scientist"
ask_question "a" "what is the cooldown before an impostor can eliminate again
called?" "kill cooldown" "elimination timer" "impostor cooldown" "kill timer"

```

```
ask_question "b" "which map uses door logs instead of cameras?" "the skeld"
"mira hq" "polus" "airship"
ask_question "e" "which map includes a task called \"refuel reactor\"?" "the
skeld" "mira hq" "polus" "airship"

echo Congratulations! You have successfully completed the quizz!
echo "Heres some crewmates for you!"

while true; do
    sleep 0.5
    printf "%s" "🌀 "
done
```

Conclusion

1. Both systems have similar core functions, but with different implementations.
Both Microsoft Windows and Linux systems give the user the same fundementle operating system functions, like process scheduling, memory management, file system management, networking and user control, but they are implemented differently. On Windows, it is GUI driven, using menus and buttons etc, with a CLI alternative if you want more control, however on Linux it is CLI first, you have a lot more direct control with CLI tools, which means on Linux, you are offered more scriptable control, whereas on Windows the user is offered more accessibility and usability.
2. CLI tools show system-level control differences.
In Linux, the commands follow a text-stream pipeline modle, which means the Linux CLI modle is more composable and flexible for chaining commands, however on Windows; Powershell provides object-based output, not just strings, meaning that Powershell is more structured, and it is integrated deeply with the operating system by utilizing .NET.
3. Scripting capabilities
Powershell scripts are very good for system administration, and it is a very versatile language, especially in enterprise environments. Whereas Bash for Linux are lightweight and very fast, ideal for automation. This shows that both systems have good scripting languages which are fast. However Powershell is more complex and more powerful as it can integrate with Windows very well, whereas Bash is more simple and portable. This difference can be seen in the scripts provided above.

4. Overall differences

Windows compared with Linux, Linux is the more “advanced” operating system, aimed for people who are more technically inclined, whereas Windows is more user-friendly. Windows is a more GUI heavy operating system, providing more applications and tools with a GUI to manage and use tools, whereas on Linux, the tools are mainly CLI. An example of this, would be that you can change passwords on Windows for users by doing so in the lusrmgr.msc. However on Linux, you do so using root in the command line interface (CLI).

Windows as an operating system is less configurable while Linux is highly configurable, with lots of theming options and style options, however on Windows it is a lot harder for custom theming.

References

[1] Microsoft, “Powershell documentation”, Microsoft Learn.

<https://learn.microsoft.com/en-us/powershell>

[2] Microsoft, “What is Powershell”, Microsoft Learn. <https://learn.microsoft.com/en-gb/powershell/scripting/overview?view=powershell-7.6>

[3] DevDocs, “Bash Documentation”, <https://devdocs.io/bash/>

[4] Microsoft, “Windows Commands”, <https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands>

[5] Arch Linux, “Arch Linux Commands”,

<https://wiki.archlinux.org/title/Category:Commands>

Appendices

Different Types of OS

General Purpose OS's

General purpose operating systems are designed to be able to perform many tasks for everyday users. They give a graphical user interface (GUI) which makes it more accessible to all types of users. They manage hardware resources, can run applications and also allow users to multitask. General purpose operating systems are commonly used on desktop computers and laptops for activities like browsing the web, gaming,

programming and office working. Examples of general purpose operating systems include: Linux, Windows and macOS.

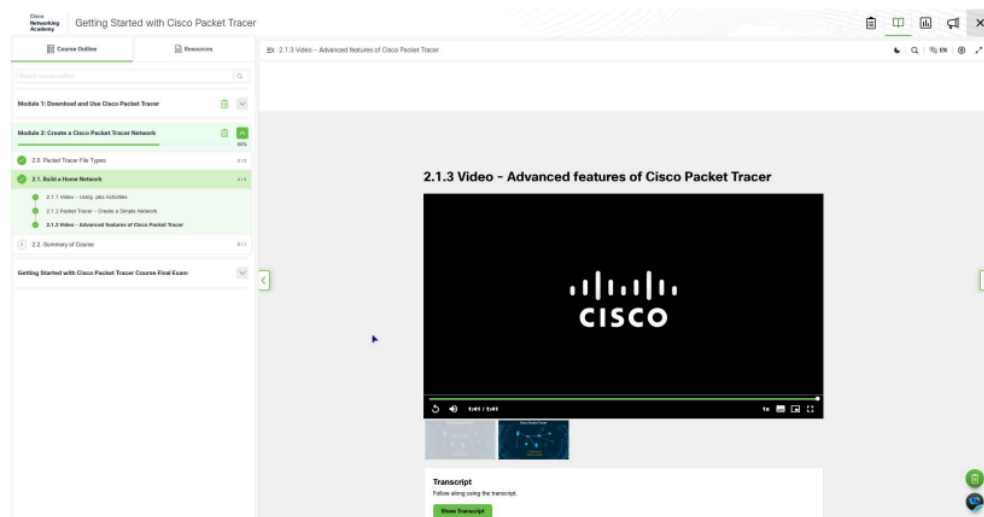
Mobile Operating Systems

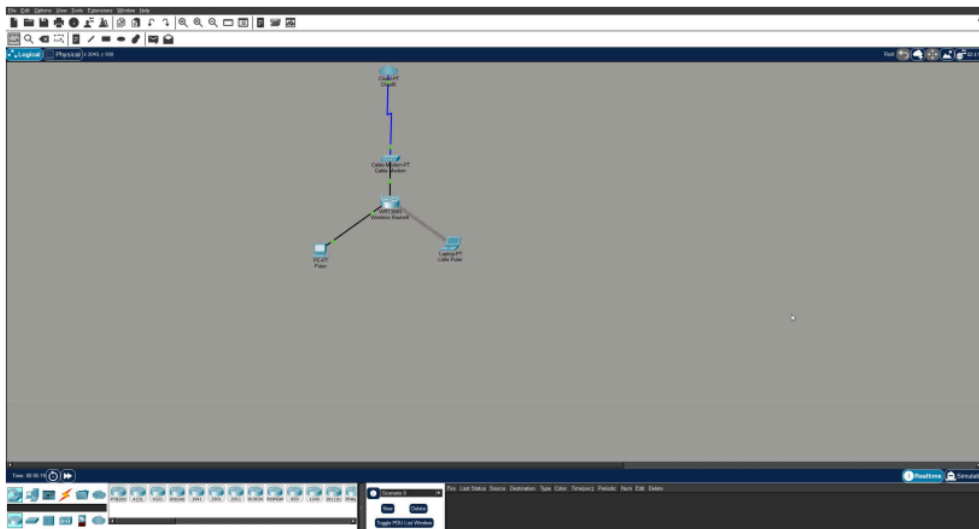
Mobile operating systems are designed specifically for mobile phones. They are optimized differently to general purpose operating systems as they need to manage things like touchscreen, battery efficiency, wireless communications and mobile applications. They also manage features like cameras and GPS's. Some examples of mobile OS's are: Android, iOS and GrapheneOS.

Network Operating Systems

Network operating systems are designed to support and manage computers that are connected across a network. They allow users to utilize netshares which allow for the transfer of files. They also allow sharing of different resources like printers, email services, databases and media across multiple devices. These systems are used commonly in schools, buisnesses and servers, because they provide centralized administration, user management and security. Examples of some network operating systems include: Ubuntu Server, Windows Server and Unix.

Proof of Engagement with an Online Course





Getting Started with Cisco Packet Tracer

1.0 Install Cisco Packet Tracer

1.0.1 Video - Welcome to Getting Started with Cisco Packet Tracer

Progress loaded successfully

Cisco Packet Tracer

Networking
CISCO Academy

Cisco
Networking
Academy

Getting Started with Cisco Packet Tracer

🏠 📖 📝 🔍 ⌵

Assignment Content

🔍 🏠 📖 📝 🔍 ⌵

📌 Getting Started with Cisco Packet Tracer Course ...

📅 EX DATE
27 September 2020, 11:58 AM

📅 AVAILABLE FROM
29 January 2020, 01:10 AM

📊 POINTS
100

📂 CATEGORY
Assessment

📌 End of Course Survey (Assignment Submitted)

📌 Course Final Exam

📌 Getting Started with Cisco Packet Tracer Course Final Exam

📌 Course Final Exam

EX Course Final Exam

🔍 🏠 📖 📝 🔍 ⌵

🔍 🏠 📖 📝 🔍 ⌵

80%

You've scored 80%.
Congratulations, you have passed the exam.

👏

OS Assignment 1

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

63/100

Total: 63

**1. Engagement
with Literature Skills:**

- No use of citations. + Well formatted IEEE reference list. 20 4

- could include 1/2 books/journal papers.

**2. Knowledge
and Understanding Skills: 4e**

Key functions: + 20 14

Overview of 1/2 key OS functions documented. + Screenshots showing where some of the functions can be seen/monitored/managed are documented. - could be extended to briefly include some additional areas.

Win Commands: + Good overview of relevant commands that demonstrates

an understanding of the language.

Bash Commands: + Good overview of some relevant commands that demonstrates an understanding of the language.

Win Scripts: + Good overview of the syntax that demonstrates an understanding of the scripting aspects of the language. - could be extended to briefly describe 1 additional syntax area (Switch)

Bash Scripts: + Good overview of the syntax that demonstrates a clear understanding of the scripting aspects of the language. - could be extended to briefly describe 1/2 additional syntax areas.

3. Cognitive and Intellectual Skills: 4+4+2

+ Evidence of some analysis and understanding of OS functions.

+ Screenshots showing where some of the functions can be seen/monitored/managed are included in the analysis. – could be extended to briefly include some additional areas.

+ Conclusion.

4. Practical Application Skills: 10e:

1. **Win Commands:** + Good use of relevant Win. language commands.

2. Bash Commands: + Good use of some relevant Bash commands.

3. Win Scripts: + Good use of Win. scripts to demonstrate applied knowledge of the core syntax. Areas. - could be extended to briefly use 1 additional syntax area (Switch)

4. Bash Scripts: + Good use of Bash scripting to demonstrate applied knowledge of the core syntax areas. - could be extended to briefly use 1/2 additional syntax areas.

5. Transferable

Skills for Life and Professional Practice:

4+2+2+1+2

+ Report 10 7
format, structure, and presentation quite good.

- Each report section could be briefly introduced.

- Toc & report content should be separated onto separate pages

- Screenshots could be cropped

+ Tutorial Activity 1 on OS

Types: relevant supplementary information on OS types is included in the Appendix of the report.

+ Evidence of engagement with a relevant CPD course documented in the appendix.

- Lecture activity evidence not documented.

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

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42
