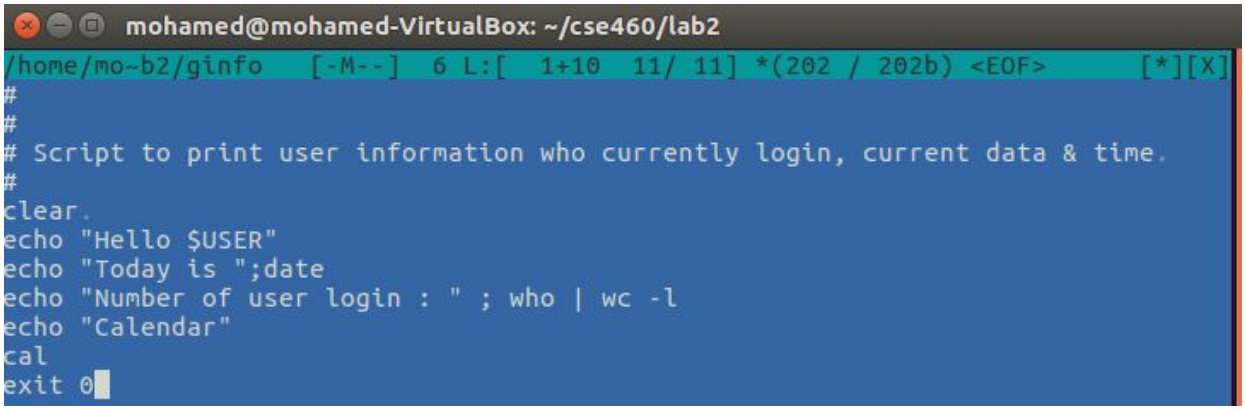
LAB 2

CSE 460

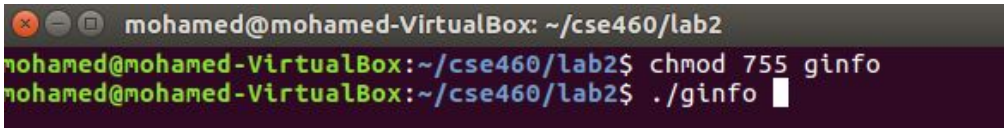
Mohamed Abdul

1) **Basic Shell Programming**

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



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

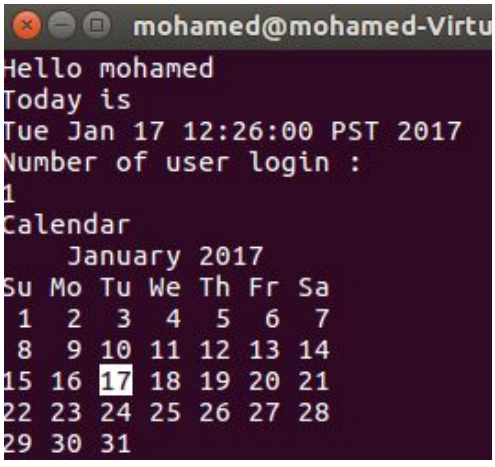


This will create new shell to execute the program.

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

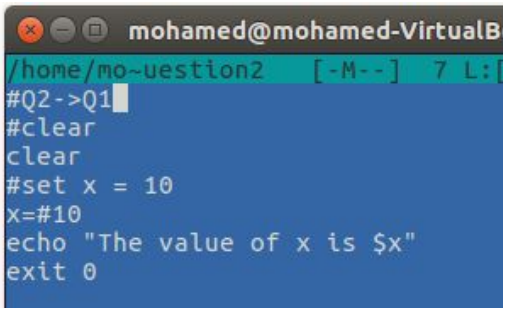
This will not create new shell to execute the program, it will execute on current bash. But, I think reason why my command line closes after I execute the program, because at the end of the program its says ‘exit 0’.

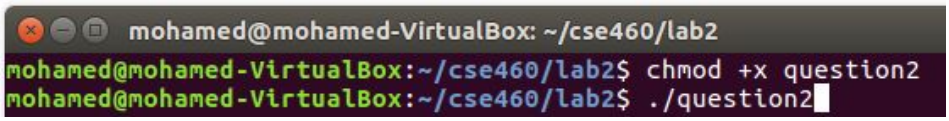
Output

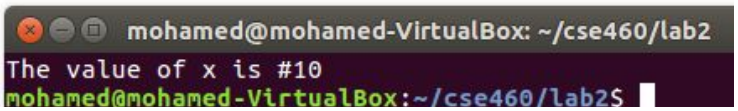


UDV

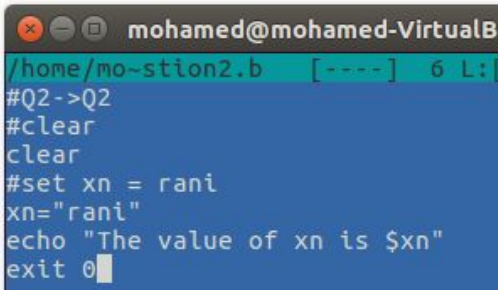
Q.1.How do you define variable x with value 10 and print it on screen?

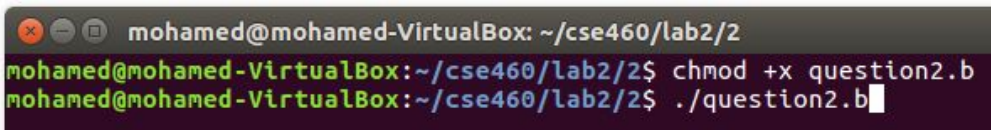


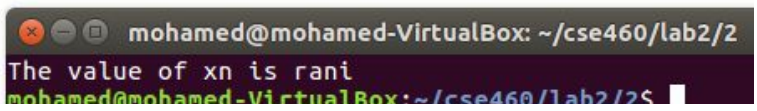




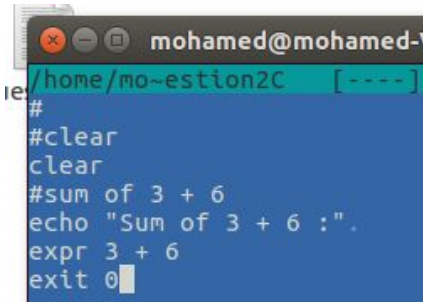
Q.2.How do you define variable xn with value 'Rani' and print it on screen?

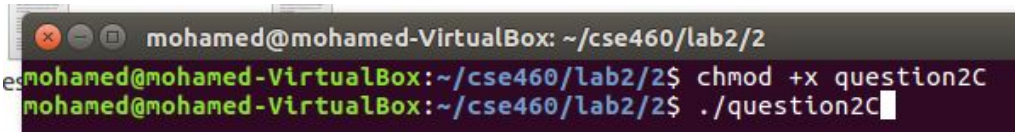


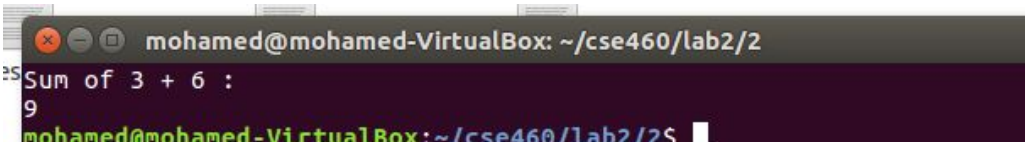




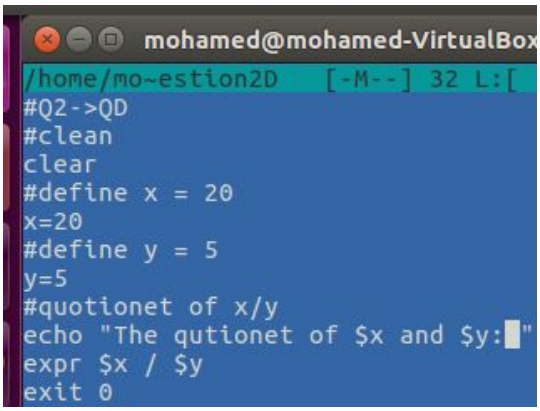
Q.3.How do you print the sum of two numbers, say, 6 and 3?

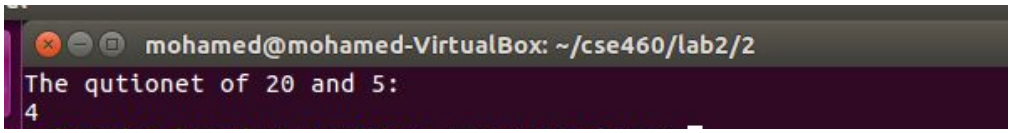




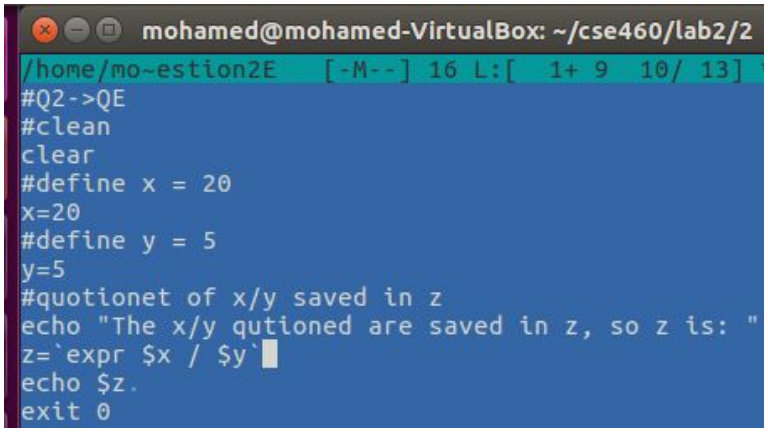


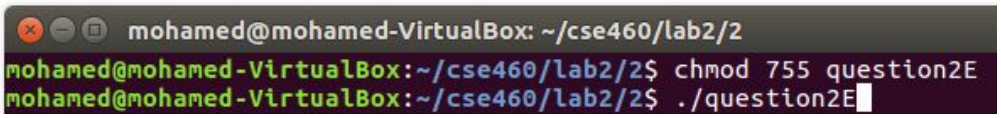
Q.4.How do you define two variables x=20, y=5 and then print the quotient of x and y (i.e. x/y)?

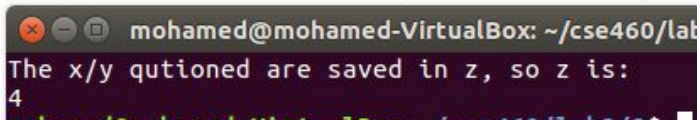




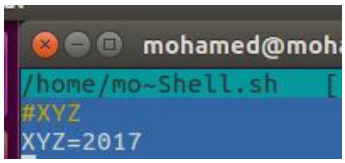
Q.5.Modify the above question to store the result of dividing x by y to a variable called z.

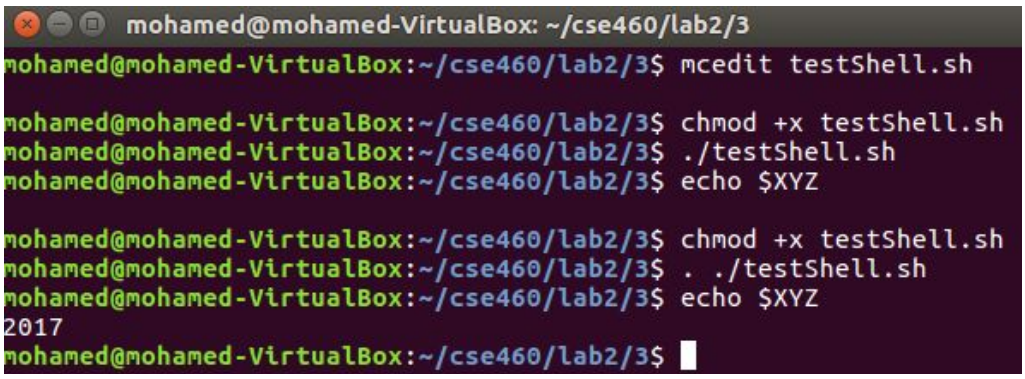






testShell.sh





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.

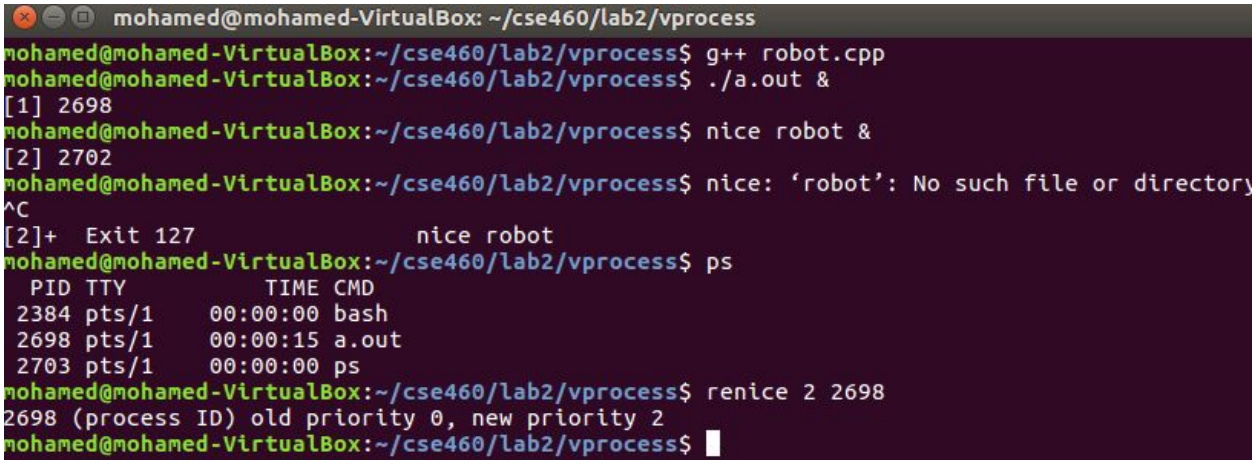
```
mohamed@mohamed-VirtualBox:~/cse460/lab2/awk$ clear

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 auwx to awk as a input, this what that command does.

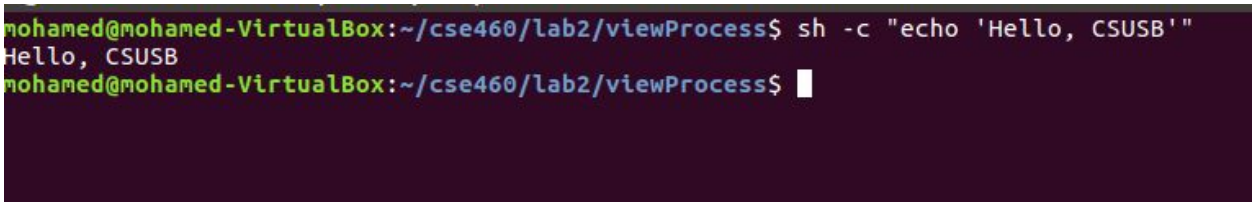
3) Viewing Processes

```
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



4) Starting New Processes



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]
```

This executes in the background of the process. Technically it's still running, you can kill it. Thats why its behind the “process table”.

5) **Shell Programming Practice**

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

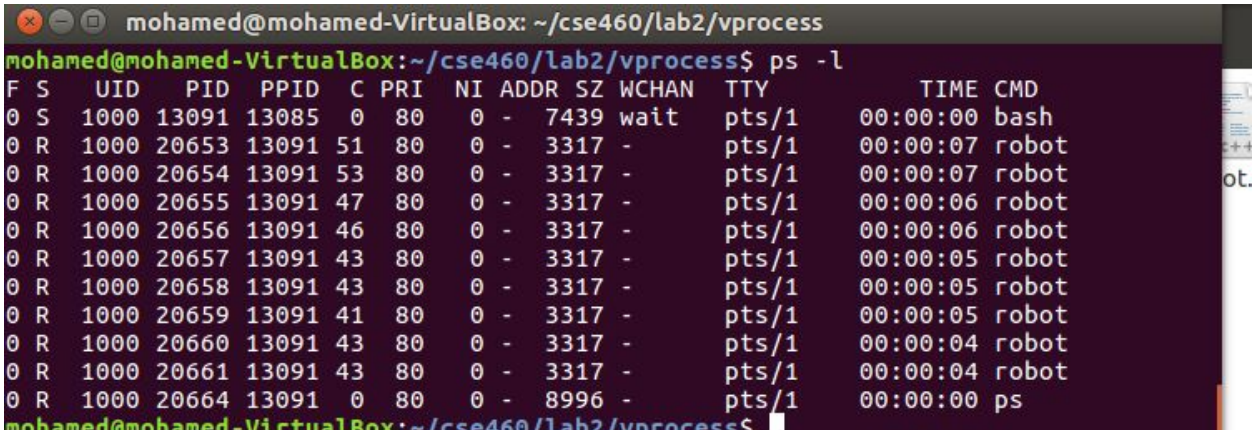
Its invert the sense of matching, to select non-matching lines

terminateProcess

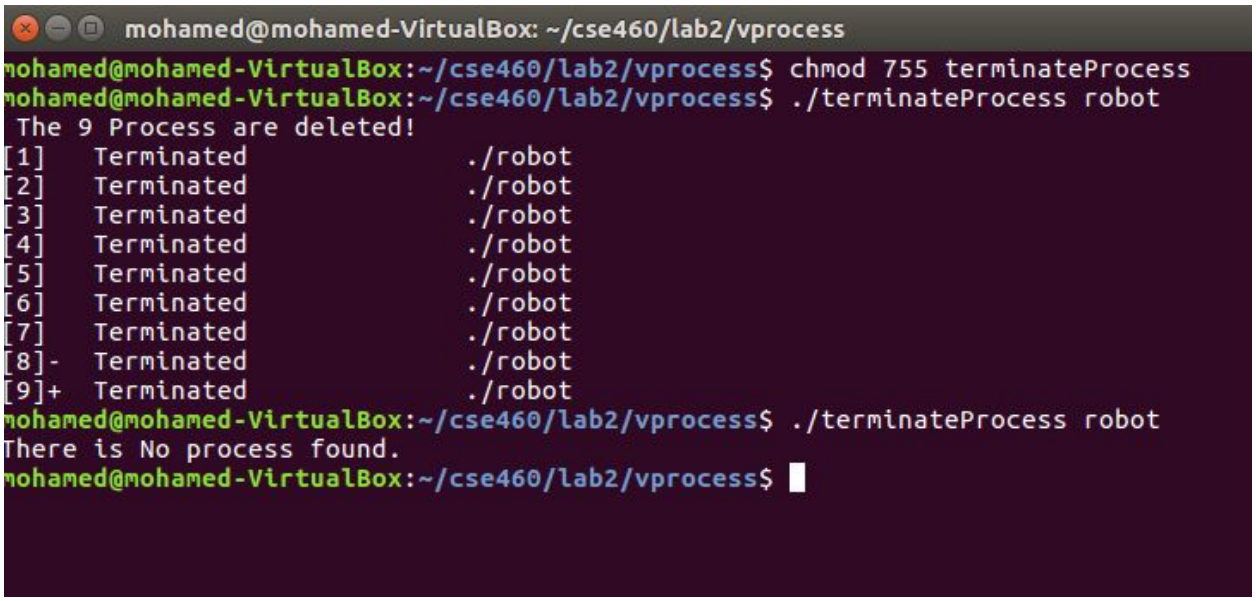
mohamed@mohamed-VirtualBox: ~/cse460/lab2/vprocess

```
#intialize 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 $pid
#increament the value by 1
# let count+=1 or
 ((count+=1))
done
#checks if the count is greater than 0
 if(($count > 0));
then
   echo " The $count Process are deleted!"
else
    echo "There is No process found."
 fi
```

Before execute the function



After the execution



Evaluation:

Part 1: In this lab I have learned about the shell scripting, I have a 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