

CMSC216: Finale

Chris Kauffman

*Last Updated:
Tue May 13 09:20:48 AM EDT 2025*

Logistics

Goals

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

Date	Event
Mon 12-May	Dis: Review Lab12 / HW12 Due
Tue 13-May	Practice Exam P5 Due
Wed 14-May	Course Evals Due
Fri 16-May 6:30-8:30pm	Final Exam Lec 1xx: IRB 0324 Lec 2xx: ESJ 0202

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 Tue 13-May-2025 11:59pm...**
- ▶ **I will reveal a Final Exam Question**
- ▶ No answers but public discussion welcome
- ▶ Feedback open through Wed 14-May

Canvas Exit Survey

- ▶ Now open on ELMS/Canvas
- ▶ <https://umd.instructure.com/courses/1377846/quizzes/1755599/>
- ▶ Worth 1 Full Engagement Point for completion
- ▶ Due prior to Final Exam (Thu 15-May 11:59pm)

Final Exam Logistics

- ▶ Final Exam in person 6:30-8:30pm Fri 16-May
- ▶ **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, 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

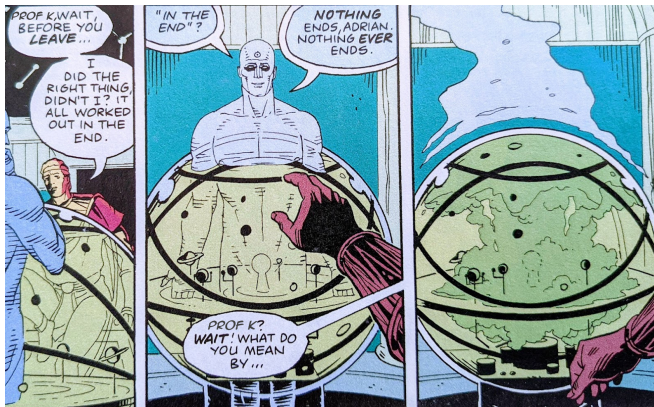
Students often ask what they could do during a break to sharpen their computing skills. Here are a few ideas.

- ▶ **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 learned 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, effort, and care.
I'm proud of all of you.
Keep up the good work.
Stay safe. Happy Hacking.



CMSC412 or bust

valgrind: got my
back on memory!

Submission
accepted

> make

nothing to be done

gdb: yeah

you know me!

Stack or Heap?
I know which!

fork()'d

and

wait()'d

Why yes, I do know assembly.

C code: I know this!

My code is full of
shift, bit shift!

```
movq $0, %eax
```

```
retq #success!
```

10/10 Tests Passed

I struggled and
built skills

Debugging
complete!

You're **STILL** filled with
DETERMINATION!

Whew,
time for
a snack



chaotichero