

CS 4210

Homework #1

Assigned: May 22, 2003

Due: May 29, 2003

This is a small mini-project/homework to refresh your Pthreads programming background to get warmed up for your (larger) projects to be assigned later in this class.

Consider that there are n threads, numbered from 0 to $n-1$. The exercise is to compute the $\text{sum}(n)=0+1+\dots+(n-1)$, by computing & adding to partial sums at each thread.

This is accomplished by making each thread i do the following:

1. Thread i goes to sleep until awakened by thread $i-1$.
2. Thread i then retrieves the value of $\text{sum}(i-1)=0+1+\dots+(i-1)$ from shared area.
3. Thread i then computes $\text{sum}(i)=\text{sum}(i-1)+i$, and places it in the shared area.
4. Thread i then awakens thread $i+1$, and itself goes to sleep.

Obviously, thread 0 is a special case, which you need to consider suitably. It doesn't go to sleep initially, but does go to sleep after it sends its value to thread 1. When thread 0 is awakened by thread $n-1$, it retrieves & prints the final sum and terminates all the threads before exiting.

Use the C language and Pthreads library to implement multi-threading. Use Pthreads conditional variables to implement thread sleeping/awakening. Give some thought to the number of mutex variables that you might want to use (e.g., one pair of (Mutex, Condition) per thread-pair, or any other alternative design).

Ensure the intermediate/partial sum values are actually read from shared area (although it is trivial for a receiving thread to compute it itself).

Turn in your well-commented, concise, yet readable, C language source code, by email to Ryan (TA) at rcollins@cc.gatech.edu. Also turn in a table of timing information on how long the program takes to run from start to end, when n is varied from 1 to 64.

You are free to compile/debug/test/run on any Linux or Solaris machine of your choice. The submitted code, however, will be graded using gcc on a Solaris machine (most probably a multi-processor), so you will need to ensure it will compile/run on the grading platform without errors.

Useful Link(s):

Sun online documentation "Multithreaded Programming Guide" with Pthreads:

<ftp://docs-pdf.sun.com/806-5257/806-5257.pdf> or

<http://docs-pdf.sun.com/806-5257/806-5257.pdf>.