**CPSC 3125 – Spring 2017 – In Class Lab 1**

Problem 1: Consider the following C program. (Assume that the appropriate #include statements are available above the function's header) Hint – Compiler and run the program within Ubuntu to experiment with it. Feel free to add printf() statements where needed to trace the program's execution.

int main() {

fork();

fork();

fork();

}

Including the original parent process, how many processes are created when we execute this program? Explain.

There are 8 processes by the end of this code's execution because every child process created by fork() will execute commands alongside its parent beginning from its point of creation within the code.

Problem 2:

Consider the following C program. (Assume that the appropriate #include statements are available above the function's header) Hint – Compiler and run the program within Ubuntu to experiment with it. Feel free to add printf() statements where needed to trace the program's execution.

int main()

{

pid\_t ret1;

ret1 = fork();

if(ret1 == 0){

pid\_t ret2 = fork();

printf(“Hey there!\n”);

if(ret2 == 0) fork();

fork();

printf(“Hey there again!\n”);

}

else if(ret1>0){

pid\_t ret3 = fork();

if(ret3 > 0) printf(“Hey there once again!\n”);

}

}

How many times will the line “Hey there!” appear on the screen? Explain.

Two times. After ret2 is created, there are only 2 processes that operate past the if statement's condition (ret1 == 0).

How many times will the line “Hey there again!” appear on the screen? Explain.

Eight times. Once ret2 is made and both processes output "Hey there!", ret2 only uses fork() once more, while its child uses it twice more. Meanwhile, ret1 is already beyond the first if condition, but it later became a parent to ret2, making it unable to get past the next if condition so it only uses fork() once more, and both processes continue.

How many times will the line “Hey there once again!” appear on the screen? Explain.

Only once. At the point where the command is executed, the main process is the only one that meets the condition at that time.

Including the original parent process, how many processes are created when we execute this program? Explain.

10 processes

