**Process Termination Lab**

Mark Tan

17 September 2017

CECS 326 MW

damtan1996@gmail.com

This lab is a hands-on activity that involves using the kill command in Linux, “kill -9 PID”, in which the process ID, also known as PID, represents the task a user intends to terminate. To check if the desired task to eliminate has been removed, a statistic task view is used to identify the tasks that are running, as well as information tied to that task. One can use the command “ps aux” to access this view. To test the kill command, a rogue program will be used in which one of its processes will run out of control.

**Rogue program (saved as processterm.cpp, stored in Documents folder):**

// Mark Tan

// processterm.cpp

// This file is intended to have a process that goes out of control.

#include <cstdlib>

#include <unistd.h>

#include <time.h>

int main()

{

struct timespec delay, left;

delay.tv\_sec = 0;

delay.tv\_nsec = 500000000;

for(;;)

nanosleep(&delay,&left);

exit(0);

}

**Terminal One**

INPUT cd Documents

INPUT g++ processterm.cpp

INPUT ./a.out&

OUTPUT [1] 2940

INPUT top (this opens a monitor that simulates a tasks and runs the program in the background)

**Terminal Two**

INPUT ps aux

OUTPUT statistic view of tasks (the ‘.’ represents another active process)

USER PID CPU MEM VMS RSS T S START TIME COMMAND

-

-

natkram+ 2940 0.0 0.0 4224 600 pts/4 S 10:27 0:00 ./a.out

-

INPUT kill -9 2940 (this should terminate the rogue program)

INPUT ps aux

OUTPUT statistic view of tasks (the ‘-’ represents another active process)

USER PID CPU MEM VMS RSS T S START TIME COMMAND

-

-

-

-