WENTWORTH INSTITUTE OF TECHNOLOGY

College of Engineering and Technology
Department of Electrical Engineering and Technology

Operating Systems
Spring 2018

Lab 7

Just like in lab 5, write a program that uses the pthreads library to create a secondary thread of execution, such that the thread function and the main routine have a common variable (you may name if `counter`) that is shared between them. The main routine should have a loop that increments the `counter` `n` times, where `n` is a parameter passed to your program during its invocation from the command-line. While the main thread is running its increment loop, the secondary thread shall be decrementing the shared `counter` variable `n` times and then exit. After the main thread has incremented the `counter` `n` times, it shall wait for the secondary thread to exit, and then print the value of `counter` to the screen.

In contrast to lab 5, you are required in this lab to protect the critical section using the following synchronization methods:

1. Using the Peterson's method.
2. Using a mutex (you may create a variable of type pthread_mutex_t and use the functions pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_lock and pthread_mutex_unlock.

In both methods, you need to compute the amount of CPU time the secondary thread has consumed using the function clock_gettime. You may use the man command to get more info about this function.

You shall run your program multiple times (let's say 10) for each value of `n` = 100,000,000 and take the average for the two synchronization methods.

Submit a text file showing the average CPU time for the secondary thread for the two methods and answer the following questions:

1. Did the two methods roughly consume the same CPU time for the secondary thread?
2. Why?

What to hand in (using Blackboard):

- Your ".c" file(s) (with appropriate comments). Do not attach project or make files.
- A screen shot of your terminal window(s) for one of the runs for each of the 2 methods.
- A document containing the results and answers to the lab questions.

RULES:

- Submit only .c, .h, image or document files. Do not submit .zip files or files with no or unknown extensions.
- Each group may consult with other groups/students about GENERAL concepts or methods, but copying code (or code fragments) or algorithms is NOT ALLOWED and is considered cheating (whether copied form other students, the internet or any other source).
- Each member of a group is required to contribute, and will be required to explain and defend every part of work done.

- Only one set of files should be submitted for each group.
- **To get full credit, you must attend the lab, show me your progress before you exit the lab (this goes for every student in the group), and submit required files before the posted deadline.**