Background	• Processes can execute concurrently ○ May be interrupted at any time, partially completing execution • Concurrent access to shared data may result in data inconsistency • Maintaining data consistency requires mechanisms to ensure the orderly execution of cooperating processes

Solution to Critical-Section Problem	• Mutual exclusion
	  ○ If process P_i is executing in its critical section, then not other processes can be executing in their critical sections
	• Progress
	  ○ If no progress is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely
	• Bounded waiting
	  ○ A bound must exist on the number of times that other processes are allowed to enter their critical sections

Critical-Section Handling in OS	• Two approaches depending on if kernel is preemptive or non-preemptive
	  ○ Preemptive – allows preemption of process when running in kernel mode
	  ○ Non-preemptive – runs until exits kernel mode, blocks, or voluntarily yields CPU
	  ▪ Essentially free of race conditions in kernel mode