CSCI 2021: Finale

Chris Kauffman

Last Updated:
Mon May 2 03:35:15 PM CDT 2022

Logistics

Goals

- Final Exam Logistics
- Evaluations
- Review

P5: Questions?

1 Required Problem: EL Malloc

Date	Event
Mon 5/2	Last Lecture, Review SRTs due by 1:25pm P5 Due
Tue 5/3	Lab 14 / HW 14 Due
Fri 5/6	Canvas Feedback Survey Due
Sat 5/7	Final Lec 001 8:00-10:00am
Mon 5/9	Final Lec 010 10:30-12:30pm

Final Exam Logistics

- Final Exam in person, normal lecture location
  - ~1.5 pages F/B Virtual Memory / Linking / Object Files / P5
  - ~1 page F/B Comprehensive Review
    (F/B = Front/Back)
- 2 hours to take Final Exam in person

Course Feedback

Official Student Rating of Teaching (SRTs)

- Official UMN Evals are done online this semester
- Available here: https://srt.umn.edu/blue
- **EVALUATE YOUR LECTURE SECTION: 001 or 001**
  Optionally evaluate lab section
- **Due** Mon 5/02/2022 by 1:25pm
- Response Rate $\geq$ 80% in **both sections** $\rightarrow$ One Final Exam Question Revealed

Course Specific Survey

- Will open a course-specific survey next week
- Open on Canvas Fri 4/29 to Fri 5/06
- Worth 1 Engagement Point to Complete it

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 Architecture

Basics of how CPUs + Memory are built, transistors/gates to do "work" and performance ramifications on code

Processing Systems/Environment

Programs exist in an environment including file formats for executables, specifics of loading, virtual memory system to catch errors/link libraries

Did I miss anything?

Further Coursework

- **CSCI 4061 Intro to Operating Systems**: Direct successor, required for CS majors, builds on 2021 content to develop the shape of an operating system.
- **CSCI 4203 Computer Architecture**: Develops hardware/software interface in more detail, study pipelines + superscalar features in more detail, examine multi-core systems
- **CSCI 5103 Operating Systems**: Study internal design issues associated with operating systems, handling hardware, tradeoffs on different approaches to management, theoretical algorithms around resource coordination.
- **CSCI 4271W Development of Secure Software Systems**: Focus on security issues, methods to circumvent OS/hardware protections and how ensure safety in programs, incorporating security features into system design.
- **CSCI 5143 Real-Time and Embedded Systems**: Small systems often lack an OS and fancy hardware, more direct interactions with hardware, must manage resources in your own programs, teaches much about what the machine does as usually less is provided in embedded systems.

Survey Says ...

SRTs Response Rate

Lec	Responded	Invited	%Response 1:25pm	%Response 2:30pm
001	143	189	75%	80%
010	129	161	80%	81%

- Thanks to all that have responded; SRTs stay open until 11:59pm last day of classes
- Didn't make the 80% threshold for both sections by 1:25pm BUT...
- Lec 001 rallied to complete by 2:30pm so will reveal a final exam problem during 3:35pm

Final Exam Question

See PDF during 3:35pm Video

Practice Final

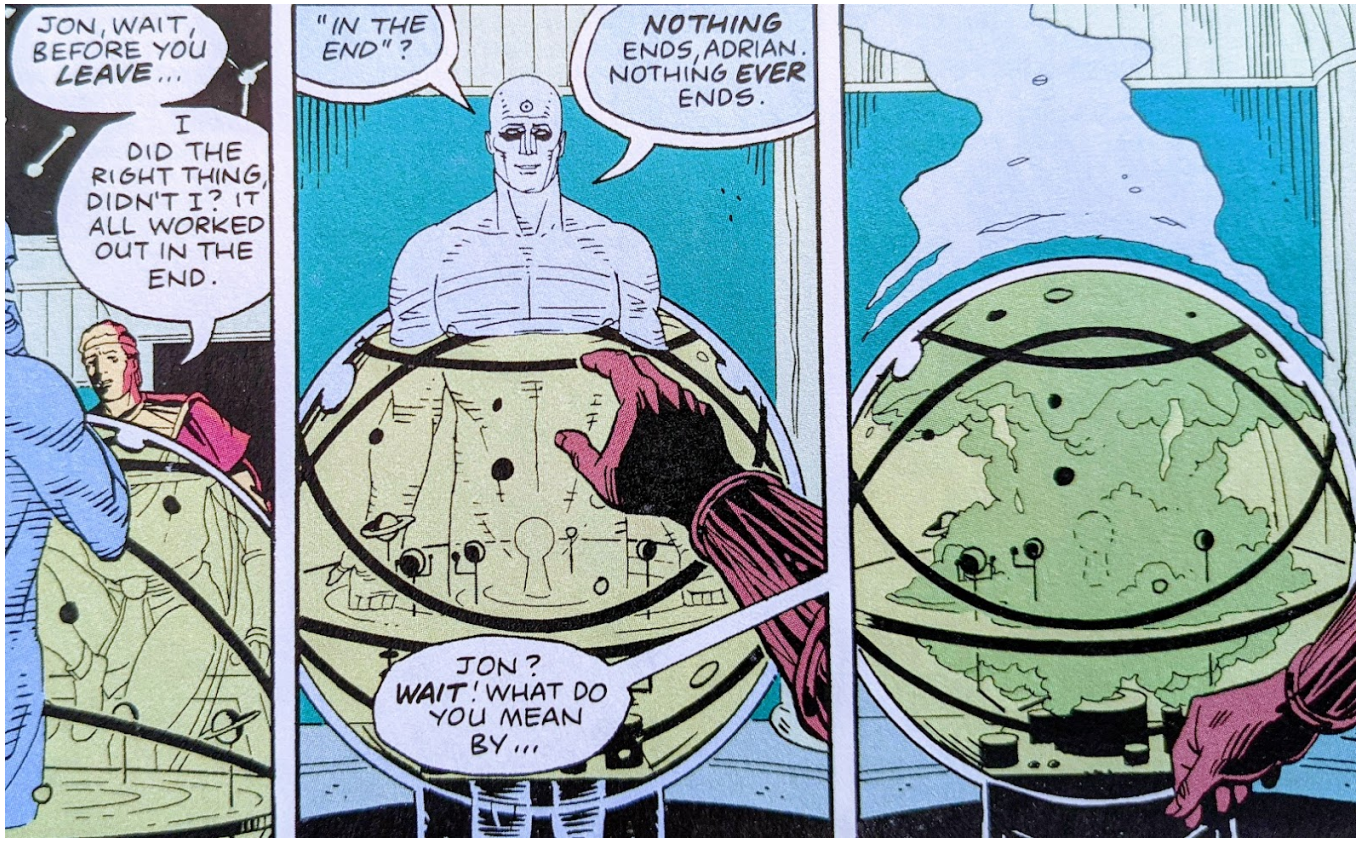
- Take a few minutes to look this over on your own then together
- Kauffman will answer a few questions on it and post solutions later today

Summer Practice

Students often ask what they could do during a break to keep up 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 a Raspberry Pi ($40) and set it up; buy an Arduino ($10) and get a "Blinky" routine to run
- 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.

Nothing Ever Ends



By now you should realize that what you learned

- Will come up again showing whether you learned it well the first time or need another pass.
- Will change in the future and make you feel old.

Expect this and stay stay patient.

Conclusion

It's been a hell of a semester.
I'm proud of all of you.
Keep up the good work.
Stay safe. Happy Hacking.