CSE 480: Distributed and Real-Time Systems

Spring 1996

Instructor: Insup Lee

Time: 12:00 - 1:30 Tue & Thr

Course Description: Ever increasing availability of inexpensive processors connected by a communication network has motivated the development of numerous concepts and paradigms for distributed systems. The primary objectives of this course are to study the principles and concepts of distributed computing and to provide students hands-on experience in developing distributed applications. The students will learn about the nature of distributed systems including their problems and solutions, as well as theoretical foundations for the design and implementation of distributed systems. The topics to be covered in the course are: models of distributed systems, distributed algorithms and protocols, operating systems support, programming paradigms, real-time systems, and case studies of experimental and commercial systems.

The course requires active student participation in group projects. Groups of no more than three students will be responsible for the design and implementation of a collection of distributed applications such as distributed leader election, naming, group membership, multicast communication, video/audio conferencing. The group projects are intended to complement the learning of principles and concepts through the application of theory in practice and the development of experimental skills in building distributed applications.

Prerequisites: CSE 380

Text Book: *Distributed Operating Systems* by Andrew Tanenbaum, Prentice Hall, 1995.

CSE 480 Course Outline

- Introduction to Distributed Systems: architecture, issues, communication network.
- Concepts and Theoretical Foundations: inherent limitations, logical clocks, global state, cuts, distributed snapshots.
- Mutual Exclusion: requirements, Lamport's algorithm, Ricart-Agrawala algorithm, Maekawa's algorithm, Token-based algorithms.
- Agreement Protocols: synchronous versus asynchronous computations, failure models, the Byzantine agreement problem, the consensus problem, the interactive consistency problem, impossibility results, applications.
- Names: bindings, interpretating names, consistency.
- Interprocess Communication: remote procedure call, system issues.
- Distributed File Systems: design issues, case studies.
- Distributed Shared Memory: implementation models, memory coherence protocols, granularity, page replacement, case studies.
- Distributed Scheduling: load balancing, stability, load distribution algorithms, task migration, case studies.
- Fault-Tolerant Communication Protocols: commit protocols, voting protocols, failure resilient processes, reliable communication.
- Data Security: model of cryptography, private key cryptography, public key cryptography, mutual encryption, authentication.
- Real-Time Systems: real-time applications, predictability, real-time communication, priority inversion problem, dynamic versus static scheduling.