Mohammed Jahangir Alom
Student Number – R00144214
Module – Operating Systems Fundamental
Class – Software Development
Lecturer – Mr. Karl O`Connell
Assessment 2 Part 2
Date – 13/03/2018

Question 1

Show your first shell script program to run using a bash shell
.Note: As with all QUESTIONS, do not forget to record your progress by copying your progress to your report.
Please ask your lab lecturer if you have any questions or require assistance.
•Open a terminal.
•Go to the Part2directory (i.e. type cd Part2).
•Create an empty file called MyFirstScript by typing touch MyFirstScript. Type ls –l and note the file size of zero bytes.
•Edit the file MyFirstScript with nano. Type nano MyFirstScript and then enter the following data for the file (caution: the script is case sensitive, so small 'e' for echo....) :echoThis is the start of my simple first script program
echo We will printout the date and time here $(date)
echo We will look at the files in the current directory, note file size ls –l
•Run the script in a bash shell by typing bash MyFirstScript.
•In your own words, briefly describe the script MyFirstScript, referring to the commands of echo, date and ls

```
mohammed@mohammed-virtual-machine: ~/OperatingSystem/Assignment2/Part2
File  Edit  View  Search  Terminal  Help
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2$ cd Part2
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ clear

mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ touch MyFirstScript
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ ls
A2Part2.pdf  MohammedAlomA2Part2.odt  MyFirstScript
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ ls -l
total 640
-rw-rw-r-- 1 mohammed mohammed 632626 Mar 12 13:28 A2Part2.pdf
-rw-rw-r-- 1 mohammed mohammed  17403 Mar 12 13:37 MohammedAlomA2Part2.odt
-rw-rw-r-- 1 mohammed mohammed      0 Mar 12 13:45 MyFirstScript
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ nano MyFirstScript
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ bash MyFirstScript
This is the start of my simple first script program
We will printout the date and time here Mon Mar 12 13:49:55 GMT 2018
We will look at the files in the current directory, note file size ls -l
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$
```

Brief descrition of the file of MyFirstScript:

echo *is a built-in command in the bash.* A command is an instruction telling a computer to do something. An argument is input data for a command. Standard output is the display screen by default, but it can be redirected to a file, printer, etc.
It is not necessary to surround the strings with quotes, as it does not affect what is written on the screen. If quotes (either single or double) are used, they are not repeated on the screen.

date – will show the system time and date information in the screen when we put command like $(date)
ls – l command will show all the directory and fies in the terminal in the specific directory.

Question 2
top, ps, pstree

•Open a terminal (we consider this terminal 1).
•Run the command ping www.cit.ie.

```
mohammed@mohammed-virtual-machine:~$ ping www.cit.ie
PING www.cit.ie (54.72.5.20) 56(84) bytes of data.
```

•Open a second terminal (we consider this terminal 2).
•Use the ps aux command to look at the processes. Clearly indicate the process id PID of the ping program running in terminal 1.

```
mohammed@mohammed-virtual-machine:~$ ps aux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.2 119792  4292 ?        Ss   12:34   0:07 /sbin/init spl
root          2  0.0  0.0      0     0 ?        S    12:34   0:00 [kthreadd]
root          4  0.0  0.0      0     0 ?        S<   12:34   0:00 [kworker/0:0H]
root          6  0.0  0.0      0     0 ?        S<   12:34   0:00 [mm_percpu_wq]
root          7  0.0  0.0      0     0 ?        S    12:34   0:01 [ksoftirqd/0]
root          8  0.3  0.0      0     0 ?        S    12:34   1:25 [rcu_sched]
```

```
root      12086  0.0  0.0      0     0 ?        S    20:00   0:00 [kworker/u256:
mohammed  12087  0.0  0.0  14948  1796 pts/0    S+   20:00   0:00 ping www.cit.i
root      12118  0.1  0.0      0     0 ?        S    20:02   0:00 [kworker/1:1]
mohammed  12206  0.0  0.1  37364  3236 pts/1    R+   20:06   0:00 ps aux
```

•Now use the grep command to make searching the output of ps easier (filter the output).Type the command ps aux | grep ping. This command sends the output from ps aux to the grep program. The grep program searches this output and displays the lines with the word ping.

```
mohammed@mohammed-virtual-machine:~$ ps aux | grep ping
mohammed  12087  0.0  0.0  14948  1796 pts/0    S+   20:00   0:00 ping www.cit.ie
mohammed  12311  0.0  0.0  14224   924 pts/1    S+   20:08   0:00 grep --color=auto ping
```

•Run the top command. Describe at least 5 of the items displayed on the screen. Note: type 'q' to exit the top program.
•Close both terminals. The programs/commands the terminals were running will close (i.e. the terminal parent process is stopped, so then the command or program (i.e. child process) will be stopped).

```
mohammed@mohammed-virtual-machine:~$ top

top - 20:09:31 up  7:35,  1 user,  load average: 0.29, 0.37, 0.60
Tasks: 219 total,   1 running, 218 sleeping,   0 stopped,   0 zombie
%Cpu(s):  4.1 us,  3.8 sy,  0.0 ni, 91.9 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
KiB Mem :  2018088 total,   182412 free,  1371584 used,   464092 buff/cache
KiB Swap:  2094076 total,  1831932 free,   262144 used.   400856 avail Mem

   PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
  1880 mohammed  20   0 2148356 425960  38712 S  11.0 21.1  44:15.78 cinnamon
   930 root      20   0  515804  83372  39224 S   5.6  4.1  12:02.69 Xorg
     8 root      20   0       0      0      0 S   1.0  0.0   1:26.22 rcu_sched
   810 mongodb   20   0  552156   3120   2508 S   1.0  0.2   2:49.94 mongod
 12342 mohammed  20   0  551976  31852  25928 S   0.7  1.6   0:00.80 gnome-screensho
   902 root      20   0   19584   1736   1604 S   0.3  0.1   0:04.83 irqbalance
  2068 mohammed  20   0  456248  22804  17624 S   0.3  1.1   0:50.22 cinnamon-screen
 12008 mohammed  20   0  532312  35324  26232 S   0.3  1.8   0:08.81 gnome-terminal-
 12306 root      20   0       0      0      0 S   0.3  0.0   0:00.12 kworker/1:0
 12336 mohammed  20   0   41804   3696   3076 R   0.3  0.2   0:00.10 top
     1 root      20   0  119792   4292   2912 S   0.0  0.2   0:07.94 systemd
     2 root      20   0       0      0      0 S   0.0  0.0   0:00.07 kthreadd
```

This first line indicates in order:

- current time (20:09:31)
- uptime of the machine (up 7:35)
- users sessions logged in (1 users)
- average load on the system (load average: 0.09, 0.37, 0.60) the 3 values refer to the last minute, five minutes and 15 minutes.

The second row gives the following information:

- Processes running in totals (219 total)
- Processes running (1 running)
- Processes sleeping (218 sleeping)
- Processes stopped (0 stopped)
- Processes waiting to be stoppati from the parent process (0 zombie)

The third line indicates how the cpu is used. If you sum up all the percentages the total will be 100% of the cpu. Let's see what these values indicate in order:

- Percentage of the CPU for user processes (4.1%**us**)
- Percentage of the CPU for system processes (3.8%**sy**)
- Percentage of the CPU processes with priority upgrade *nice*(0.0%**ni**)
- Percentage of the CPU not used (91.9%**id**)
- Percentage of the CPU processes waiting for I/O operations(0.0%**wa**)
- Percentage of the CPU serving hardware interrupts (0.0% **hi** — Hardware IRQ
- Percentage of the CPU serving software interrupts (0.2% **si** — Software Interrupts
- The amount of CPU 'stolen' from this virtual machine by the hypervisor for other tasks (such as running another virtual machine) this will be 0 on desktop and server without Virtual machine. (0.0%**st** — Steal Time)

The fourth and fifth rows respectively indicate the use of physical memory (RAM) and swap.

And as last thing ordered by CPU usage (as default) there are the processes currently in use. Let's see what information we can get in the different columns:

- **PID**– l'IDof the process(1880)
- **USER**– The user that is the owner of the process (mohammed)
- **PR**– priority of the process (20)
- **NI**– The "NICE" value of the process (0)
- **VIRT**– virtual memory used by the process (2148356m)
- **RES**– physical memory used from the process (425960m)
- **SHR**– shared memory of the process (38712)
- **S**– indicates the status of the process:**S**=sleep**R**=running**Z**=zombie (S)
- **%CPU**– This is the percentage of CPU used by this process (11.0)
- **%MEM**– This is the percentage of RAM used by the process (21.1)
- **TIME+**–This is the total time of activity of this process (44:15.78)
- **COMMAND**– And this is the name of the process (cinamon)

Question 3

Using the kill command

- *Open a terminal.*
- *Run the command ping www.cit.ie*

```
mohammed@mohammed-virtual-machine:~$ ping www.cit.ie
PING www.cit.ie (54.72.5.20) 56(84) bytes of data.
```

- *Open a second terminal.*
- *Run the command ping www.mycit.ie*

```
mohammed@mohammed-virtual-machine:~$ ping www.mycit.ie
PING www.mycit.ie (54.72.5.20) 56(84) bytes of data.
```

- *Open a third terminal. Using the ps/grep commands, clearly display information about the*

  ping programs running (i.e. type ps aux | grep ping).
- *Identify the process id PID of the two ping programs, and then kill the two ping programs*

  using the kill command (format: kill -9 process_id).
- *Verify that two ping processes were terminated, by repeating the ps command above.*
- *Close the 3 terminals.*

```
mohammed@mohammed-virtual-machine:~$ ps aux | grep ping
mohammed  13202  0.0  0.0  14948  1816 pts/0    S+   20:48   0:00 ping www.cit.ie
mohammed  13203  0.0  0.0  14948  1812 pts/1    S+   20:48   0:00 ping www.mycit.ie
mohammed  13209  0.0  0.0  14224   924 pts/2    S+   20:48   0:00 grep --color=auto ping
mohammed@mohammed-virtual-machine:~$ kill 13202
mohammed@mohammed-virtual-machine:~$ kill 13203
mohammed@mohammed-virtual-machine:~$ ps aux | grep ping
mohammed  13216  0.0  0.0  14224   924 pts/2    S+   20:49   0:00 grep --color=auto ping
mohammed@mohammed-virtual-machine:~$
```

```
mohammed@mohammed-virtual-machine:~$ ping www.cit.ie
PING www.cit.ie (54.72.5.20) 56(84) bytes of data.
Terminated
mohammed@mohammed-virtual-machine:~$
```

```
mohammed@mohammed-virtual-machine:~$ ping www.mycit.ie
PING www.mycit.ie (54.72.5.20) 56(84) bytes of data.
Terminated
mohammed@mohammed-virtual-machine:~$
```

Question 4

Setting programs in the background.

- *Open a terminal. Only one terminal is used for question 4.*
- *Go to the Part2 directory (i.e. type cd Part2).*
- *Make sure you are in the correct folder by typing pwd.*
- *Run the command ping www.cit.ie >> hold &. This places the ping program/process in the*

  background. The output is appended to the file hold.
- *Run the command ping www.mycit.ie >> hold &. This places the ping program/process in*

  the background. The output is also appended to the file hold.
- *Using the ps/grep commands, clearly display information about the ping*

  programs/processes running (i.e. type ps aux | grep ping).

```
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ pwd
/home/mohammed/OperatingSystem/Assignment2/Part2
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ ping www.cit.ie >> hold &
[1] 13525
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ ping www.mycit.ie >> hold &
[2] 13531
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ ps aux | grep ping
mohammed  13525  0.0  0.0  14948  1812 pts/2    S    21:01   0:00 ping www.cit.ie
mohammed  13531  0.0  0.0  14948  1796 pts/2    S    21:01   0:00 ping www.mycit.ie
mohammed  13536  0.0  0.0  14224  1080 pts/2    S+   21:01   0:00 grep --color=auto ping
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$
```

- ***T****ype fg to bring the last ping program/process to the foreground. Type ^C (hold the control key down and then type C) to kill the ping process.*
- *Using the ps/grep commands, clearly display information about the ping*

  programs/processes running (i.e. type ps aux | grep ping).
- *Repeat the previous two steps, until your ping programs/processes are terminated.*

```
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ fg
ping www.mycit.ie >> hold
^Cmohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ ps aux grep ping
mohammed  13525  0.0  0.0  14948  1812 pts/2    S    21:01   0:00 ping www.cit.ie
mohammed  13578  0.0  0.0  14224   924 pts/2    S+   21:03   0:00 grep --color=auto ping
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ fg
ping www.cit.ie >> hold
^Cmohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ ps aux grep ping
mohammed  13585  0.0  0.0  14224   924 pts/2    R+   21:03   0:00 grep --color=auto ping
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$
```

- *Display the file hold by typing cat hold.*
- *Close the terminal.*

```
mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$ cat hold
PING www.mycit.ie (54.72.5.20) 56(84) bytes of data.

--- www.mycit.ie ping statistics ---
86 packets transmitted, 0 received, 100% packet loss, time 87333ms

PING www.cit.ie (54.72.5.20) 56(84) bytes of data.

--- www.cit.ie ping statistics ---
144 packets transmitted, 0 received, 100% packet loss, time 146548ms

mohammed@mohammed-virtual-machine:~/OperatingSystem/Assignment2/Part2$
```

Question 5

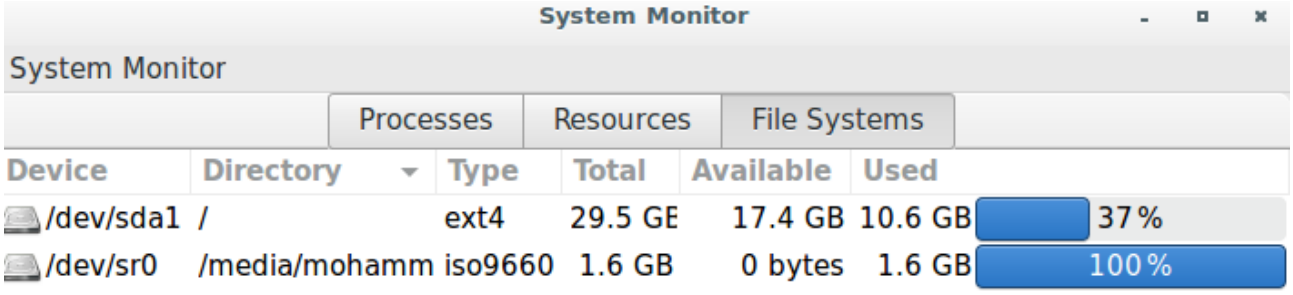
Graphical task manage, discuss similarities to top/ps commands and differences.

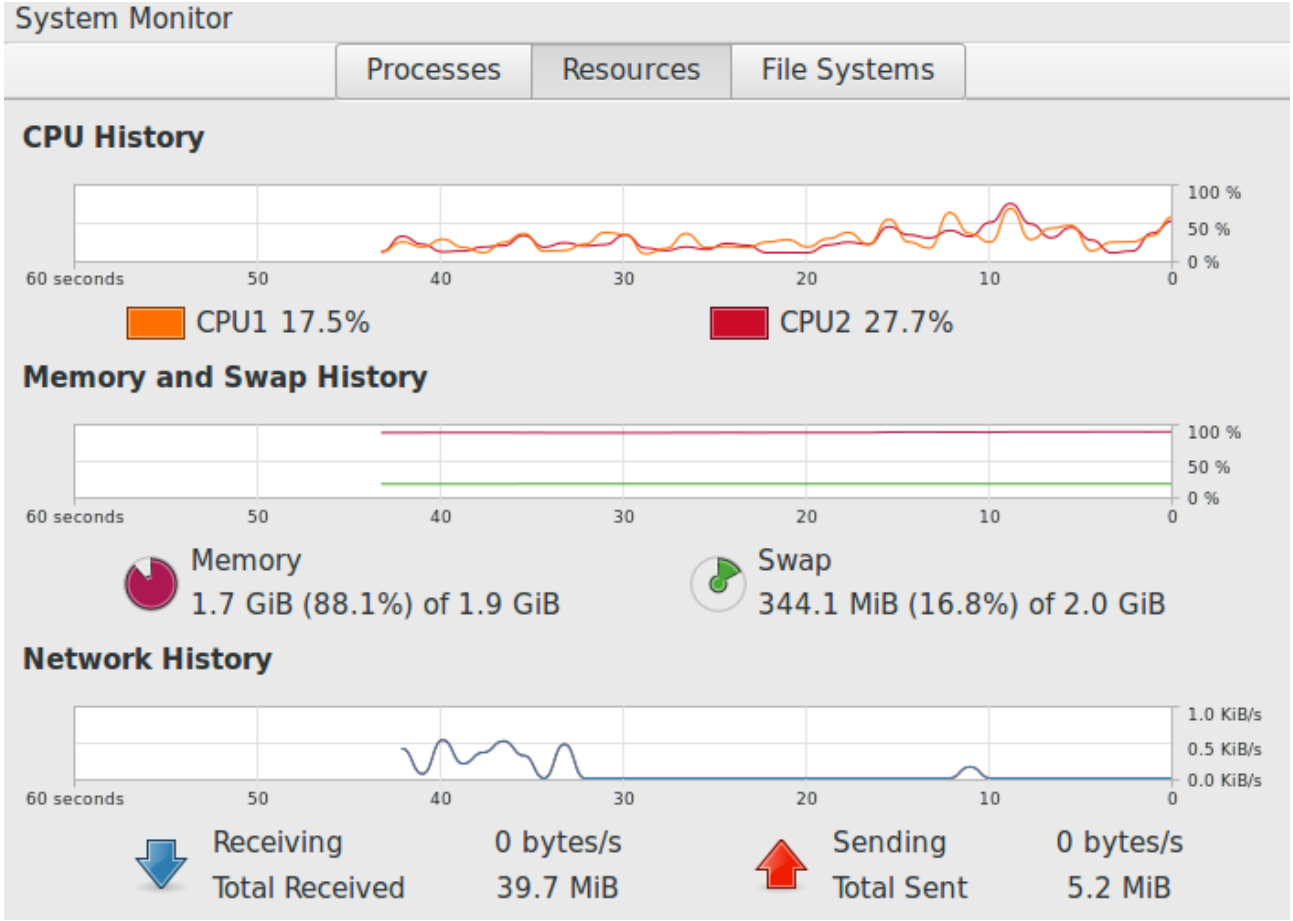
- *Open a terminal.*
- *Run the gnome-system-monitor graphical program which is similar to the task manager on*

  windows. Type gnome-system-monitor. You might have to install it if it does not exist on

  your machine (i.e. sudo apt-get update followed by sudo apt-get install gnome-system- monitor).

```
mohammed@mohammed-virtual-machine:~$ gnome-system-monitor
```


System Monitor
System Monitor
Processes
Resources
File Systems
Device
Directory
Type
Total
Available
Used
/dev/sda1 / ext4 29.5 GE 17.4 GB 10.6 GB 37 %
/dev/sr0 /media/mohamm iso9660 1.6 GB 0 bytes 1.6 GB 100 %


System Monitor
Processes
Resources
File Systems
CPU History
100 %
50 %
0 %
60 seconds
50
40
30
20
10
0
CPU1 17.5%
CPU2 27.7%
Memory and Swap History
100 %
50 %
0 %
60 seconds
50
40
30
20
10
0
Memory
1.7 GiB (88.1%) of 1.9 GiB
Swap
344.1 MiB (16.8%) of 2.0 GiB
Network History
1.0 KiB/s
0.5 KiB/s
0.0 KiB/s
60 seconds
50
40
30
20
10
0
Receiving 0 bytes/s
Total Received 39.7 MiB
Sending 0 bytes/s
Total Sent 5.2 MiB

System Monitor

Processes | Resources | File Systems

Process Name	User	% CPU	ID	Memory	Priority
applet.py	mohammed	0	2111	15.4 MiB	Normal
at-spi2-registryd	mohammed	0	1798	316.0 KiB	Normal
at-spi-bus-launcher	mohammed	0	1791	180.0 KiB	Normal
bash	mohammed	0	12974	1.9 MiB	Normal
cinnamon	mohammed	2	1880	384.1 MiE	Normal
cinnamon-killer-daemon	mohammed	0	1930	568.3 KiB	Normal
cinnamon-launcher	mohammed	0	1866	4.2 KiB	Normal
cinnamon-screensaver	mohammed	0	2068	3.7 MiB	Normal

- ***Report on the similarities to top/ps commands (around 8 lines).***

PS and TOP

All of these commands are useful in gathering information on what is malfunctioning; what is and isn't running; and what, if anything, needs to be restarted.

PS Command

Mainly used to find out what mplus services are running. This command can also be used for other services. Other services that can also be found are ndsd, web, cfs, and postgresql, to name a few.

ps -ef|grep service

This command stands for 'Process Status'. It is similar to the "Task Manager" that pop-ups in a Windows Machine when we use Cntrl+Alt+Del. This command is similar to 'top' command but the information displayed is different.

Top

Another command that is mainly for informational and troubleshooting. Running top will show a list of processes, how much of the CPU they're taking up, how much RAM is available, etc. This can be useful if a client is complaining about high CPU utilization, or the system seeming to run very slow.

- ***Report on the differences (around 6 lines). Do not forget to click on the tabs at the top of the monitor program.***

  - *top enables us to see our processes ordered by the amount of processor power they use.*
  - *ps enables to see all our processes, or just the processes used by certain users, for example root or myself.*
  - top should be used to see which processes are most active,
  - ps could be used to see which processes you (or any other user) are running currently.
  - *ps - Display currently active processes.*
  - *top - Display all running processes.*
  - *top allows you display of process statistics continuously until stopped vs. ps which gives you a single snapshot.*