

# **CSE 120**

# **Principles of Operating Systems**

**Fall 2024**

**Final Review**

Geoffrey M. Voelker

# Review Q&A Session

---

- Monday (Dec 9) at 3pm
  - ◆ To bypass room scheduling hassles, we'll do it over zoom
  - ◆ Same format at the midterm review
  - ◆ Bring your questions and I will answer them
- Replaces my Monday office hours

# Related Courses

---

- If you enjoy CSE 120 topics, you might find some other courses interesting going forward
  - ◆ CSE 122: Wireless Networks (WI25)
  - ◆ CSE 123: Networking (2025-2026)
  - ◆ CSE 124: Networked Services (SP25)
  - ◆ CSE 125: Software System Design & Impl. (SP25)
  - ◆ CSE 127: Computer Security (WI25, SP25)
  - ◆ CSE 160: Parallel Computing (WI25)
  - ◆ CSE 190: Wireless Embedded Systems (WI25)
  - ◆ CSE 190: Data Center Systems (WI25)
  - ◆ CSE 190: ML Systems (SP25)

# Overview

---

- Final mechanics
- Memory management
- Paging
- Page replacement
- File systems
- Protection
- The End

# Final Mechanics

---

- Tue Dec 10 8–11am in → [The Jeannie](#) ←
- Most of the final covers material after the midterm
  - ♦ Lectures Oct 31 – Nov 21: VM, file systems, protection
  - ♦ Projects 2 & 3, Homeworks 3 & 4
- Some material on concurrency, synchronization
  - ♦ Synch primitives, synch problems
- Based upon lecture material, homeworks, and project
- **Closed book, one double-sided 8.5"x11" page of notes**
  - ♦ Can be typed or hand-written, must be single layer
  - ♦ No devices (calculators) or any other materials
    - » We do expect everyone to be familiar with binary, hex notation
- **Obligatory: Please, do not cheat**

# Suggested Study Strategy

---

- Quickly review slides for lectures 10/1–10/24, hw #1-2
  - ◆ Good for context
  - ◆ Will be surprised how much sense it makes now
- Review **synchronization primitives and problems**
- Focus on lectures 10/31–11/21, hw #3-4, projects #2-3
  - ◆ Remaining lectures not on the final

# 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

---

- What are the advantages/disadvantages of paging?
- What are page tables?
- What are page table entries (PTE)?
- Know these terms
  - ◆ Virtual page number (VPN), physical page number (PPN)/page frame number (PFN), offset
- Know how to break down virtual addresses into page numbers, offset
- How have you implemented paging in Nachos?

# Page Table Entries

---

- What is a page table entry? (In Nachos?)
- What are all of the PTE bits used for?
  - ◆ Modify (dirty)
  - ◆ Reference (used)
  - ◆ Valid
  - ◆ Protection

# Segmentation

---

- What is segmentation?
- How does it compare/contrast with paging?
- What are its advantages/disadvantages with respect to paging?
- What is a segment table?

# 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?
  - ♦ What is done in hardware, what is done in software?

# Advanced VM Topics

---

- 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
- What is thrashing?



# Disk

---

- Can skip physical disk structure, characteristics
  - ◆ High-level point is that accessing disks is very slow relative to accessing memory

# File Systems

---

- Topics
  - ◆ Files
  - ◆ Directories
  - ◆ Sharing
  - ◆ Implementation
  - ◆ 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 is a directory?
  - ◆ What are they used for?
  - ◆ What is a directory entry?

# 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?
- How can we use the file buffer cache for read ahead?

# File System Implementation

---

- How do we manage information on disk?
  - ◆ What are advantages of using disk blocks?
  - ◆ What kind of fragmentation does it have?
- What are bitmap blocks used for?
- What is the superblock?

# 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?
- What is an inode?
  - ♦ How are inodes different from directories?
- How are inodes and directories used to do path resolution, find files?

# File Operations

---

- How do soft links work?
- How do hard links work?
- How does create work?
- How does delete work?
- How does rename work?
- How do we stitch together multiple file systems into a single, global name hierarchy?

# Protection

---

- What are the principles of protection?
- How is user identity used in protection?
- What is file protection used for?
  - ◆ What are access control lists (ACLs)?
  - ◆ How are they represented, implemented?
- How does protection work with running processes?
  - ◆ How do we check whether a process is allowed to use files, memory?
  - ◆ What are capabilities?
  - ◆ How do we derive capabilities from ACLs?
    - » Why open, then read/write?
  - ◆ What are advantages/disadvantages of ACLs & capabilities?



# Final Thoughts

---

- Any remaining questions?
- Just a few final things...