

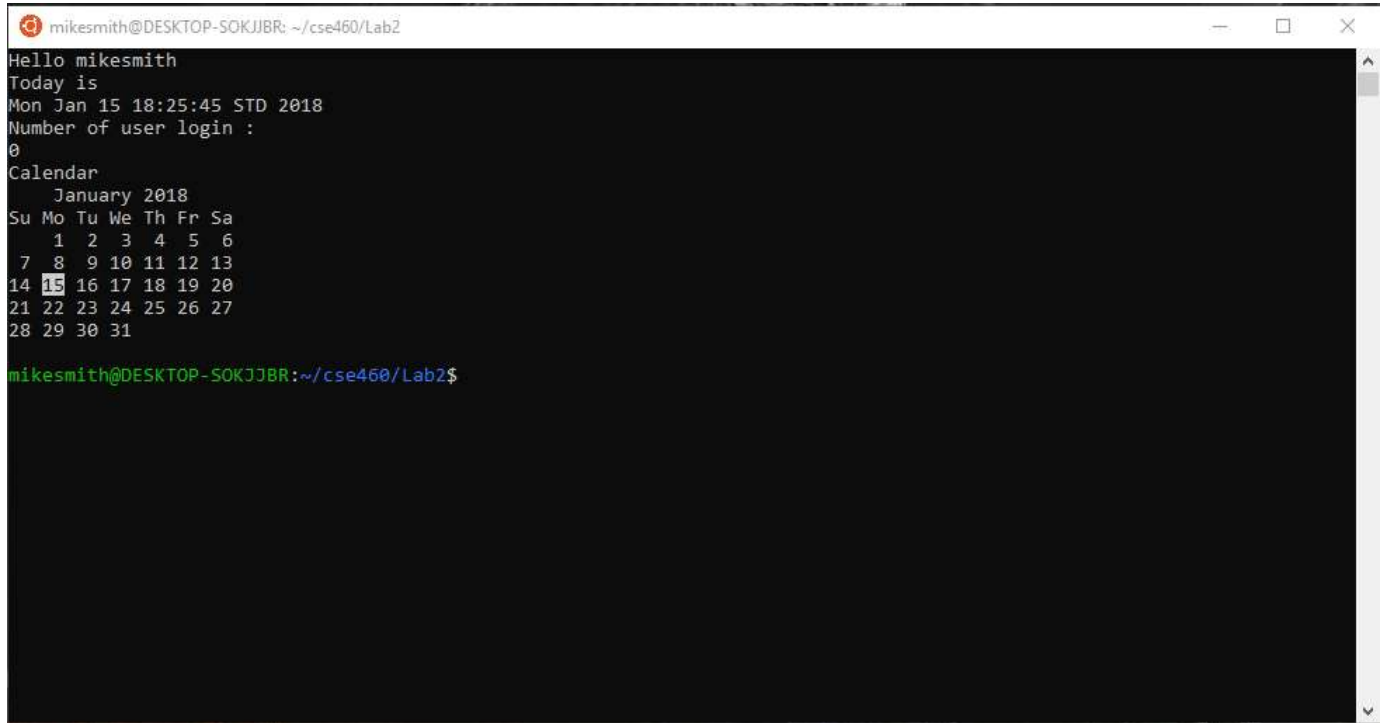
Michael Smith

CSE460

Lab 2

20 Possible Points

1.) Basic Shell Programming
 - a. How to write shells Exercise -



```
mikesmith@DESKTOP-SOKJJBR: ~/cse460/Lab2
Hello mikesmith
Today is
Mon Jan 15 18:25:45 STD 2018
Number of user login :
0
Calendar
  January 2018
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

mikesmith@DESKTOP-SOKJJBR:~/cse460/Lab2$
```

The difference is that when executing the `./ginfo` the output is as intended. But when executed as `./ginfo` the shell opens a new window and is closed upon exit. This is due to the final line being `exit 0`.

- b. Variable Exercise -

2.)To define variable xn with the value of 'Rani' and print it you use:

\$ x_n=Rani

\$echo \$xn

3.)To print the sum the sum of two number you do:

```
$ expr 6 + 3
```

4.) To define the two variables and print the quotient you do:

\$ x=20

\$ y=5

\$ expr x / y

5.) To modify question 4 to store quotient in z you do:

\$ x=20

\$ y=5

```
$ z='expr x / y'
```

\$ echo \$z

Shell script:

```
mikesmith@DESKTOP-SOKJBR: ~/cse460/Lab2
```

```
# Holds the statement for question 5
#
XYZ=2017
```

```
"testShell.sh" 4L, 50C                                     1,1      All
```

```
mikesmith@DESKTOP-SOKJJBR: ~/cse460/Lab2
mikesmith@DESKTOP-SOKJJBR:~$ cd cse460/Lab2
mikesmith@DESKTOP-SOKJJBR:~/cse460/Lab2$ ls
first ginfo testShell.sh varinfo
mikesmith@DESKTOP-SOKJJBR:~/cse460/Lab2$ ./testShell.sh
mikesmith@DESKTOP-SOKJJBR:~/cse460/Lab2$ echo $XYZ

mikesmith@DESKTOP-SOKJJBR:~/cse460/Lab2$ . ./testShell.sh
mikesmith@DESKTOP-SOKJJBR:~/cse460/Lab2$ echo $XYZ
2017
mikesmith@DESKTOP-SOKJJBR:~/cse460/Lab2$
```

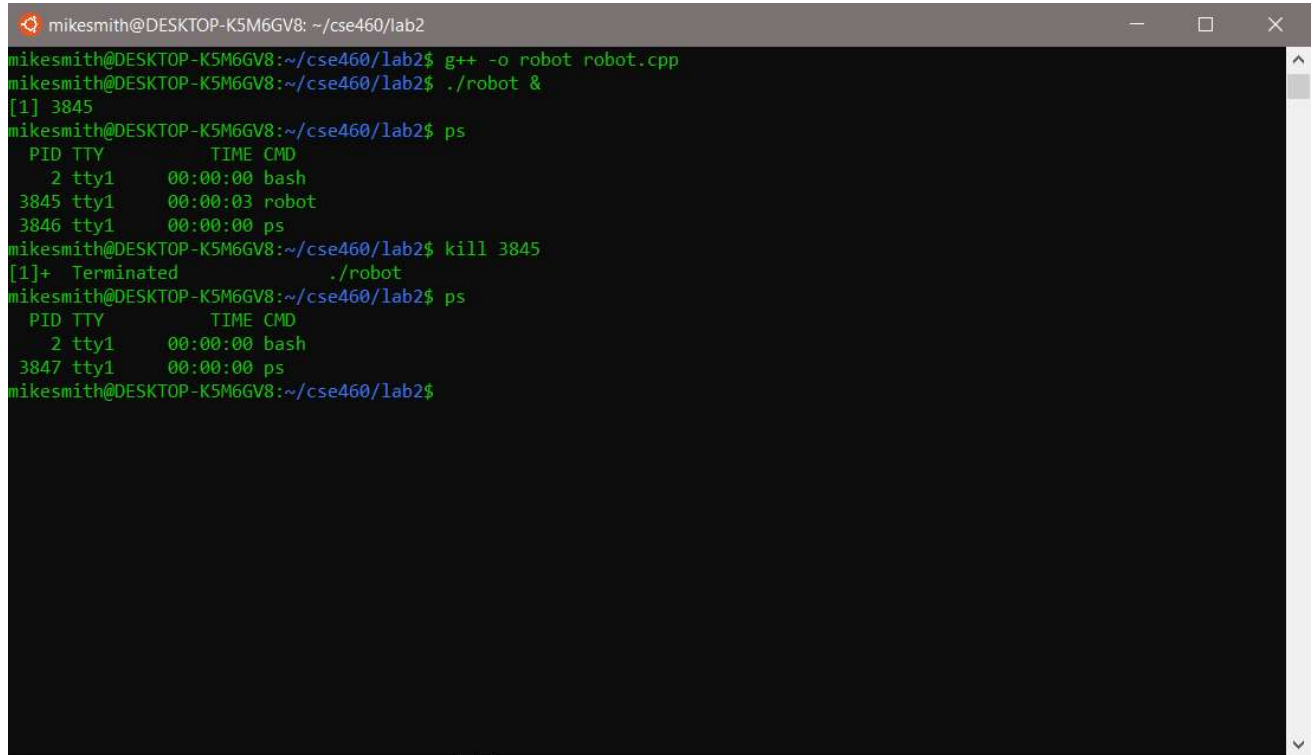
When executed as `./testShell.sh` the `echo` command for `XYZ` displays nothing, while executed as `. ./testShell.sh` the `echo` command properly prints out the value. The difference being the `.` Command executing the shell in the current shell and not creating a new copy of a shell.

2.) Awk Exercise -

```
mikesmith@DESKTOP-K5M6GV8: ~/cse460
mikesmith@DESKTOP-K5M6GV8:~/cse460$ ps auxw | awk '{print $1 "\t\t" $2}'
USER      PID
root      1
mikesmi+  2
mikesmi+  23
mikesmi+  24
mikesmith@DESKTOP-K5M6GV8:~/cse460$
```

This command will output the process table. You see two lists, the first being the name, with two tabs in the middle and the process ID number. This command outputs `ps auxx` to `awk` as the input.

3.) Kill command



```
mikesmith@DESKTOP-K5M6GV8: ~/cse460/lab2
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ g++ -o robot robot.cpp
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ./robot &
[1] 3845
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ps
  PID TTY          TIME CMD
    2 tty1      00:00:00 bash
   3845 tty1      00:00:03 robot
   3846 tty1      00:00:00 ps
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ kill 3845
[1]+  Terminated                  ./robot
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ps
  PID TTY          TIME CMD
    2 tty1      00:00:00 bash
   3847 tty1      00:00:00 ps
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$
```

Nice and renice

```
mikesmith@DESKTOP-K5M6GV8: ~/cse460/lab2
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ g++ robot.cpp
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ./robot &
[1] 3854
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ nice robot &
[2] 3855
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ nice: cannot set niceness: Permission denied
nice: 'robot': No such file or directory
ps
  PID TTY          TIME CMD
    2 tty1      00:00:00 bash
   3854 tty1      00:00:18 robot
   3856 tty1      00:00:00 ps
[2]+  Exit 127              nice robot
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ renice 2 3854
3854 (process ID) old priority 0, new priority 2
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$
```

4.) Starting a new process

```
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ sh -c "echo 'Hello, CSUSB.'"
Hello, CSUSB.
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$
```

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 the &

```

mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ g++ -o test_system test_system.cpp
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ./test_system
Running ps with system
  PID TTY          STAT       TIME COMMAND
    1 ?           Ss          0:00 /init
    2 tty1        Ss          0:00 -bash
 3854 tty1        RN          8:50 ./robot
 3879 tty1        S           0:00 ./test_system
 3880 tty1        S           0:00 sh -c ps -ax
 3881 tty1        R           0:00 ps -ax
Done
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$

```

Compiled with the &

```

mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ g++ -o test_system test_system.cpp
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ./test_system
Running ps with system
Done
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$

```

	PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:00		/init
2	tty1	Ss	0:00		-bash
3854	tty1	RN	9:50		./robot
3891	tty1	R	0:00		ps -ax

With the '&' the process is started as a background process that can be killed. While without the '&' the program is executed and then exits on finish.

5.) Shell Programming Practice

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

The -v command inverts the matching, to force select non-matching lines.

TerminateProcess

```

#Set count to 0 for 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 count
    ((count+=1))
done

#Check to make sure the aboved worked
if(($count > 0));
then
    echo "$count processes were deleted."
else
    echo "Not an existing process."
fi

```

Execution of Terminate Process


```
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ps
  PID TTY          TIME CMD
    2 tty1      00:00:00 bash
   17 tty1      00:00:05 robot
   18 tty1      00:00:04 robot
   19 tty1      00:00:04 robot
   21 tty1      00:00:00 ps
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ./terminateProcess robot
3 processes were deleted.
[1] Terminated ./robot
[2]- Terminated ./robot
[3]+ Terminated ./robot
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$ ./terminateProcess robot
Not an existing process.
mikesmith@DESKTOP-K5M6GV8:~/cse460/lab2$
```

6.) Evaluation

This lab was to cover the basics of shell commands and to explore some possibilities with shells. We also covered user created variables and system commands. There was a lot of content covered in this lab, but creating the final shell to kill a specific process I spent the most time on to finish. I was able to successfully execute and answer each step of the lab.

Score 20/20