Operating Systems
ECE344

Lecture 13: Final Review

Ding Yuan
Announcements

• Piazza bonus points
  – Will keep “endorsing” students’ answers and questions until Final exam

• Additional office hours
  – Tuesdays 2-3PM
  – You can also send me emails for appointments

• Please keep checking Piazza for important announcements
Course Plugs

• If you like ECE 344 topics and you like programming, you might find some other courses interesting next year
  • ECE454: Computer System Programming
  • ECE419: Distributed Systems
  • ECE552: Computer Architecture
Overview

• Final mechanics
• What we have learnt
  – Scheduling
  – Memory management
  – Paging
  – Page replacement
  – Disk I/O
  – File systems
  – Advanced topics (won’t appear in the final exam)
• The End
Final Exam

• April 24\textsuperscript{th}, 9:30 AM
  – NR-25

• Closed book
  – Last year’s final exam is posted on the class website
Final Mechanics

• Bulk of the final covers material after midterm
  – Scheduling, memory management (paging and replacement), file systems

• Some material on concurrency, synchronization
  – Synch primitives, synch problems

• Based upon lecture material and project

• Again, please, do not cheat
Scheduling

• When does scheduling happen?
  – Job changes state (e.g., waiting to running)
  – Interrupt, exception
  – Job creation, termination
Scheduling Goals

• Goals
  – Maximize CPU utilization
  – Maximize job throughput
  – Minimize turnaround time
  – Minimize waiting time
  – Minimize response time

• Different systems have different goals

• What is the goal of a batch system?
• What is the goal of an interactive system?
Starvation

• Starvation
  – Indefinite denial of a resource (CPU, lock)

• Causes
  – Side effect of scheduling
  – Side effect of synchronization

• Operating systems try to prevent starvation
Scheduling Algorithms

• What are the properties, advantages and disadvantages of the following scheduling algorithms?
  – First Come First Serve (FCFS)/First In First Out (FIFO)
  – Shortest Job First (SJF)
  – Priority
  – Round Robin
  – Multilevel feedback queues

• What scheduling algorithm does Unix use? Why?
Memory Management

• Why is memory management useful?
  – Why do we have virtual memory if it is so complex?
• What are the mechanisms for implementing MM?
  – Physical and virtual addressing
  – Partitioning, paging, and segmentation
  – Page tables, TLB
• What are the policies related to MM?
  – Page replacement
• What are the overheads related to providing memory management?
Virtualizing Memory

• What is the difference between a physical and virtual address?
• What is the difference between fixed and variable partitioning?
  – How do base and limit registers work?
• What is internal fragmentation?
• What is external fragmentation?
• What is a protection fault?
Paging

• How is paging different from partitioning?
• What are the advantages/disadvantages of paging?
• What are page tables?
• What are page table entries (PTE)?
• Know these terms
  – Virtual page number (VPN), page frame number (PFN), offset
• Know how to break down virtual addresses into page numbers, offset
• How have you implemented paging in OS161?
Page Table Entries

• What is a page table entry?
• What are all of the PTE bits used for?
  – Modify
  – Reference
  – Valid
  – Protection
Segmentation

• What is segmentation?
• How does it compare/contrast with paging?
• What are its advantages/disadvantages with respect to partitioning, paging?
• What is a segment table?
• How can paging and segmentation be combined?
Page Tables

• Page tables introduce overhead
  – Space for storing them
  – Time to use them for translation

• What techniques can be used to reduce their overhead?

• How do two-level (multi-level) page tables work?
TLBs

• What problem does the TLB solve?
• How do TLBs work?
• Why are TLBs effective?
• How are TLBs managed?
  – What happens on a TLB miss fault?
• What is the difference between a hardware and software managed TLB?
Page Faults

• What is a page fault?
• How is it used to implement demand paged virtual memory?

• What is the complete sequence of steps, from a TLB miss to paging in from disk, for translating a virtual address to a physical address?
Advanced Mem Management

• What is shared memory?
• What is copy on write?
• What are memory mapped files?
Page Replacement

• What is the purpose of the page replacement algorithm?

• What application behavior does page replacement try to exploit?

• When is the page replacement algorithm used?

• Understand
  – Belady’s (optimal), FIFO, LRU, Approximate LRU, LRU Clock, Working Set

• What is thrashing?
Disk

• Understand the memory hierarchy concept, locality

• Physical disk structure
  – Platters, surfaces, tracks, sectors, cylinders, arms, heads

• Disk interface
  – How does the OS make requests to the disk?

• Disk performance
  – What steps determine disk request performance?
  – What are seek, rotation, transfer?
File Systems

• Topics
  – Files
  – Directories
  – Sharing
  – Protection
  – Layouts
  – Buffer Cache

• What is a file system?

• Why are file systems useful (why do we have them)?
Files and Directories

• What is a file?
  – What operations are supported?
  – What characteristics do they have?
  – What are file access methods?

• What is a directory?
  – What are they used for?
  – How are the implemented?
  – What is a directory entry?

• How are directories used to do path name translation?
File System Layouts

• What are file system layouts used for?
• What are the general strategies?
  – Contiguous, linked, indexed?
• What are the tradeoffs for those strategies?
• How do those strategies reflect file access methods?
• What is an inode?
  – How are inodes different from directories?
  – How are inodes and directories used to do path resolution, find files?
File Buffer Cache

• What is the file buffer cache, and why do operating systems use one?
• What is the difference between caching reads and caching writes?
• What are the tradeoffs of using memory for a file buffer cache vs. VM?
Final words on the lab

- Likely the hardest lab you do in your undergraduate years
  - But if you survived it, your programming & hacking capabilities have significant improvements
  - OS: one of the hardest program to write & debug
  - Debug concurrent programs, user- AND kernel space, low-level hardware, interrupts and exceptions, assembly, etc.
  - Hack into a large, unfamiliar code base and implement additional features
  - Work as a team
  - Using version control systems
  - etc.

- I am very very proud of you!
Summary

• Now you understand how a computer works internally
  – More importantly, you had your hands dirty and implemented one
  – If you found such ‘hand-dirty’ experience interesting:
    • Take my ECE 454 course in Fall
    • Consider doing a Master with me
• Any remaining questions?
The End

• Congratulations on surviving ECE 344!
  – It’s a very challenging course, but I hope you found it worthwhile

• Good luck, and thanks for a great class!