CS4023 – Lab Exercise, Week 7

Implementing Kernel Threads on Linux

The goal of this exercise is understand how to use the system call clone() fro creating kernel threads on Linux.

Step 1. Load http://evanjones.ca/software/threading.html and find the section *Implementing Kernel Threads on Linux* in that webpage.

Step 2. Read the text above Figure 3.

Step 3. Copy the source code in Figure 3 and save it as **`example1.c`** in your home directory.

Step 4. Compile **`example1.c`** and build an executable **`a.out`** with the **gcc** compiler.

Step 5. Run the executable. What does it print on the screen? Can you explain the output?

Step 6. Analyse the source code. Search for info/help about any system or library function you do not understand. For example, make sure you understand **`malloc()`**, **`perror()`**, **`waitpid()`** as well as all arguments supplied to the **`clone()`** system call.

> **Hitn:** Use the **man** command (on the command line) followed by the name of the function you do not understand in order to get help.

Step 7. Modify the code in the following manner:

a. Create another thread associated with the same function **`threadFunction()`** and make the parent process (i.e. the **`main()`** function in this example) wait for both threads to complete their execution.

b. Declare a global variable **`int x`**. Add **`x = 0`** in the **`main()`** function before creating the threads. Then add **`x = x + 1`** into the **`threadFunction()`** and change the **`printf()`** call to **`printf( "x = %d\nchild thread exiting\n", x )`**.

Step 8. Repeat steps 4 and 5.

IMPORTANT: In your own time write a report that describes your work at steps 5 – 7. This summary will become part of your end-of-semester project.