COP 6611
Program 2

Due: Feb. 22, 2017 at 9:00.m.

Implement a solution to the critical section problem with threads using semaphores. You will modify the same code as in the previous project, but you must add a third counting thread which counts by 1 each time it enters its critical section to 2,000,000. Each counts to 2,000,000 for a total of 6,000,000. You will need to read about semaphore operations from linux manual pages. The important ones are sem_init, sem_wait, and sem_post. You can find pages online at:

`http://man7.org/linux/man-pages/man3/sem_init.3.html`

`http://man7.org/linux/man-pages/man3/sem_wait.3.html`

`http://man7.org/linux/man-pages/man3/sem_post.3.html`

Please compare this solution with the earlier mutex lock solution in terms of time used. Just compare the two thread version (though you hand in with 3 threads). Use the same time commands included with the last project. In your comments when you hand in compare the run times. Explain as best you can why they are the same or different in your opinion.

You will hand in the program on netcluster.cse.usf.edu by running /usr/local/os/turn_in3 and carefully following the instructions. Make sure your name is in the code file in comments!