Test 1 will be 11-12:15 PM on Thursday, March 1, in ITT 328. The test will cover chapters 1-4 and will be closed-book and closed-note, except for one sheet of paper with notes (you can write on the front and back). Review topics are:

Chapter 1. Introduction
Definition of the operating system and middleware; OS objectives and functions
CPU dual-modes of operation: user mode and system(/supervisor/monitor) mode with mode-bit(s) within the CPU's *processor-status-word* (PSW) register; privileged vs. nonprivileged instructions
CPU-timer; general idea of interrupts
I/O module/controller: operation and function
I/O instructions: memory-mapped I/O and isolated I/O
I/O techniques: programmed I/O, interrupt-driven I/O, and direct-memory access (DMA)

Chapter 2. Threads
Definition of program, process, and thread
Reasons for using concurrent threads: responsiveness, resource utilization, and tool for modularization
Switching CPU between threads and processes; yield statement
Preemptive multitasking

Chapter 3. Scheduling
thread state diagram; TCB (thread control block); cache impact performance; Gantt chart
performance scheduling goals: throughput, response time
control scheduling: urgency, importance, resource allocation (CPU)
fixed-priority scheduling: rate-monotonic schedule
dynamic-priority scheduling: Earliest Deadline First (EDF), decay-usage scheduling (e.g., Mac OS X / MS Windows)
proportional-share scheduling: weighted round-robin scheduling (WRR), weighted fair queuing (WFQ)/stride scheduling/virtual time round-robin scheduling (VTRR), lottery scheduling
Linux example: Completely Fair Scheduler (CFS); niceness level; virtual-runtime

Chapter 4. Synchronization and Deadlocks
Need for synchronization - races, mutual exclusion
pthread mutex API: pthread_mutex_init, pthread_mutex_lock, pthread_mutex_unlock, pthread_mutex_destroy, pthread_mutex_trylock, pthread_mutex_timedlock
Monitor: Java class with synchronized methods
Mechanism for implementing mutex: exchange operation
Mutex implementations: spinlock, cache-conscious spinlock, queuing mutex
Common synchronization patterns: bounded buffers, readers/writers locks, barriers
Condition variables used with monitors: Java usage (wait, notifyAll, notify)
Semaphores - an unsigned integer with operations restricted to: release (or V), acquire (or P)
Deadlock problem; necessary conditions for deadlock; resource-allocation graph
Deadlock prevention through resource ordering
deadlock detection
immediate deadlock detection
deadlock avoidance: Banker's Algorithm for Deadlock Avoidance
deadlock detection using banker-like algorithm
Interaction of synchronization and scheduling: priority inversion, convoy phonomenon - improved queuing mutex
Nonblocking synchronization/lock-free: compareAndSet operation