CSMC 412

Operating Systems

Prof. Ashok K Agrawala

Set 6

Process Synchronization

- Background
- The Critical-Section Problem
- Synchronization Hardware
- Semaphores
- Classical Problems of Synchronization
- Monitors
- Java Synchronization
- Solaris Synchronization
- Windows XP Synchronization
- Linux Synchronization
- Pthreads Synchronization
- Atomic Transactions
- Log-based Recovery
- Checkpoints
- Concurrent Transactions
- Serializability
- Locking Protocols

Concurrency and Synchronization

- Concurrency
- The Critical-Section Problem
- Synchronization Hardware
- Semaphores
- Classical Problems of Synchronization
- Critical Regions
- Monitors
- Synchronization in Solaris 2 & Windows 2000

Systems = Objects + Activities

- Safety is a property of objects, and groups of objects, that participate across multiple activities.
  - Can be a concern at many different levels: objects, composites, components, subsystems, hosts, …
- Liveness is a property of activities, and groups of activities, that span across multiple objects.
  - Levels: Messages, call chains, threads, sessions, scenarios, scripts workflows, use cases, transactions, data flows, mobile computations, …

Violating Safety

- Data can be shared by threads
  - Scheduler can interleave or overlap threads arbitrarily
  - Can lead to *interference*
    - Storage corruption (e.g. a *data race/race condition)*
    - Violation of representation invariant
    - Violation of a protocol (e.g. *A* occurs before *B*)

How does this apply to OSs?

- Any resource that is shared could be accessed inappropriately
  - Shared memory
    - Kernel threads
    - Processes (shared memory set up by kernel)
  - Shared resources
    - Printer, Video screen, Network card, …
- OS must protect shared resources
  - And provide processes a means to protect their own abstractions