CS F321 Operating Systems

Lab 02 Process and Signal Handling

Justyn P. Durnford

To complete this lab assignment, I had to rewrite some of the given code to compile correctly. At the same time, I also swapped the bodies of the main() and sigHandler() functions and adjusted some of the printf() messages to be more readable. Lastly, in an attempt to enhance my own experience, I also wrote a new file that will compile on Windows rather than Unix. Because I have both of them in the same Visual Studio project folder, I added some preprocessor macros to ensure the right file gets compiled depending on the operating system I am actively using. The output and behavior of the files are identical, though I had to substitute the Unix pause() function with an infinite while-loop in the Windows program, as the Windows equivalent, system("pause"), does not behave in the same way when ran in Windows Powershell. Both files, “sampleProgram1Unix.C” and “sampleProgram1Windows.C” will be handed in the same folder as this document.

When either program is run on their respective systems, the program prints “Waiting…”. It will not print anything else until CTRL and C which sends an interrupt signal to the program so that it prints: Received an interrupt signal. Going to sleep. Notice the process is not exiting immediately.

The program then goes to sleep for 1 second before printing:

Out of here.

Now the process will terminate with the exit system call.

The program then exits.