

## CS403 Parallel Programming

### Lab02 - Synchronization

Topics covered:

1. Mutex
2. Conditional Variable
3. Producer-consumer

#### Lab problems:

1. Familiarize yourself with `rw_lock` and `barrier` code [in the sample code folder](#). Run the code and attach the screen-shots with your observations.
2. For the given serial code (`dotprod.c` in the sample code folder), write the equivalent parallel code. Using the `time` command, measure the execution time and corresponding speed-ups for:
  - a. vector length = 100,000 and 200,000
  - b. number of processors = 2, 4 and 8

Execution Time				
	$p = 1$	$p = 2$	$p = 4$	$p = 8$
Vector Length = 100,000				
Vector Length = 200,000				

$$\text{Speedup} = \frac{\text{Execution Time } (p)}{\text{Execution Time (serial code)}}$$

Speed-up			
	$p = 2$	$p = 4$	$p = 8$
Vector Length = 100,000			
Vector Length = 200,000			

#### 3. Multi-access threaded queue

- a. Implement a multi-access threaded queue with multiple threads inserting and multiple threads extracting from the queue. Use mutex-locks to synchronize access to this queue. Document the time for 1000 insertion and 1000 extractions each with 4 insertion threads (producers) and 4 extraction threads (consumers).
- b. Repeat above problem with condition variables (in addition to mutex locks). Document the time for the same test case as above. Comment on the difference in the times.

**Lab-report:**

1. Numerical problem and solution
2. Programming problem
  - Objective or summary of problem statement
  - Pseudo-code
  - Measurements / results
  - Conclusion

**Attachments:**

- a) Lab report
- b) dotprod\_parallel.c
- c) producer\_consumer\_mutex.tar.gz
- d) producer\_consumer\_conditional\_variables.tar.gz

NOTE: (1) In your lab-report, attach screenshots whenever necessary.

- (2) For the programming questions, include your pseudo-code (in lab-report) and C-code *separately*.