

Please consider the associated Links and References (Canvas, On-Line) provided in the class schedule.

Main topics:

- **Ch 6&7: Process Synchronization**

- Shared memory and process synchronization
- Race Condition
- Critical Section Problem
 - Entry section, Critical Section, Exit Section, and Remainder Section
- Critical-section problem solution
 - Requirements
 - Mutual Exclusion
 - Progress
 - Bounded Waiting
 - Solutions
 - Software-based solutions
 - Turn-based,
 - Flag-based,
 - Peterson
 - Hardware-based solutions
 - Disabling interrupts
 - Test and Set instruction
 - Exchange/Swap instruction
 - Operating system solution (semaphore)
 - What is a semaphore? Types of semaphores (binary, counting/general)
 - wait() and signal()
 - busy waiting and spinlock
 - Semaphore Implementation
 - waiting queue
 - block and wakeup system calls
 - Classic Problems of Synchronization
 - Producer-Consumer (Bounded-Buffer) Problem
 - Problems with Semaphores
 - Programming languages solution (monitor)
 - What is a monitor? Condition variables and wait(), signal()
 - Producer-Consumer Problem with monitor
 - Condition Variables Choices
 - Signal and wait (Hoare)
 - Signal and continue (Lampson/Redell)

- **Ch 8: Deadlocks**

- What is a deadlock in OS?
- Necessary conditions (Coffman) for deadlocks (Deadlock Characterization)
 - Mutual exclusion

- Hold and wait
- No preemption
- Circular wait
- System Model
 - Resource-Allocation Graph
 - Request Edge, Assignment Edge
- Methods for Handling Deadlocks
 - Ostrich
 - Deadlock Prevention
 - Deadlock Avoidance
 - Safe, Unsafe, Deadlock State
 - Banker's Algorithm
 - Deadlock Detection and Recovery