LAB 2

CSE 460

Waled Salem

1) Basic Shell Programming

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

#

#

# Script to print user info who currently log in, current date and time

#

clear

echo "Hello $USER"

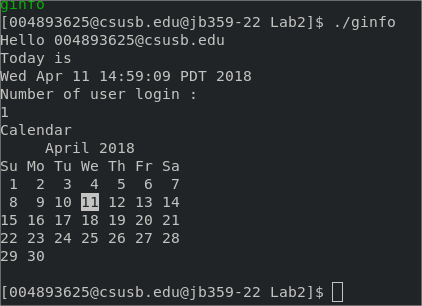
echo "Today is ";date

echo "Number of user login : " ; who | wc -l

echo "Calendar"

cal

exit 0



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

This will create another shell to execute the program. But the shell automatically exits due to the exit 0 line.

UDV

Q1 How do you define variable x with value 10 and print it on screen?

#Q1

#clear

clear

#set x = 10

x=#10

echo "The value of x is $x"

exit 0





Q2: How do you define variable xn with value 'Rani' and print it on screen?

#Q2B

#cler

clear

#set xn = rani

xn="rani"

echo "The value of xn is $xn"

exit 0



Q3: How do you print the sum of two numbers, say, 6 and 3?

#

#clear

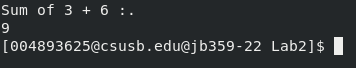
clear

#sum of 3 + 6

echo "Sum of 3 + 6 :".

expr 3 + 6

exit 0



Q4: How do you define two variables x=20, y=5 and then print the quotient of x and y (i.e.

x/y)?

#clear

clear

#let x = 20

x=20

#let y = 5

y=5

#quotients of x/y

echo "x/y is"

echo $(($x/$y))

exit 0

Q5: Modify the above question to store the result of dividing x by y to a variable called z.

#Q5

clear

#assign variables

x=20

y=5

#assin divide

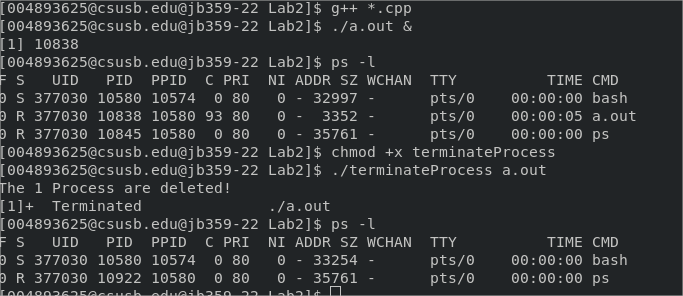
z=$(($x/$y))

echo "x/y is"

echo $z

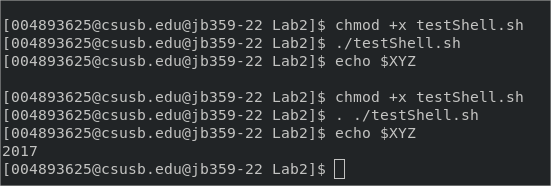
exit 0



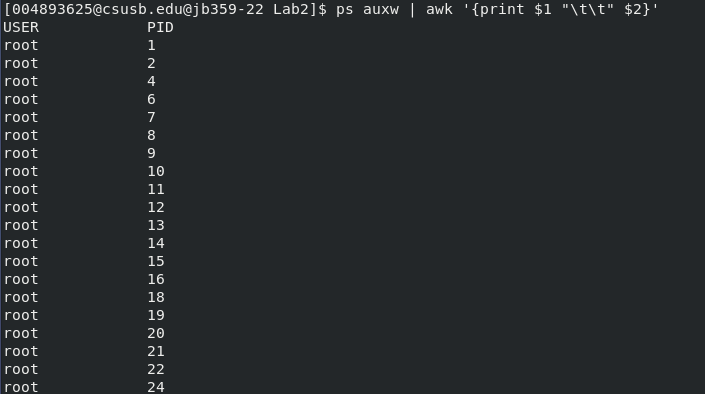


testShell.sh

#set xyz

XYZ=2017

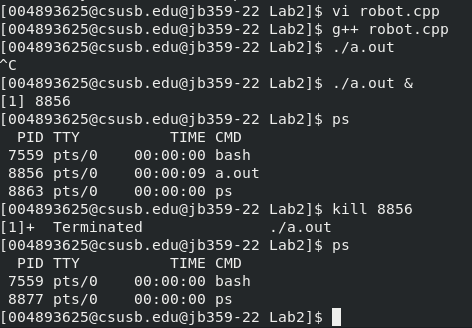
I did both executions. No output on the first execution, but the second one ran perfectly.

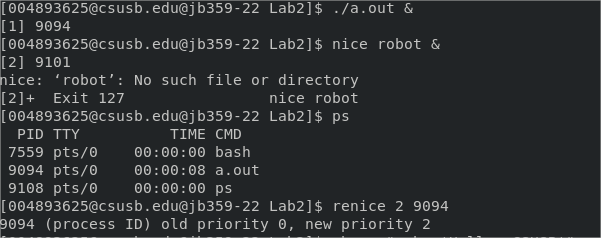


The Process ID table is printed along with what is running each process.

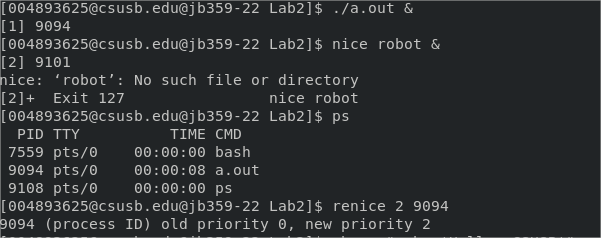
3)Viewing processes

How to kill processes.





Nice and renice



4)Starting new processes



test\_system.cpp

//test\_system.cpp

#include <stdlib.h>

#include <iostream>

using namespace std;

int main()

{

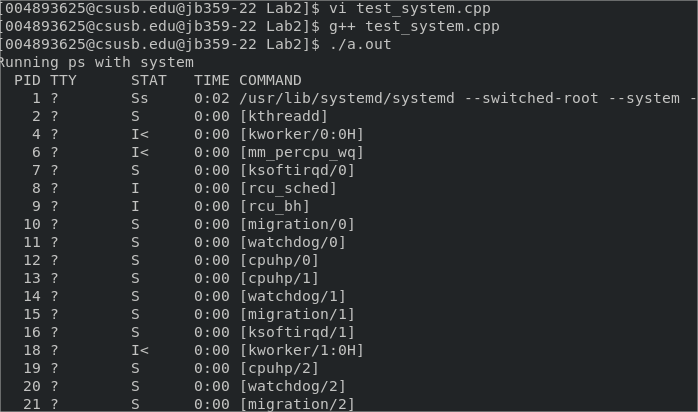
cout << "Running ps with system\n";

system ( "ps -ax &" );

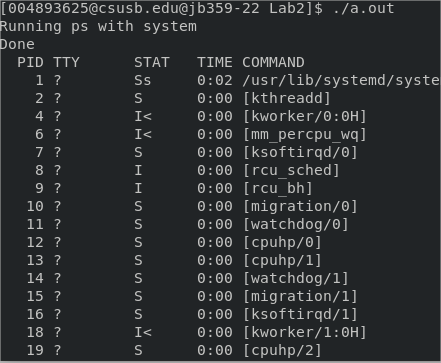
cout << "Done \n";

return 0;

}

Using just system(“ps -ax”)

Using just system(“ps -ax &”)



5) Shell Programming Practice

What does the option “-v” in the grep command do?

It inverts allowing grep to select all non-matching lines.

TerminateProcess

#initialize

count=0

for pid in $(ps -e -f | grep $1 | grep -v grep | grep -v $0 | awk '{print $2}')

do

kill $pid

#increment

((count+=1))

done

#check

if(($count > 0));

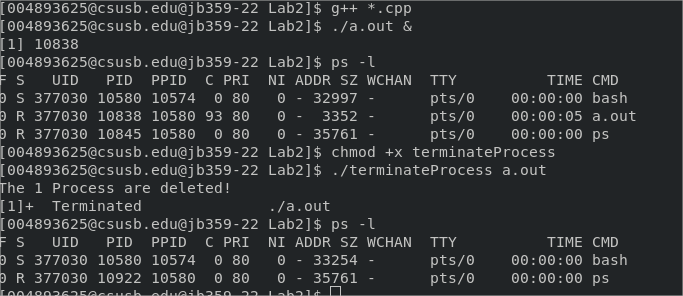
then

echo "The $count Process are deleted!"

else

echo "No process found."

fi

exit 0

Evaluation” In the lab I learned how to shell script and run tasks along with read running process and how to kill running processes. All programs have been successfully executed and display the correct output.

**Score: 20/20**