CSE 4300 Programming Assignment 2
Assigned: Mar 13, 2019
Due: Mar 28, 2019, 11:59 pm (Midnight)

The purpose of this assignment is to introduce you to the following concepts:

1. C programming in Linux environment
2. Writing Multithreading programs
3. Use of synchronization primitives

Suggested Reading for this assignment (Please note that you can find many other great tutorials by simply searching on Google!).

https://computing.llnl.gov/tutorials/pthreads/
(Please read this! You can also find sample code here!)

You may find the followings useful for part of this project –

- pthread_mutex_t *mutex;
- pthread_cond_t *condition;
- pthread_create ();
- queue data structure (This is not part of pthread!)

Given the proliferation of multicore architecture, learning multithreaded programming is of utmost importance these days. Hence, the goal of this part of the assignment is to learn multithreaded programming. We will also learn how to synchronize among multiple threads using mutex.

Please note that, to compile multithreaded program, you need to specify "-lpthread" as follows

```
gcc -o output.txt program.c -lpthread
```

Part 1: (50 points)

The goal of this assignment is to write a multithreaded program that explores synchronization challenge. For this part, assume that, we have a shared variable CurrentID. This is initialized to 1 at the beginning.

Now create 5 threads in your program and assign ID 1,2,3,4,5 to them respectively. You can pass the ID as a parameter when you create the threads.

Each of the threads will try to access the variable "CurrentID".

Whenever a thread acquires the variable, it checks whether the CurrentId is equal to its own Id or not.

If it is not equal, it will output "Not My Turn!", then print its threadId, and then release the variable.

If it is equal, the thread will print "My turn!", then print its threadId, increase the CurrentId by 1, and then release the variable. However, after increasing CurrentID by 1, the thread will check if the value is 6 or not. If it is 6, it will reset it to 1 before releasing the variable.

The program should execute until each thread prints "My Turn!" 10 times. Once a thread prints for 10 times, it terminates.

Count the number of times each thread prints "Not my Turn!" and include that in the report.

Part 2. (50 points)

In this part, you are going to implement the producer-consumer problem using pthreads. After implementing, let the program run until at least 200 items are consumed by the consumer.

You are going to create 2 producer threads and 3 consumer threads in the program.

Repeat the experiment for the following settings:

1. Make the queue size 5
2. Make the queue size 10
3. Make the queue size 25

For each setting, count the number of times the producer and consumer goes to sleep because of empty queue or full queue. **Print the total number of times each goes to sleep at the end of the experiment. Include this in your report.**

What to submit

Submit source code file as part1.c and part2.c (you will get points off if filenames are incorrect)
Submit the output results and explanations of your code in a report file. You can copy paste your code in the file if you want.