First Course Handout

Course Title: Analysis of Concurrent Programs
Course No: CS 636
Credits: 3-0-0-0-[9]

Prerequisite:

- Exposure to CS 330 (Operating Systems), CS 335 (Compiler Design), and CS 422 (Computer Architecture) (or equivalent non-IITK courses) is desirable.
- Programming maturity (primarily C/C++/Java) is desirable.

Lecture Hours: MW 5:15-6:30 PM (online, synchronous)

Course Objective: This course will focus on the challenges in developing correct and scalable concurrent programs. We will discuss the types of errors that we make while writing concurrent programs, and techniques and tools to debug them. We will learn about techniques and abstractions that help write correct concurrent programs, for example, synchronization primitives and transactional memory. We will also discuss testing of concurrent programs, and ideas to avoid common performance bottlenecks.

The course will include programming assignments and a course project.

Course Contents: The course will primarily focus on the following topics.

- Concurrency Bugs: data races, atomicity violations, and deadlocks
- Shared Memory Synchronization: locks, monitors, semaphores, flags, barriers, condition variables
- Concurrent programming paradigms: shared-memory, message-passing, partitioned global address space
- Transactional Memory
- Memory Consistency Models
- Concurrent Data Structures
- Testing of Concurrent Programs
- Performance Challenges: PAPI counters, performance analysis tools, false sharing

- We might add new, drop existing, or reorder topics depending on progress and class feedback
- The course may also involve reading and critiquing related research papers
- The recommended reading material is part of the course material

Evaluation:

Class participation/quizzes	10%
Assignments/paper critiques	30%
Term Project	35%
Endsem	25%

- This is a tentative allocation
    - Might change allocations depending on the strength of the class
- Grading will be relative

References:

1. The Art of Multiprocessor Programming - Maurice Herlihy et al. (either one of editions 1 and 2 should do)
2. A Primer on Memory Consistency and Cache Coherence - Daniel J. Sorin et al.
3. Shared Memory Synchronization - Michael S. Scott
4. Transactional Memory - Tim Harris et al. (2nd edition)
5. Parallel Computer Architecture: A Hardware/Software Approach - D. Culler, J, Singh with A Gupta

We will also distribute relevant handouts and research papers.