CCPS590 Lab 5 – Concurrency with Mutexes

Preamble

For most of you, this week's lab will be your first exposure to the difficulties of concurrent programming. It is not easy, and it does not come naturally. Personally, I found the leap from serial to concurrent programming to be similar to the leap from iteration to recursion. Take your time with this week's lab. Make sure you solve it correctly, but more importantly make sure you truly *understand your solution*. Solving a problem does not necessarily mean you understand your solution (or the problem).

Lab Description

1) This lab requires some light (I promise) background reading on your part. Read about **`pthreads`** and **`mutex`** usage at the following link:

http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html

It looks like a lot, but much of it is code and function descriptions. Use this website as a reference while you're working on this lab. The example program named cond1.c (posted with this lab description) is from this website also. Look at cond1.c. Compile it, run it, understand it. You know the drill by now. Write a program named lab05.c that is like cond1.c except that:

thread1: increments count only when count is currently an even number
thread2: increments count only when count is currently an odd number

Further Requirements:

- A thread prints the value of count AFTER the increment is performed.
- Threads terminate when count reaches COUNT_DONE
- After threads terminate, **`main()`** prints the final count (like cond1.c does).
- Thread2 may test count to determine its parity (even/odd), but thread1 may not.
- Your program must terminate correctly. No child thread should be left hanging.
- Your lab05.c output should look *exactly* like the output on the next page.

```
Counter value functionEven: 1
Counter value functionOdd:  2
Counter value functionEven: 3
Counter value functionOdd:  4
Counter value functionEven: 5
Counter value functionOdd:  6
Counter value functionEven: 7
Counter value functionOdd:  8
Counter value functionEven: 9
Counter value functionOdd:  10
Final count: 10
```

Submission

For this lab you will submit one file - your code lab05.c. Your code must compile and run out of the box. I will not fix syntax errors. Be sure to test it out on an SCS moon!

Labs are to be submitted ***individually***! Make sure your code is clean and easy to read.