# Lecture Summaries

## l7

Processes can execute concurrently May be interrupted at any time, partially complete execution Concurrent access to shared data may result in data inconsistency . Maintaining data consistency requires mechanisms to ensure consistency of data . processes can be executed concurrently and may be interrupted, partially completed execution .

counter++ could be implemented as register1 = counter register1 + 1 counter = register1 . counter is set to 0. it is incremented by the producer after it produces a new buffer . it is decremented by the consumer after it consumes the item in next consumed .

producer execute register1 = register1 + 1 register1 = 6 S2: consumer execute register2 = counter counter = 4 counter-- could be implemented as register 2 = counter register2 - 1 counter = register2 Critical section problem is to design protocol to solve this critical section problem .

the selection of the processes that will enter the critical section next cannot be postponed indefinitely . a bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section .