Week 1

Week 2

Week 3

Week 4

* What is tightly coupled multi-processing?
* What are the different classes of multiprocessor parallelism based on synchronization granularity?
* Different design issues involved in scheduling on multiprocessor and their consequences.
* How process scheduling is done in multiprocessor environment?
* Why thread scheduling is different than process scheduling in multiprocessor environment?
* Different approaches to thread scheduling and their relative advantages/disadvantages.
* What are the additional issues to be considered in multi-core thread scheduling?
* What are the characteristics/properties of Real-time systems? What are different types of real-time systems?
* What are the differences between periodic and aperiodic real time tasks?
* What are different approaches to real time scheduling?
* Different real-time scheduling algorithms: EDFS, RMS. Their relative advantages/disadvantages.
* What is priority inversion? What is unbounded priority inversion? What are the approaches to avoid unbounded priority invesion

Week 5

* Different contexts in which concurrency can arise
* Interleaved concurrency and overlapped concurrency
* The difficulties of concurrency
* What is race condition?
* Different ways of process interaction: competition, coordination and communication and
* Concurrency problems posed by different types of process interactions
* General approach to mutual exclusion
* The requirement for mutual exclusion
* Hardware supported approaches to mutual exclusion: interrupt disabling, special machine instruction
* Advantage & disadvantage of hardware methods to mutual exclusion
* Semaphores: definitions, types (weak VS strong; binary VS counting), operations and usage
* Mutual exclusion using semaphores: example/application
* Implementation of semaphores
* Monitor: definition, components, mechanism
* Relative advantage/disadvantage of semaphore and monitors
* Difference between Mesa and Hoare Monitors
* Mutual exclusion using monitors: example/application
* Classic synchronization problems: Producer/Consumer Problem, Readers/Writers problem. Solution to these problems using monitor/semaphores

Week 6

* Software approaches to mutual exclusion
* Dekker’ algorithm/Peterson algorithms
* Understand the flaws in alternate software solutions presented in lecture slides
* Be able to justify a software solution to mutual exclusion is correct or wrong
* What is a deadlock? Difference between deadlock and livelock.
* Resource categories: reusable and consumable
* Resource allocation graphs
* Necessary and sufficient conditions for deadlocks
* Deadlock handling mechanisms: prevention, avoidance and detection and recovery
* Different deadlock preventions strategies and their limitations.
* Deadlock avoidance strategy.
* Banker’s algorithm for deadlock avoidance:
* Deadlock detection mechanism/algorithm
* Different deadlock recovery strategies and their relative advantage/disadvantages.
* Why we need an integrated approach to deadlock handling?
* Dining philosopher problem and solution with mutation/semaphore