COP 6611
Busy wait Critical Section Solution

Due: Feb. 10, 2022 at 9:00.m.

Implement a solution to the critical section problem for 2 processes using mutex locks. Specifically, in pthreads using **pthread_mutex_trylock.** You are provided with code in threadlock22template.c in Canvas which does not work correctly, but has the necessary pthread Unix thread calls to start up two threads from the main process. Remember that all global memory is shared among threads of a process. Both threads need to count to 3,000,000 which means a correct overall count is 6,000,000. Thread1 needs to be modified so that every time it sees the shared value ($counter->value\%100) == 0$ it increments $counter->value$ by 100. That counts for 100 individual updates. You also need to keep track of how many times this happens and report it. Use the POSIX implementation of threads. You will need to look at the pthread_create, pthread_join and threads manual pages. A tutorial and the man pages are here:
http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html
information on trylock is here:

```
http://man.yolinux.com/cgi-bin/man2html?cgi_command=pthread_mutex_trylock&cgi_section=
```

Time your code with getrusage. This is for a comparison project. It will look something like the following. The first four lines are what you need to set this up.

```
#include <sys/time.h>
#include <sys/resource.h>
struct rusage mytiming;
struct timeval mytimeval;
getrusage(RUSAGE_SELF, &mytiming);
printf("Time used is sec: %d, usec %d\n",mytiming.ru_utime.tv_sec,
mytiming.ru_utime.tv_usec);
printf("System Time used is sec: %d, usec %d\n",mytiming.ru_stime.tv_sec,
mytiming.ru_stime.tv_usec);
```

You will hand in the program on osnode16.cse.usf.edu by running /usr/local/os/turn_in2 and carefully following the instructions. Make sure your name is in the code in the comments! Again note this is an individual project and must be your own code. If you use any other code (besides that given by me), it must be acknowledged in comments. Failure to do so is academic dishonesty and is dealt with as described in the syllabus.