
CS 422/522 Design & Implementation
of Operating Systems

Lecture 20: Midterm Review

Zhong Shao
Dept. of Computer Science
Yale University

Acknowledgement: some slides are taken from previous versions of the CS422/522 lectures taught by Prof. Bryan Ford and Dr. David Wolinsky, and also from the official set of slides accompanying the OSPP textbook by Anderson and Dahlin.

The big picture

- ◆ OS roles: referee, illusionist, and glue (AD 1.1)
- ◆ Kernel and process abstraction (AD 2.1-2.4, 3.1-3.5)
 - Why process abstraction?
 - Dual-mode operation (privileged instructions; timer interrupts; memory protection)
 - Safe control transfer
 - Interrupts vs. exceptions vs. system calls
- ◆ CPU & concurrency (AD 4.1-4.8, 5.1-5.8, 6.5, 7.1-7.2)
- ◆ Memory management (AD 8.1-8.3, 9.1-9.6)
- ◆ I/O devices (AD 11, 12, 13, 14)

CPU & concurrency

- ◆ Thread vs. process
- ◆ How to implement threads/processes ?
 - * thread/process state transition diagram & context switch
 - * thread/process creation / finish & fork-join parallelism
 - * kernel vs. user threads
- ◆ How to write concurrent programs ?
 - * how to eliminate race condition ? how to synchronize?
 - * what is the "shared-objects" approach?
 - * what are locks, condition variables, monitors, and semaphores?
 - * how to use locks & condition variables to support synchronization?
 - * how to implement locks & condition variables on uni- & multi-processors?
- ◆ How to deal with deadlocks
 - * banker's algorithm
- ◆ Uniprocessor and multiprocessor scheduling

Memory management

- ◆ Address translation
 - segmentation + paging + multilevel paging
 - efficiency via TLB
 - virtually addressed vs. physically addressed caches
- ◆ Caching and virtual memory
 - cache concept & memory hierarchy (Figure 9.3)
 - when caches work: working set vs. Zipf model
 - cache replacement policies & Belady's anomaly
 - memory-mapped files

I/O devices

- ◆ File system abstraction & device drivers
- ◆ Storage devices
 - * magnetic disk access and performance
 - * various disk scheduling algorithms
 - * flash storage vs magnetic disk: how they differ?
- ◆ Files and directories
 - * how are they implemented?
 - * How Unix (FFS) file system works? What is an inode?
 - * FAT vs FFS vs NTFS (Fig 13.8)
- ◆ Reliable storage
 - * What is transaction? Why we need it?
 - * How to use redo-logging to implement transaction
 - * What are RAID1 and RAID5?