CS5323 – Operating Systems II
Programming Assignment 2
Due: March 20, 2023, 11:59 p.m.
Submission: via Canvas

Implement a solution via semaphores and threads to the n reader 1 writer problem. Fairness always matters. You will accept the number of readers from the command line. In no case will more than 14 readers be used and always at least 1 reader will be used. Each reader must access a shared counter value 250000000 times in the critical section. Note, it does not update anything, just "reads". For convenience code is below that will do this. A reader reads just one time and a writer writes just one time. Each reader needs to print its name when done. The writer will update the value 25000 times and print done. The writer will also set a shared flag, in-cs, when it enters the critical section and reset it just before it leaves the critical section. The reader must, upon entering the critical section, check this flag and write an error message if the flag is set. You can help us out for testing by using a version of the following code to give the writer a chance to run while readers are also running. So, start it in the midst of the readers.

```
k = (int) (numOfReaders/2);
for(i = 0; i < k; i++)
{
    pthread_create(&readers[i], &attr[0], reader_thread,
(void*) i);
}
/* Create the writer thread */
pthread_create(&writer[0],   &attr[0],   writer_thread,
NULL);
for(i = k ; i < numOfReaders ; i++)
{
    pthread_create(&readers[i], &attr[0], reader_thread,
(void*) i);
}
void relaxandspendtime()
{
    int i;
    for(i = 0; i < 250000000; i++) i=i;
}
```

Deliverables:

1. **Well-documented, readable code written in C. Clearly mark the entry section, critical section, exit section and remainder section in your code. You code will be executed on CSX so ensure that it works on there.**
2. **A README file with instructions on how to run your code.**
3. **A short report (~1 page) on how the synchronization logic works in your implementation and a justification on your choice between mutexes and semaphores.**
4. **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, it must be acknowledged in comments.**