First Course Handout

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

Lecture Hours

TuThF 8:00-9:00 AM in KD 102

Instructor

Swarnendu Biswas,
Assistant Professor,
KD 302, Department of CSE.
Email: swarnendu@cse.iitk.ac.in
Phone (O): +91-512-259-2055

TAs

Srinjoy Sarkar	srinjoys23@cse.iitk.ac.in
Sahil Basia	sahilbasia24@cse.iitk.ac.in
Binong Kiri Bey	binong@cse.iitk.ac.in
Vishal Ramdas Junjare	vrjunjare25@cse.iitk.ac.in

Course Objective

This course will focus on the challenges of developing correct and scalable concurrent programs. We will discuss the errors we make when writing concurrent programs and the techniques and tools to debug them. We will learn about techniques and abstractions that help write correct concurrent programs, such as synchronization primitives and transactional memory. Furthermore, we will discuss testing of concurrent programs and ideas to avoid common performance bottlenecks.

We will have 2-3 programming assignments and a course project.

Prerequisite

- Exposure to the following courses (or equivalent) is desirable: CS220 (Computer Organization) and CS330 (Operating Systems).
- Programming maturity with popular programming languages like C, C++, and Java

Course Contents

The course will focus on a subset of the following topics.

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

We may add new topics, drop existing ones, or reorder them depending on progress and class feedback. The course may also involve reading and critiquing related research papers.

Policies

- Please be on time for class.
- Please try to avoid using laptops and/or mobile devices in class, as they can be distracting for both the instructor and other students who want to concentrate.
- You are allowed up to **two late** days to submit your assignment, with a 25% penalty for each day.
- Make-up for the midsem and endsem exams will be in accordance with the Institute policy.

Evaluation

The following is a tentative allocation that may change slightly depending on the strength of the class. Grading is relative.

Class participation (e.g., interaction and quizzes)	5%
Assignments	15%
Midsem	25%
Course project	25%
Endsem	30%

We will take attendance in the class as it is now mandated by the Institute. Attending all classes is not mandatory, but is recommended. There is no minimum attendance requirement to appear in the midsem and endsem exams. However, I strongly encourage you to attend classes because:

- There are no standard textbooks that cover the topics that we will discuss in class
- There is 5% allotted based on performance in class

References

I have listed (NOT in any particular order) a few popular references.

1. The Art of Multiprocessor Programming, 2nd edition - M. Herlihy et al.
2. A Primer on Memory Consistency and Cache Coherence, 2nd edition - V. Nagarajan et al.
3. Shared-Memory Synchronization, 2nd edition - M. Scott and T. Brown
4. Transactional Memory, 2nd edition - Tim Harris et al.
5. Parallel Computer Architecture: A Hardware/Software Approach - D. Culler et al.
6. Java Concurrency in Practice - B. Goetz et al.
7. Operating Systems: Three Easy Pieces - R. Arpaci-Dusseau and A. Arpaci-Dusseau.