

LAB 2

CSE 460

Mohamed Abdul

1) Basic Shell Programming

Write the following shell script, save it, execute it and note down its output.

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2
/home/mo~b2/ginfo [-M--] 6 L:[ 1+10 11/ 11] *(202 / 202b) <EOF> [*][X]
#
#
# Script to print user information who currently login, current data & time.
#
clear.
echo "Hello $USER"
echo "Today is ";date
echo "Number of user login : " ; who | wc -l
echo "Calendar"
cal
exit 0
```

What difference do you see when executing the script with `$./ginfo`

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2
mohamed@mohamed-VirtualBox:~/cse460/lab2$ chmod 755 ginfo
mohamed@mohamed-VirtualBox:~/cse460/lab2$ ./ginfo
```

This will create new shell to execute the program.

What difference do you see when executing the script with `$.ginfo`

This will not create new shell to execute the program, it will execute on current bash.

But, I think reason why my command line closes after I execute the program, because at the end of the program its says 'exit 0'.

Output

```
mohamed@mohamed-Virtu
Hello mohamed
Today is
Tue Jan 17 12:26:00 PST 2017
Number of user login :
1
Calendar
    January 2017
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

UDV

Q.1.How do you define variable x with value 10 and print it on screen?

```
mohamed@mohamed-VirtualB
/home/mo~uestion2 [-M--] 7 L:
#Q2->Q1
#clear
clear
#set x = 10
x=#10
echo "The value of x is $x"
exit 0
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2
mohamed@mohamed-VirtualBox:~/cse460/lab2$ chmod +x question2
mohamed@mohamed-VirtualBox:~/cse460/lab2$ ./question2
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2
The value of x is #10
mohamed@mohamed-VirtualBox:~/cse460/lab2$
```

Q.2.How do you define variable xn with value 'Rani' and print it on screen?

```
mohamed@mohamed-VirtualB
/home/mo~stion2.b [----] 6 L:
#Q2->Q2
#clear
clear
#set xn = rani
xn="rani"
echo "The value of xn is $xn"
exit 0
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/2
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$ chmod +x question2.b
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$ ./question2.b
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/2
The value of xn is rani
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$
```

Q.3.How do you print the sum of two numbers, say, 6 and 3?

```
mohamed@mohamed-  
/home/mo~estion2C [----]  
#  
#clear  
clear  
#sum of 3 + 6  
echo "Sum of 3 + 6 :".  
expr 3 + 6  
exit 0
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/2  
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$ chmod +x question2C  
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$ ./question2C
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/2  
Sum of 3 + 6 :  
9  
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$
```

Q.4.How do you define two variables x=20, y=5 and then print the quotient of x and y (i.e. x/y)?

```
mohamed@mohamed-VirtualBox  
/home/mo~estion2D [-M--] 32 L:[  
#Q2->QD  
#clean  
clear  
#define x = 20  
x=20  
#define y = 5  
y=5  
#quotient of x/y  
echo "The quotient of $x and $y:"  
expr $x / $y  
exit 0
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/2  
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$ chmod +x question2D  
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$ ./question2D
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/2  
The quotient of 20 and 5:  
4
```

Q.5.Modify the above question to store the result of dividing x by y to a variable called z.

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/2
/home/mo~estion2E [-M--] 16 L:[ 1+ 9 10/ 13]
#Q2->QE
#clean
clear
#define x = 20
x=20
#define y = 5
y=5
#quotionet of x/y saved in z
echo "The x/y qutioned are saved in z, so z is: "
z=`expr $x / $y`
echo $z.
exit 0
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/2
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$ chmod 755 question2E
mohamed@mohamed-VirtualBox:~/cse460/lab2/2$ ./question2E
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab
The x/y qutioned are saved in z, so z is:
4
```

testShell.sh

```
mohamed@moh
/home/mo~Shell.sh [
#XYZ
XYZ=2017
```

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/3
mohamed@mohamed-VirtualBox:~/cse460/lab2/3$ mcedit testShell.sh
mohamed@mohamed-VirtualBox:~/cse460/lab2/3$ chmod +x testShell.sh
mohamed@mohamed-VirtualBox:~/cse460/lab2/3$ ./testShell.sh
mohamed@mohamed-VirtualBox:~/cse460/lab2/3$ echo $XYZ
2017
mohamed@mohamed-VirtualBox:~/cse460/lab2/3$
```

I tried both ways to execute, but the first one wasn't printing the output, but second one does as you can see on the screen shot. And the first one executes the program in new shell.


```

mohamed@mohamed-VirtualBox:~/cse460/lab2/awk$ clear

mohamed@mohamed-VirtualBox:~/cse460/lab2/awk$ ps auxw | awk '{print $1 "\t\t" $2}'
USER      PID
root      1
root      2
root      3
root      5
root      7
root      8
root      9
root     10
root     11
root     12
root     13
root     15
root     16
root     17
root     18
root     20
root     21

```

This prints the process table, this command only prints the field 1, two tabs in the middle and the second field of the process which is process ID. It's using pipeline command to to the output of ps auxw to awk as a input, this what that command does.

3) Viewing Processes

```

mohamed@mohamed-VirtualBox: ~/cse460/lab2/vprocess
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ls
robot.cpp
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ g++ robot.cpp
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ls
a.out  robot.cpp
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ./a.out
^C
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ./a.out &
[1] 2529
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ps
  PID TTY          TIME CMD
 2384 pts/1    00:00:00 bash
 2529 pts/1    00:00:09 a.out
 2530 pts/1    00:00:00 ps
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ kill 2529
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ps
  PID TTY          TIME CMD
 2384 pts/1    00:00:00 bash
 2533 pts/1    00:00:00 ps
[1]+  Terminated                  ./a.out

```

Nice and renice

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/vprocess
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ g++ robot.cpp
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ./a.out &
[1] 2698
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ nice robot &
[2] 2702
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ nice: 'robot': No such file or directory
^C
[2]+  Exit 127                  nice robot
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ps
  PID TTY          TIME CMD
 2384 pts/1        00:00:00 bash
 2698 pts/1        00:00:15 a.out
 2703 pts/1        00:00:00 ps
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ renice 2 2698
2698 (process ID) old priority 0, new priority 2
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$
```

4) Starting New Processes

```
mohamed@mohamed-VirtualBox:~/cse460/lab2/viewProcess$ sh -c "echo 'Hello, CSUSB'"
Hello, CSUSB
mohamed@mohamed-VirtualBox:~/cse460/lab2/viewProcess$
```

test_system.cpp

```
//test_system.cpp
#include <stdlib.h>
#include <iostream>

using namespace std;

int main()
{
    cout << "Running ps with system\n";

    system ( "ps -ax" ); //system ( "ps -ax" );

    cout << "Done \n";

    return 0;
}
```

Compiled without '&' system ("ps -ax ")

```
mohamed@mohamed-VirtualBox:~/cse460/lab2/viewProcess$ gedit test_system.cpp
^C
mohamed@mohamed-VirtualBox:~/cse460/lab2/viewProcess$ g++ -o test_system test_system.cpp
mohamed@mohamed-VirtualBox:~/cse460/lab2/viewProcess$ ./test_system
Running ps with system
  PID TTY          STAT       TIME COMMAND
  1 ?        Ss          0:02 /sbin/init splash
  2 ?        S           0:00 [kthreadd]
  3 ?        S           0:00 [ksoftirqd/0]
  5 ?        S<          0:00 [kworker/0:0H]
  7 ?        S           0:00 [rcu_sched]
  8 ?        S           0:00 [rcu_bh]
```

Compiled with '&' system ("ps -ax &")

```
mohamed@mohamed-VirtualBox:~/cse460/lab2/viewProcess$ g++ -o test_system test_system.cpp
mohamed@mohamed-VirtualBox:~/cse460/lab2/viewProcess$ ./test_system
Running ps with system
Done
mohamed@mohamed-VirtualBox:~/cse460/lab2/viewProcess$ PID TTY          STAT       TIME COMMAND
  1 ?        Ss          0:02 /sbin/init splash
  2 ?        S           0:00 [kthreadd]
  3 ?        S           0:00 [ksoftirqd/0]
  5 ?        S<          0:00 [kworker/0:0H]
  7 ?        S           0:00 [rcu_sched]
```

This executes in the background of the process. Technically it's still running, you can kill it. That's why it's behind the "process table".

5) Shell Programming Practice

What does the option "-v" in the grep command do?

It's invert the sense of matching, to select non-matching lines

terminateProcess

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/vprocess
#initialize the value to count the deleted processors
count=0
for pid in $(ps -e -f | grep $1 | grep -v grep | grep -v $0 | awk '{print $2}')
do
    kill $pid
#increment the value by 1
# let count+=1 or
((count+=1))
done
#checks if the count is greater than 0
if(($count > 0));
then
    echo " The $count Process are deleted!"
else
    echo "There is No process found."
fi
```


Before execute the function

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/vprocess
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ps -l
F S    UID    PID    PPID    C  PRI   NI ADDR SZ WCHAN  TTY          TIME CMD
0 S    1000    13091   13085    0   80    0  -  7439 wait   pts/1        00:00:00 bash
0 R    1000    20653   13091   51   80    0  -  3317 -      pts/1        00:00:07 robot
0 R    1000    20654   13091   53   80    0  -  3317 -      pts/1        00:00:07 robot
0 R    1000    20655   13091   47   80    0  -  3317 -      pts/1        00:00:06 robot
0 R    1000    20656   13091   46   80    0  -  3317 -      pts/1        00:00:06 robot
0 R    1000    20657   13091   43   80    0  -  3317 -      pts/1        00:00:05 robot
0 R    1000    20658   13091   43   80    0  -  3317 -      pts/1        00:00:05 robot
0 R    1000    20659   13091   41   80    0  -  3317 -      pts/1        00:00:05 robot
0 R    1000    20660   13091   43   80    0  -  3317 -      pts/1        00:00:04 robot
0 R    1000    20661   13091   43   80    0  -  3317 -      pts/1        00:00:04 robot
0 R    1000    20664   13091    0   80    0  -  8996 -      pts/1        00:00:00 ps
```

After the execution

```
mohamed@mohamed-VirtualBox: ~/cse460/lab2/vprocess
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ chmod 755 terminateProcess
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ./terminateProcess robot
The 9 Process are deleted!
[1] Terminated ./robot
[2] Terminated ./robot
[3] Terminated ./robot
[4] Terminated ./robot
[5] Terminated ./robot
[6] Terminated ./robot
[7] Terminated ./robot
[8]- Terminated ./robot
[9]+ Terminated ./robot
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$ ./terminateProcess robot
There is No process found.
mohamed@mohamed-VirtualBox:~/cse460/lab2/vprocess$
```

Evaluation:

Part 1: In this lab I have learned about the shell scripting, I have created several programs that run but shell scripting and executed them. I learned a lot from this assignment. And also most interesting thing about the assignment is terminating the process using pid command.

Part 2: I have successfully executed the program, and I also provided the output for the program. And also learned in this lab.

Score: 20/20