

Concurrency: Mutual Exclusion and Synchronization

Chapter 5

Definitions

- ❁ critical section: a section of code which reads/writes shared data
- ❁ race condition: potential for interleaved execution of a critical section by multiple threads => results are non-deterministic
- ❁ mutual exclusion: synchronization mechanism to avoid race conditions by ensuring exclusive execution of critical sections
- ❁ deadlock: permanent blocking of threads
- ❁ starvation: execution but no progress

Conventional solutions for ME

- ❁ software reservation: a thread must register its intent to enter CS and then wait until no other thread has registered a similar intention before proceeding
- ❁ spin-locks using memory-interlocked instructions: require special hardware to ensure that a given location can be read, modified and written without interruption (i.e. TST: test&set instruction)
 - ❁ they are equivalent !
- ❁ OS-based mechanisms for ME: semaphores, monitors, message passing, lock files
 - ❁ they are equivalent !