Processes & Concurrency
Process Scheduling

- Process vs. program
- Process States
  - Running
  - Ready (or Waiting)
  - Blocked
- Transitions between the states
- Scheduler
  - round robin
  - priority scheduling (fixed or dynamic)
  - A process can yield
Threads

• Multithreaded Processes
  • multithreaded process
  • share the address space
  • share the files

• scheduler schedules threads

• process states vs. thread states

• context switch

• thread control block vs. process control block
Interprocess Communication

- message passing
- shared memory
- Remote Procedure Call
- Pipe
- atomic modification
  - atomic read-modify-write
  - test-and-set
  - fetch-and-add
  - compare-and-swap
Mutual Exclusion

- race condition
- Mutual Exclusion
  - Mutual Exclusion
  - No deadlock
  - No starvation (or no lockout)
  - Unobstructed exit
- Synchronization
- Peterson’s Algorithm
  - Does not work on modern computers
Semaphores

• Semaphore
  • Wait
  • Signal

• Mutex
  • Locking
  • unlocking

• Counting Semaphore
Classic Problems

• Dining Philosophers Problem
  • Deadlock
  • even-numbered chopsticks first

• Producer-Consumer Problem
  • one process needs the output of another process to continue working
  • semaphores

• Readers-Writers Problem
  • Readers-writers problem: the possibility of writer starvation
    • readCount, readCountMutex
  • readers-Writers problem: the possibility of reader starvation
    • accessMutex
Monitors

• Monitor
• Active
• Block
• condition variables
  • conditionWait(Semaphore monitorMutex, Process P)
  • conditionSignal()}