CPSC 213: Assignment 8

Due: Sunday, November 14, 2010 at 6:00pm.

Goal

This is the first of two assignments that deal with threads; Assignment 9 will also include synchronization. In this first assignment you are given an implementation of user-level threads (i.e., threads implemented in application code). You will read this code and add comments, extend the code, and use it in a program.

Notes

The uthreads package runs on Intel x86 machines running Linux, MacOS or Cygwin. You can use the department linux machines by connecting to lulu.ugrad.cs.ubc.ca.

To compile on Linux or Cygwin it is necessary to explicitly include the pthread library by adding "-lpthread" to the gcc command line.

Requirements

Here are the requirements for this week's assignment:

1. Compile the files ping_pong.c and uthread.c. Run the resulting program
   ```
   gcc -c -o uthread.o uthread.c -lpthread
   gcc -o ping_pong ping_pong.c uthread.o -lpthread
   ```
2. Read uthread.c carefully. Describe the control-flow path involved in creating and starting a new thread by listing the uthread procedures that execute, in order, starting with the creation of a thread and ending when the thread's start procedure begins executing.
3. Execute the ping_pong program and examine its output. Carefully explain this output by describing the execution of the ping and pong threads. Your explanation should be detailed and should include a description of control flow paths (i.e., procedure names executed) in the uthread package relevant to explain the execution of these two threads.
4. Modify the ping_pong procedure to add a call to uthread_yield in both ping and pong at the end (but inside) of the iteration loop (i.e., just after the "j" for loop, but inside of the "i" for loop). Run this modified program, examines its output and compare it to the previous output. Explain what you see by again describing the execution of the two threads in a detailed fashion including the relevant uthread-thread control flow.
5. Modify the ping_pong procedure to change the argument to uthread_init() from 1 to 2. Run ping_pong again (you should run it at least 3-5 times; they should be different from each other), compare and explain its output as you did in question 4. Carefully explain what this change did and why the procedure gave the output it did.
6. Implement the problem from Assignment 7 using threads instead of calls to doAsync. Change the "Triple" struct to remove the "result" field and now have the add and sub routines return the resulting value. Use uthread_join to get this value and to synchronize

the three phases of the computation (the inner additions, the subtracting, and outer addition).

Material Provided

The files uthread.h, uthread.c and ping_ping.c are provided in the file code.zip.

What to Hand In

Use the handin program. The assignment directory is **a8**.

1. Your answers to questions 2-5, including samples of ping_pong output for questions 3-5.
2. Your Assignment-7 program implemented with uthreads.