**CSCD 240  
Lab 4**

**Note: You have to complete this lab by ssh login into cslinux.eastern.ewu.edu.**

**Turn in on Canvas**🡪**Assignments**🡪**Lab4**

In **question 1 through 6,** you should capture the prompt, the command, and the output from the command.

1. What is the difference between a shell variable and an environment variable in the bash shell?
2. Define what a process is and what a job is, clearly explain how jobs differ from processes.
3. Consider the following command ls – al | less.
   1. How many processes are created with that command?
   2. What exactly does “|” do in this command?
4. Using the man page for **ps**
   1. Issue and capture the **ps** command with the appropriate options to allow listing of all processes in the system.
   2. Using the output from part A, what was the first process started and by whom was it started?
   3. What was the first non-root process that was started?
   4. What was the last process started and by whom?
5. Using a text editor create the following C programmed named almostEndless.c

#include <stdio.h>

int main()

{

int x = 0;

while(x < 20000000)

{

printf("..");

fflush(stdout);

sleep(3);

x ++;

}// end while

return 0;

}// end main

* 1. Compile your program with gcc almostEndless.c
  2. Start your program with ./a.out
  3. With your program running, describe the commands you would use (without using ctrl-c) to terminate that program, from the same terminal window in which it was started.
  4. Execute and capture the commands, using process notation (PID), to terminate a.out
  5. Restart your program with ./a.out
  6. Execute and capture the commands, using job notation (job ID), to terminate a.out

1. In a single terminal window capture the command to start a.out 3 times each running as background jobs: **(Hint**: **redirection** is useful here)
   1. What are the job numbers of the above?
   2. What are the process ID numbers of the above?
   3. Capture the command and output to bring the second a.out to the foreground.
   4. Capture the command(s) to send a.out back to the background.
   5. Capture the command(s) to kill the third a.out using its job number.
   6. Capture the command(s) to kill the first a.out using its process number.
   7. Can CTRL C be used to kill any job? Why or why not? Clearly explain why or why not.