
Synchronization

Chapter 6

Race Conditions

- **Race Condition:** *Any situation in which the outcome of execution depends on the particular order in which multiple processes/threads execute*
 - **Prevention:** Implement Critical Sections
-

Critical Section

- Requirements:
 - Mutual Exclusion
 - Progress
 - Bounded Waiting
 - No Timing Assumptions
 - Classic Solutions:
 - Peterson's Solution
 - TSL Instructions
-

Semaphores

- ▣ Abstracts details of critical section to simplify programming
 - ▣ wait() / P
 - ▣ signal() / V
 - ▣ Variations:
 - ▣ Counting
 - ▣ Binary - (Mutex)
-

Semaphores

- Implementation
 - Spin Lock
 - Efficient Waiting

Programming

- POSIX: **(make sure you look at the code posted!)**
 - Supports mutex through pthread library
 - Semaphores are directly supported
 - Windows:
 - Support for mutex, semaphores directly
 - Java: Supports Mutex, Semaphore, and Monitor
-