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

Fall 2021 **Due Date:** 11/13/2021, Midnight

Assignment-4c (Concurrency-Semaphores)

The goal of this part is to allow you exploring the applicability of semaphores in solving two simple synchronization related problems. Please use the starter codes (in StarterCodes.zip) to implement solutions for following two problems.

Problem#1 (Randezvous Problem)

The problem is as follows: you have two threads, each of which are about to enter the rendezvous point in the code. Neither thread should exit this part of the code before the other enters it. Consider using two semaphores for this task, and see ***rendezvous.c*** for details.

Problem#2 (Barrier Synchronization Problem)

Now, you need to go one step further by implementing a general solution to **barrier synchronization**. Assume there are two points in a sequential piece of code, called P1 and P2. We are needed to put a barrier between P1 and P2, which will guarantee that all threads will execute P1 before any one thread executes P2.

Your task: write the code to implement a *barrier()* function that can be used in this manner. It is safe to assume you know **N** (the total number of threads in the running program) and that all N threads will try to enter the barrier. Again, you should likely use two semaphores to achieve the solution, and some other integers to count things. See ***barrier.c*** for details

To submit:

a. Programs with appropriate documentation and instruction to execute
b. A report on demonstrating your program's execution and correctness. Also mention any references that you used while developing your program.
c. Discussion on challenging aspects of this assignment.

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
4. References (including links where you found some sample code).