EECS 482
Introduction to Operating Systems
Fall 2020

Peter Chen

University of Michigan

https://grader2.eecs.umich.edu/eecs482

Professors

Peter Chen

Harsha Madhyastha

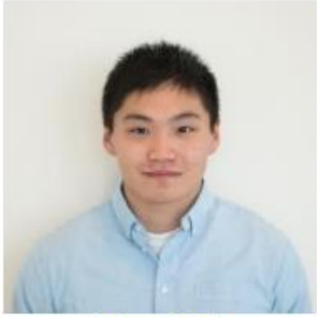
Brian Noble

3-4:30 lecture

1:30-3 lecture

Post-lecture Q&A

GSIs and IAs


Tony Bai

Justin Beemer

Morgan Borjigin-Wang

Nathan Brown

Louis Gouirand

Ryan Johnson

Brandon Kayes

Austin Kiekintveld

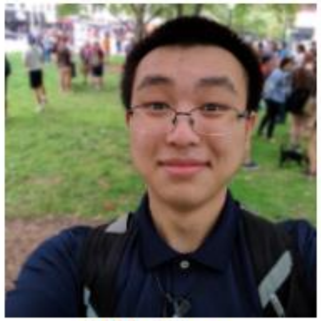
Audrey Ladd

Joseph Nwabueze

Vaughn Taylor

Victoria Whall

Yuxi Xie

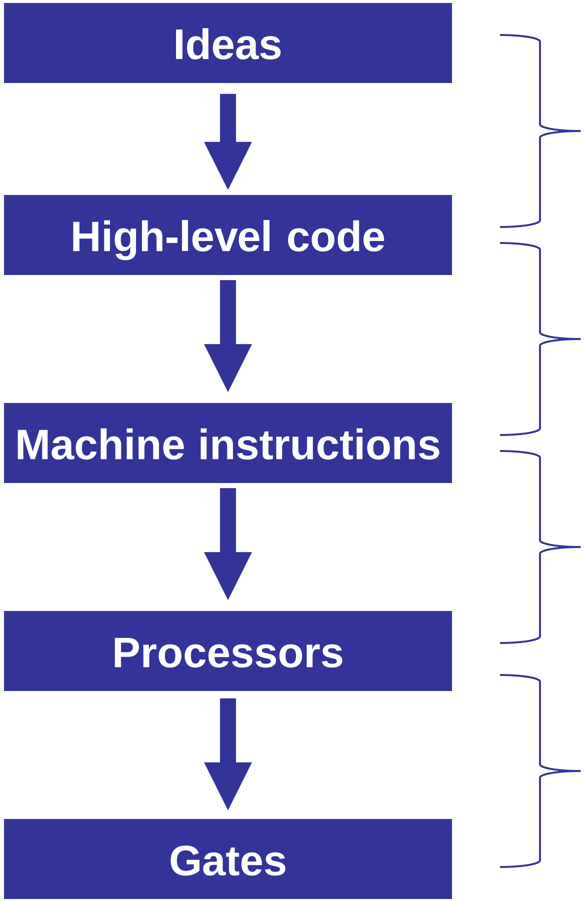

Yilin Yang

Agenda for today

- Why do we need EECS 482?
- Course syllabus and logistics

- Why do we need an OS and what does it do?
- How did OSes evolve to what we have today?

Overview of CS/CE Curriculum



Ideas

High-level code

Machine instructions

Processors

Gates

EECS 280, 281 (programming)

EECS 483 (compilers)

EECS 370 (computer organization)

EECS 270 (digital design)

But some mysteries remain…

How does a computer start when you turn it on?

How to get new program into memory and start executing it?

How to … en?

How to

How to

How to

How to … ff?

How to … ?

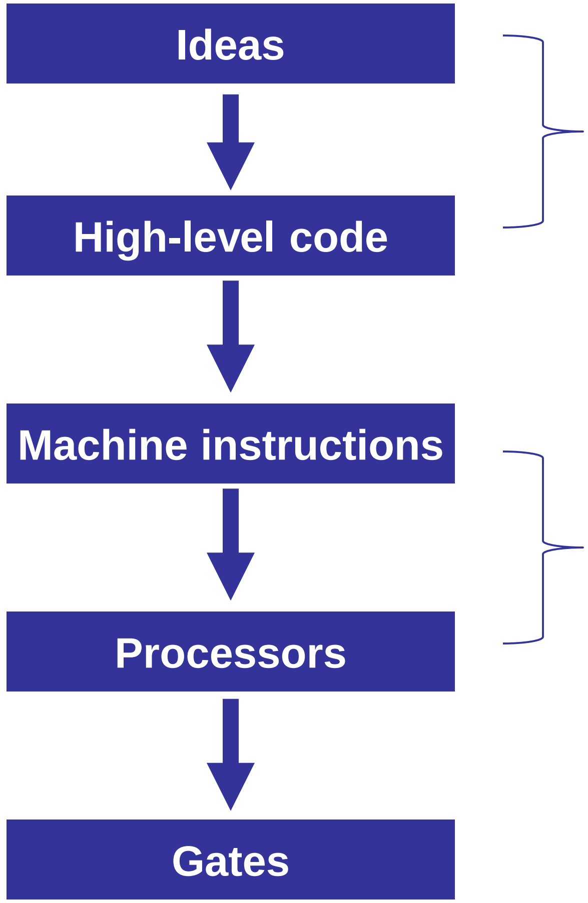
How to receive e-mail when you're not on the computer?

The OS does all of this.

After this semester, you should be able to answer all of these questions!

Syllabus

Prerequisites



Ideas

High-level code

Machine instructions

Processors

Gates

EECS 281 (programming and data structures)

EECS 370 (computer organization)

Lectures

- Two lecture sections (loosely synchronized)
- Lectures will be interactive
- Attend lecture and participate
  - Via chat
  - Via audio/video: “raise hand” in participant window, then turn on camera and mic when called on
- Recordings and slides posted after lecture
  - Ask/upvote questions on Piazza live Q&A
- If I lose connectivity during lecture
  - Stay on call and wait 10 minutes
  - If I’m not back, I’ll record and post rest of lecture

Labs

7 lab sections

Pre-lab questions will be posted on course web page a week in advance

- Do them before going to your section

First lab section is this Friday

Course material

Course web page:
https://grader2.eecs.umich.edu/eecs482

Piazza

Textbook (recommended)

- Operating Systems: Principles and Practice, by Tom Anderson and Mike Dahlin

Projects

Project 1: Concurrent programming

Project 2: Thread manager

Project 3: Virtual memory manager

Project 4: Multi-threaded, secure network file system

By the end of the semester, you’ll have built the main parts of an operating system

Projects

Challenging

Little code

- My total code = 1137 lines

Lots of thinking

- Many new concepts
- Concurrent threads running on multiple processors
- Multiple address spaces spread across memory and disk
- Persistent data structures
- Distributed computing

4-credits vs. 6-credits

What’s the same?

- Lectures
- Lab sections
- Project 1
- Exams (probably)

What’s different?

- 6-credit students implement advanced features in Projects 2, 3, 4
- Class time for EECS 498-002 structured as extra office hours (12-24 hours/week)

4-credits vs. 6-credits

	Core project	Advanced features
Project 1		
Project 2	1 CPU	> 1 CPU
Project 3	Process starts with empty arena	Process starts with copy of parent's arena
Project 4	Reader/writer locks Statically allocated locks	Upgradable reader/writer locks Dynamically allocated locks

4-credits vs. 6-credits

	Core project	Advanced features
Project 1	67 LOC	
Project 2	1 CPU 210 LOC	> 1 CPU
Project 3	Process starts with empty arena 242 LOC	Process starts with copy of parent’s arena
Project 4	Reader/writer locks Statically allocated locks 523 LOC	Upgradable reader/writer locks Dynamically allocated locks
Total	1042 LOC	

4-credits vs. 6-credits

	Core project	Advanced features
Project 1	67 LOC	67 (+0) LOC
Project 2	1 CPU 210 LOC	> 1 CPU 249 (+39) LOC
Project 3	Process starts with empty arena 242 LOC	Process starts with copy of parent’s arena 268 (+26) LOC
Project 4	Reader/writer locks Statically allocated locks 523 LOC	Upgradable reader/writer locks Dynamically allocated locks 553 (+30) LOC
Total	1042 LOC	1137 (+95) LOC

Project tips

Start early!

Develop project incrementally

- Build a little functionality, then test, test, test, …

Expect to spend most of your time debugging

Project tips

Make good use of staff – we are here to help

- Visit office hours (45 hours/week + extra hours when projects are due)
- Participate in discussion on Piazza
- Look for help in lecture, lab, and textbook

Our goal is to help you solve your problems (not to solve them for you)

Project groups

Projects 2-4 done in groups of 2 or 3

- Declare your group (by Sep. 17) – redeclare if you declared before August 31
- Post to Piazza if you don't know anyone

Choose group members carefully

- Check schedule, class goals, working style, etc.
- Mix of 4-credit and 6-credit students strongly discouraged

We'll evaluate every member's contributions

- Peer feedback
- git statistics

Project policies

Development

- Github: EECS 482 will provide private repositories
- Commits should match contributions

Submission

- 1 feedback/day + 3 bonus feedbacks/project
- Due by midnight ET (hard deadline!)
- 3 late days across all projects

Collaboration

- Okay to clarify project handout or discuss C++ syntax
- Not okay to discuss solutions
- Not okay to benefit from past solutions

Exams

- Midterm: Oct. 20, 7-9 pm (ET)
- Final: Dec. 10, 8-10 am (ET)
- Alternate time for those in incompatible time zones

Schedule of topics

- Operating system manages and abstracts hardware resources

- Before midterm: CPU, memory

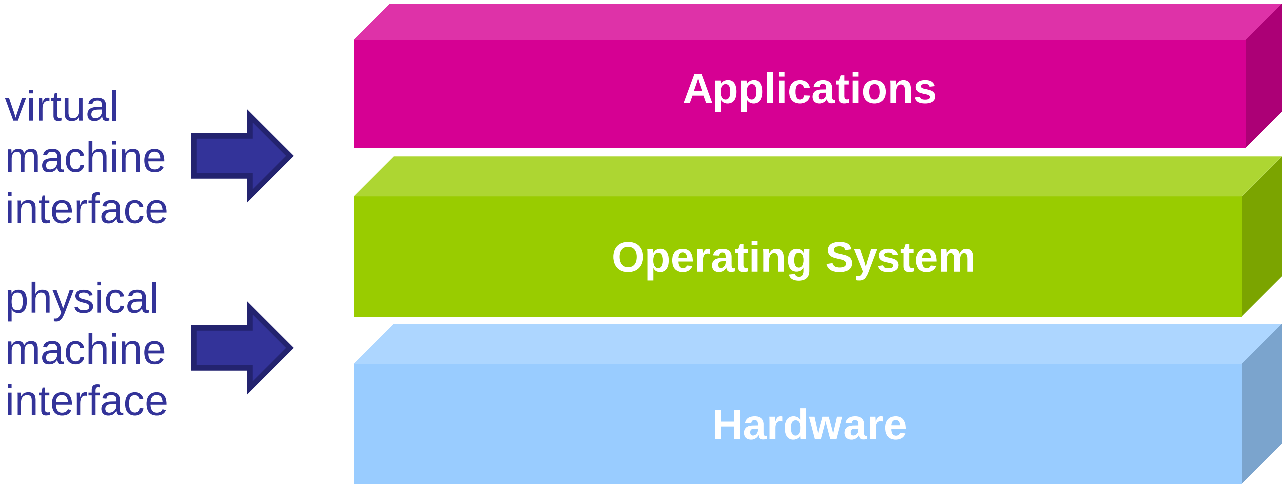
- After midterm: storage, distributed systems, case studies

Administrative break

- Wait list
  - All students on wait list should be able to enroll
- Register your github ID on course web page
- Look for a project group
- Go to lab this Friday

What is an OS?

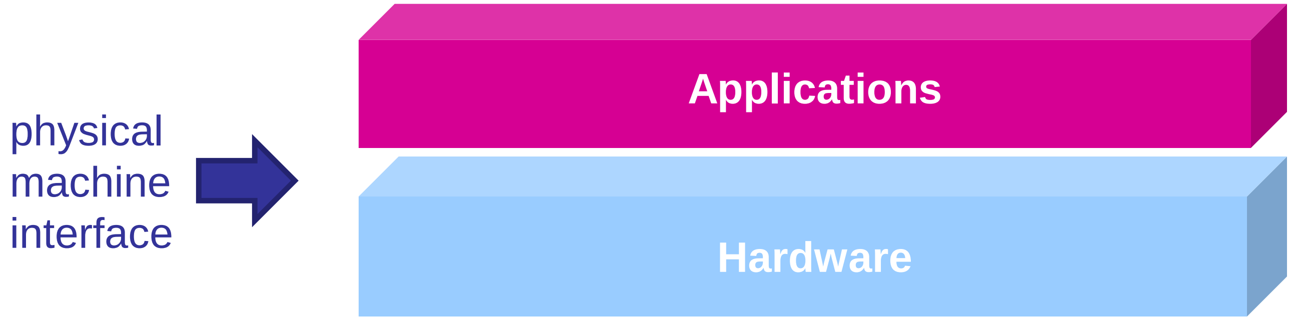
- The operating system is the software layer between application programs and the hardware



- What if there were no OS?

What is an OS?

- The operating system is the software layer between application programs and the hardware



- What if there were no OS?

-