Project 3: Race Condition and Semaphore **CS 300, Hong**

Project description

In this project, you will use threads to perform a counting task. The counting task is carried by two threads, counting over a shared variable. The program to do the work, os-p3.c, is attached, which is modified from t1.c from OSTEP. The maximum for performing the counting is taken from input. Please notice that in os-p3.c, a number 1000 is used to scale up your initial input. Each thread will count up to the scaling number multiplied by the input. Printf() is used to show counter values and addresses of a few variables.

Analyze the outputs. If you see a problem, modify the program by using a mutex. Then, run your modified code, also analyze the outputs.

You will write a report to describe the problem you observed, explain the reasons, and explain your solution. Include your name and course # in the report. The report should be one page in letter size, single space, maximum font size 11. If figure is used, its size must be smaller than 1/6 of the page size. Two-column format is welcomed. 5% points will be taken off if formatting is not correct, or it is over 1 page.

Additional requirements

1) Name your report as *YourFirstLastName.pdf* (.docx is fine too). Include your name and course # in the report.
2) Name your code as your *YourFirstLastName.c.* include comment lines at the beginning to show your full name and the project info.
3) Compile your code using gcc -Wall filename -std=c99 -pthread at the cs-intro.ua.edu server.

Submission requirements:

4) Submit to Blackboard "Proj 3" before deadline.
5) Submit your report and source code
6) Grading will use gcc -Wall filename -std=c99 -pthread at the cs-intro.ua.edu server. You should try to use the same command at the same server to compile your code. Errors leads to points drop.
7) 10% points are allocated to the submission and compiling.

Delay policy:

Follow the course policy. Each one-day delay will drop 20% points. You should start your projects early to avoid potential last minute issues that cause late turn-in.

Grading will count:

(1) Submission and Compilation; (2) Outputs from your code; (3) Code inspection; (4) Correctness and clarity of your report.