CS4375: Theory of Operating Systems **Instructor: Dr. Deepak Tosh**

Fall 2021 **Due Date:** 10/29/2021, Midnight

Assignment-4a (Concurrency)

The goal of this assignment is to allow you exploring the concurrency abstractions like threads and locks, which are vastly used in real life software applications. We will start with two warm up exercises and continue building a Master-Worker Thread pool that would implement allow you to understand how real-world systems like webservers and database servers work.

For this assignment, I would encourage you to use C program to develop your program.

Task -1

Write a program that has a **counter** as a global variable. Spawn **T** threads in the program, and let each thread increment the counter **N** times in a loop. Print the final value of the counter after all the threads finish—the expected value of the counter is **N*T**. Make sure you read the value of **N** and **T** from the user.

- First, run this program first without using locking across threads and observe the incorrect updation of the counter due to race conditions (the final value will be slightly less than 10000).
- Then, use locks when accessing the shared counter and verify that the counter is now updated correctly.
- Test your program with different # of threads and counter values.

Task -2

Write a program with **N** threads. **N** can be taken as an argument from the main program. Thread i must print number i in a continuous loop.

- First try without any synchronization between the threads, the threads will print their numbers in any order.
- Then, add synchronization to your code using locks such that the numbers are printed in the order 1, 2, ..., N, 1, 2, ..., N, and so on.

Note: You may want to start with N = 2 and then move on to larger values of N.

To submit:

a. Programs for both with appropriate documentation and instruction to execute your programs
b. A report on demonstrating your program's execution and correctness. Also mention any references that you used while developing your program.

Grading:

The assignment will be graded on following items:

1. Completeness and correctness on your responses, explanations, and observations.
2. Inclusion of appropriate evidence (in form of screenshots)
3. Clarity of Report with all the required responses
4. References (including links where you found some sample code.