CMSC216: Finale

Chris Kauffman

Last Updated:
Thu Dec 11 09:14:01 AM EST 2025

Logistics

Goals

- ▶ Threads Wrap
- ▶ Final Exam Logistics
- ▶ Review / Practice Exam
- ▶ Parting Words

Date	Event
Tue 09-Dec	Threads
Wed 10-Dec	Dis: Bonus Review
Thu 11-Dec	Practice Exam
Fri 12-Dec	P5 Due
Sat 13-Dec	CourseExp Due
Mon 15-Dec	Exit Survey Due
Tue 16-Dec 6:30-8:30pm	Final Exam
	Lec 1xx: IRB 0324
	Lec 2xx: ESJ 0202

Office Hours

- ▶ Kauffman OH Thu 3-4pm
- ▶ Staff OHs end Friday
- ▶ Kauffman may have special OHs on Monday, will post about it

Announcements: Student Feedback Opportunities

Course Experiences Now Open

e.g. Rate your Professor

- ▶ https://www.courseexp.umd.edu/
- ▶ **If response rate reaches 80% for every section...**
- ▶ **by Sat 13-Dec-2025 11:59pm...**
- ▶ **Prof K will reveal a Final Exam Question**
- ▶ No answers but public discussion welcome
  (Final Exam is on Tue 16-Dec-2025)

Canvas Exit Survey

- ▶ Now open on ELMS/Canvas
- ▶ https://umd.instructure.com/courses/1388320/
  quizzes/1808628
- ▶ Worth 1 Full Engagement Point for completion
- ▶ Due prior to Final Exam (Mon 15-Dec 11:59pm)

Final Exam Logistics

- ▶ Final Exam in person 6:30-8:30pm Tue 16-Dec
- ▶ Location Based on Section
  Lec 1xx (09:30am): IRB 0324
  Lec 2xx (11:00am): ESJ 0202
- ▶ 2 hours long, Open Resource
- ▶ 6 sides of paper (3 pages front/back)
- ▶ ~4 sides like 3rd Midterm Exam
  System Calls, Processes, Low-level I/O, Memory System
  Hardware and Cache Efficiency, Virtual Memory, Threads,
  P4/P5 Material
- ▶ ~2 sides Comprehensive Review
  Tie together concepts that pervaded the semester like memory
  layout, physical parts of the computing system,
- ▶ Question format like previous exams: coding, debugging,
  conceptual, practical applications

What have we done?

C Programming

Lowest of the "high-level" languages, gives fairly direct control over capabilities of the machine at the expense of coding difficulty and ease of mistakes

Assembly Programming

Tied directly to what a processor can do, studied x86-64 specifically, exposes processor internals like registers, instructions, operand sizes, etc.

Computing Hardware

Basics components like CPU, Registers, Cache Memory, DRAM, Disks, how they interact

Operating System Basics

Programs exist in an environment usually managed by an OS, provides abstractions like Processes, Files, Threads, along with the ability to manipulate and coordinate these through System Calls

We studied Computing Systems, as you might expect

Further Coursework / Activities

- **CMSC411 Computer Systems Architecture**: Develops hardware/software interface in more detail, study pipelines + superscalar features in more detail, cache architecture / implementation, examine multi-core systems

- **CMSC412 Operating Systems**: Study internal design issues associated with operating systems, handling hardware, tradeoffs on different approaches to management, theoretical algorithms around resource coordination.

- **CMSC414 Computer and Network Security**: Builds on the buffer overflow attacks we studied, discusses other attacks via networks, counter measures, design aspects of secure systems

- **CMSC416 Introduction to Parallel Computing**: Hardware and Programming Models for parallelism. Builds substantially on using multiple threads/processes to cooperatively compute faster. *Kauffman will teach the Spring 2026 section.*

Off-Season Practice

Students often ask what they could do during a break to sharpen their computing skills. Here are a few ideas.

- ▶ **Read**: The Art of Unix Programming by Eric S. Raymond Fantastic philosophical and pragmatic discussion of how to build systems that work especially in the Unix environment. (free online)
- ▶ **Complete**: If you didn't finish a project in this course or another, take some time to do so.
- ▶ **Extend**: If you use VS Code, Write an Extension for it that does something interesting. This will teach you MUCH about modern software development
- ▶ **Build**: Buy an Arduino Microcontroller ($10) and get a "Blinky" routine to run; it's C code! Adafruit has tons of fun toys with accompanying tutorials.
- ▶ **Rest**: Take some time away from the screen for fun. Recharging is as important for people as for phones. Play outside. See some people in person. Breathe.

Practice Final

- ▶ Take a few minutes to look this over on your own then together
- ▶ Kauffman will answer a few questions on it and post solutions later today

Nothing Ever Ends
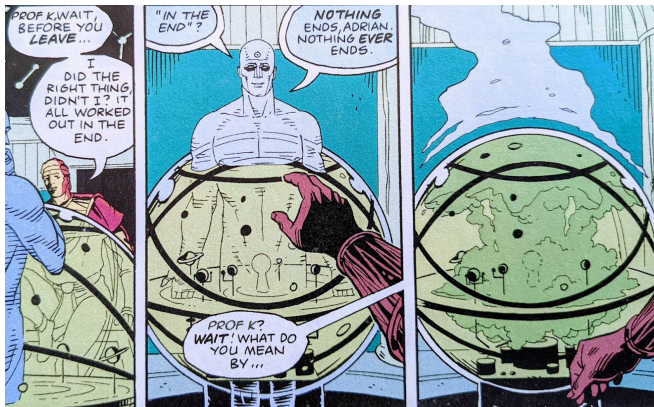
- ▶ What you studied will recur in your career at some point and demonstrate whether you learned it well the first time or need another pass.
- ▶ Some of it will change in the future and make you feel old.
- ▶ Expect this and stay determined.

Conclusion

It's been a hell of a semester.
Thank you for your time,
attention, and effort.
I'm proud of all of you.
Keep up the good work.
**Keep caring, about your work,
your peers, and yourself.**
Stay safe. Happy Hacking.