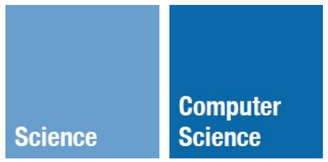
Preliminaries



CS 450: Operating Systems
Michael Lee <lee@iit.edu>

Instructor

- Michael Lee
  - Email: lee@iit.edu
  - Office: SB 226C
  - Hours: Wed & Fri 1-3PM
    - By appointment only! (Zoom or In-person)

Teaching Assistants

- Lan Nguyen <lnguyen18@hawk.iit.edu>
- Aashesh Kumar <akumar88@hawk.iit.edu>
- Office hours will be posted on course website

Agenda

- Prerequisites
- Resources: website, textbooks, etc.
- Evaluation: assignments, exams, grading
- Class overview

§ Prerequisites

CS Essentials

- Essential algorithms & runtime analysis
- Data structures
- Data representation (bin/hex) and manipulation (shifting/masking/etc.)

Programming Knowledge

- Languages: Assembly (x86 or other), C (or other procedural)
- Compilation process (assembly, compilation, linking, etc.)
- Runtime stack usage and conventions
- Dynamic memory allocation

Computer Organization

- Von Neumann model
- Instruction Set Architectures (RISC/CISC)
- Cache organization and operation
- Interrupt procedures

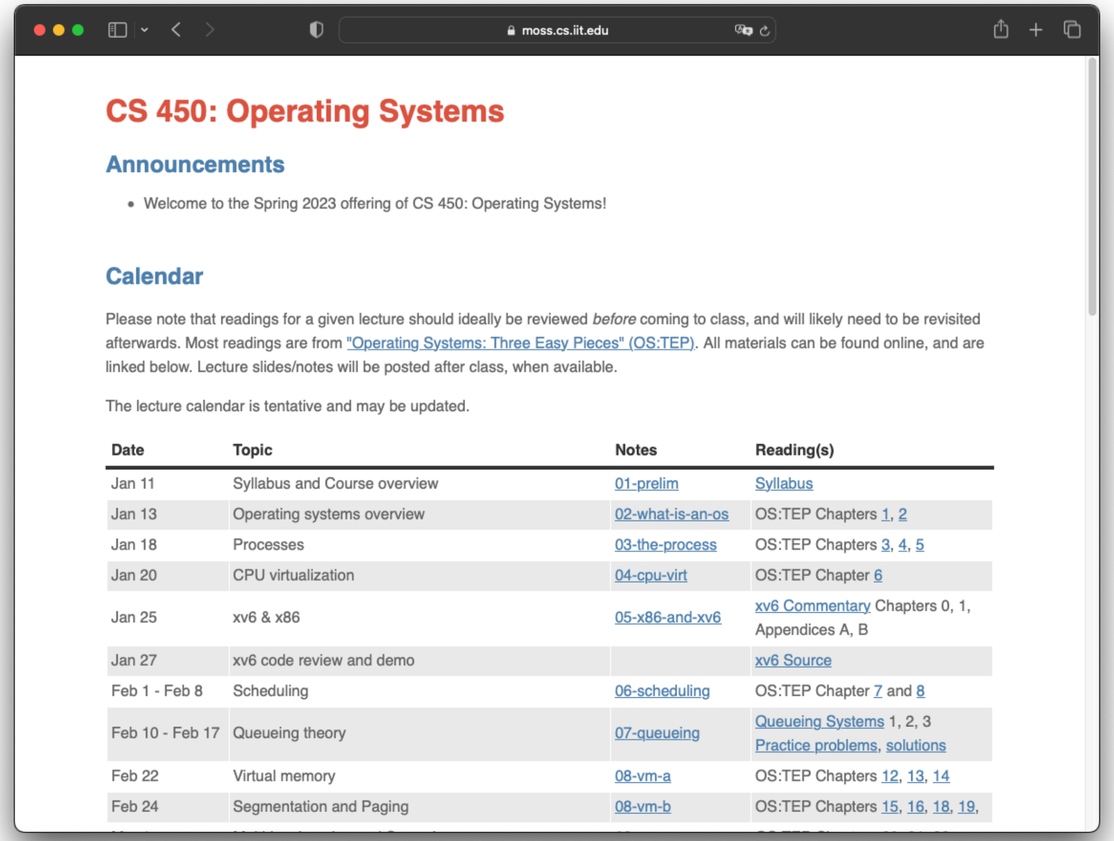
Operating System API

- Ideally: knowledge of Unix syscalls
  - process management (fork/exec/wait)
  - memory management (sbrk/mmap)
  - I/O (open/close/read/write/seek)

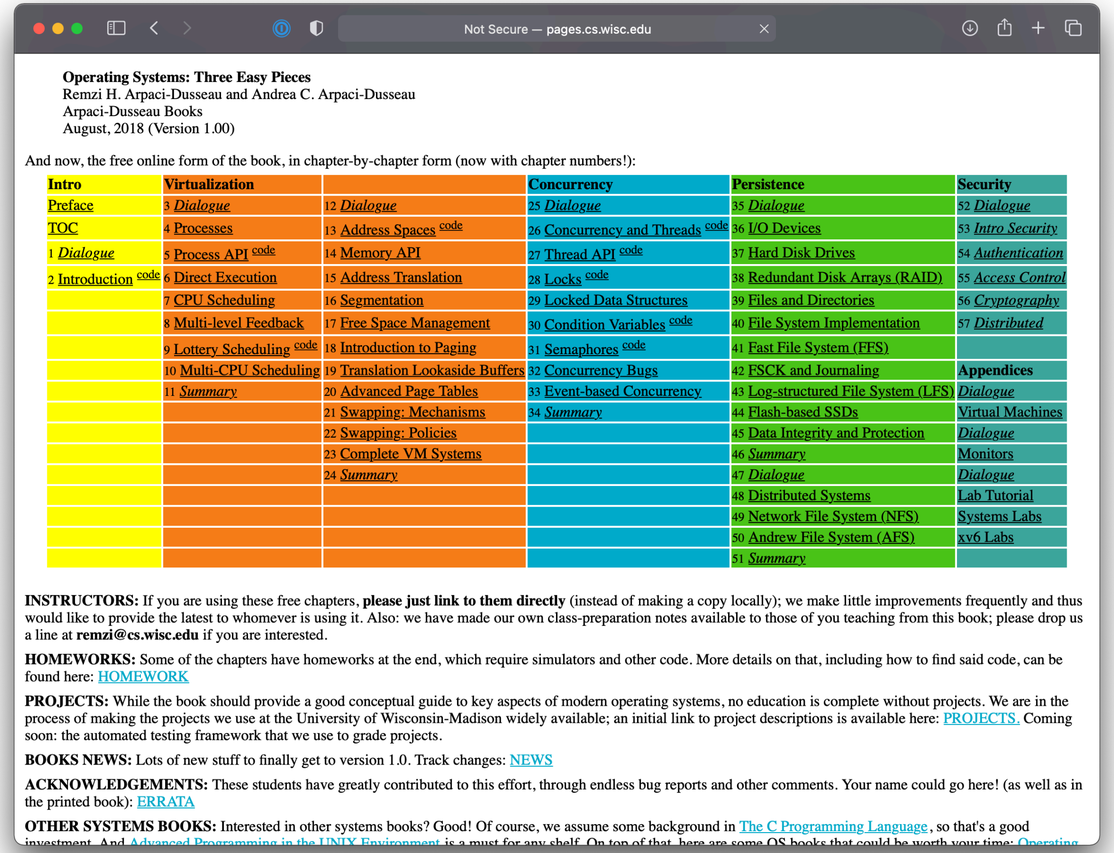
Support Tools

- Command line / Shell (e.g., bash)
- Debugger/Tracer (e.g., GDB)
- Build automation (e.g., Make)
- Version control (e.g., Git)

§ Resources



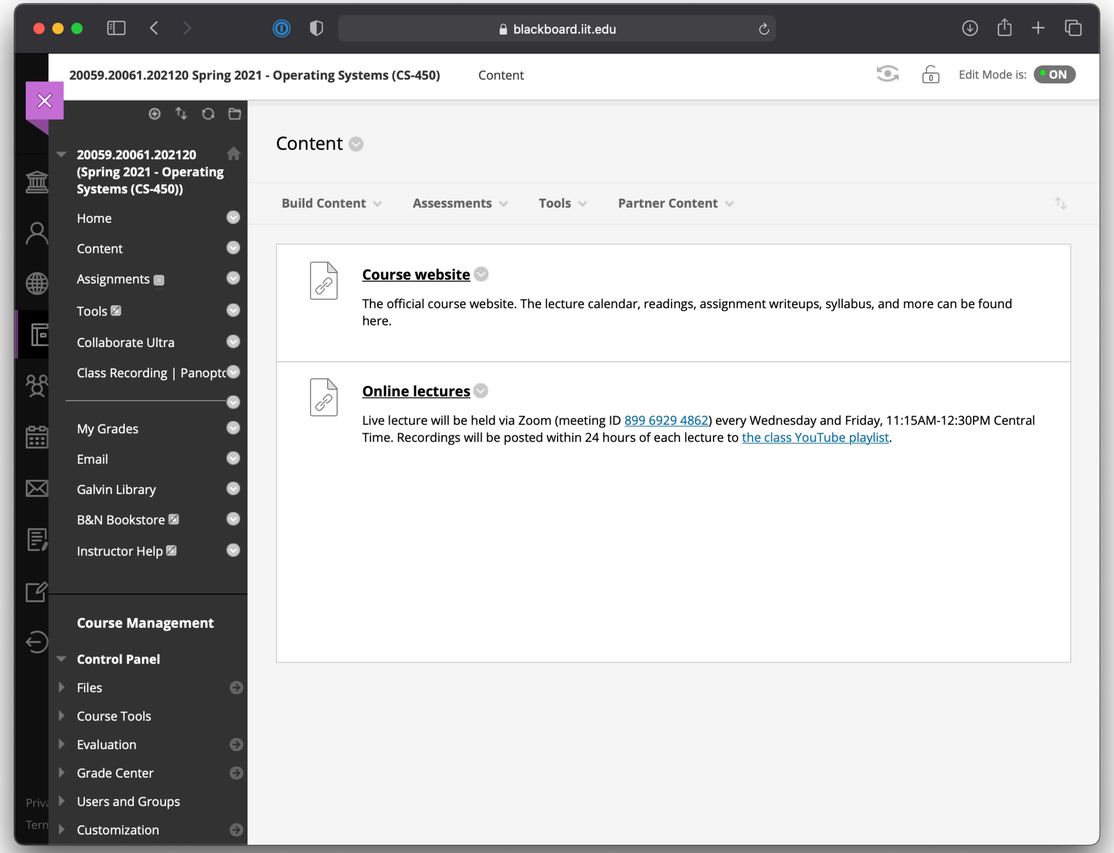
Course website: http://moss.cs.iit.edu/cs450



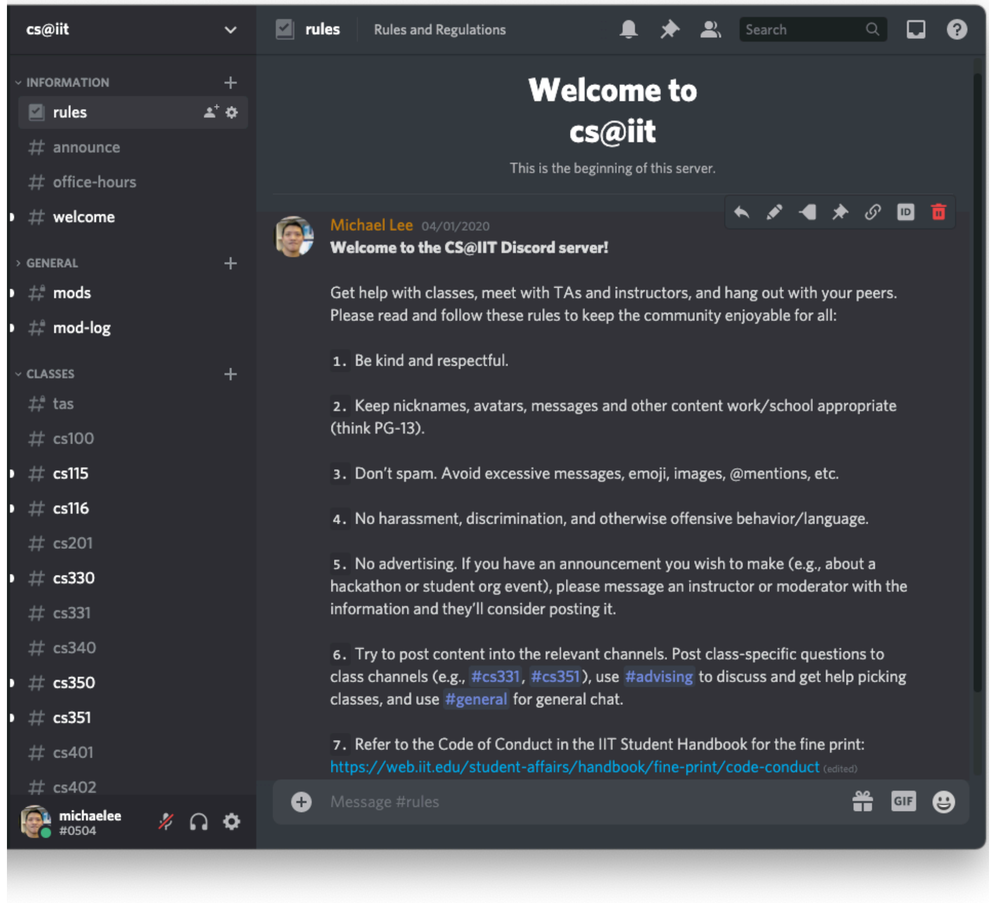
Textbook: “Operating Systems: Three Easy Pieces”
http://pages.cs.wisc.edu/~remzi/OSTEP/

Readings before class!

- Please read (or at least, skim) readings *before* associated lecture
  - OS:TEP is *very* readable, especially for an OS text!
- Check website frequently in case schedule changes



Blackboard: http://blackboard.iit.edu



Discord: Class discussion and Q/A

§ Grading / Evaluation

Assignments

- 5-7 assignments = 60% of grade:
  - Problem sets (quantitative analysis)
  - Machine problems (coding): simulation and kernel hacking

Exams

- Two exams (midterm & final) @ 20% each:
  - scores may be linearly scaled so median/mean is 75%
  - tentative midterm exam date: **March 10**

Grade scale

- A: ≥ 90%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- E: < 60%

§ Class Overview

CS 450

- Capstone of the systems sequence (CS 350 → 351 → 450)
- Wrap up answer to “how do modern (general purpose) computers work under the hood?”
- The OS is the bedrock of almost all modern software!

You should already know what services are *provided* by OSes, along with:

- how to invoke them (syscalls)
- how to use them effectively and efficiently

lingering questions:

- how are processes actually created/tracked?
- how do processes safely & efficiently share resources (e.g., CPU/Mem)?
- how to correctly/safely leverage concurrency?
- how does the file system work?
- how are protection/security enforced?

Primary topics

- Kernel architecture
- Processes and Threads
- Scheduling
- Virtual memory
- I/O architectures and device programming
- File Systems
- Interprocess Communication
- Concurrency and Synchronization

Theory vs. Implementation

- Grand academic debate
- OSes is a huge topic; hard to adequately address both!
  - theory comes first — (hopefully) broad application
  - but it'd be nice to see some working OS code, too …
- Liberal Arts, Architecture majors have "art history/appreciation" classes
  - Why don't we have "code appreciation"?

… the best way to prepare [to be a programmer] is to write programs, and to study great programs that other people have written. In my case, I went to the garbage cans at the Computer Science Center and fished out listings of their operating system.

- Bill Gates

We’ll read/tinker with an existing OS codebase, while making modifications and additions

- great way to understand how OSes work without writing millions of lines of code!
- Our OS, xv6, is based on UNIXv6 — released in 1975, among the first preemptive multitasking OSes, and still a great software engineering blueprint!

Topic for next time: What is an OS?

- Ideas for definition(s):
  - Based on its place in the hardware/software stack
  - Based on its privileges
  - Based on its roles & responsibilities

Before Friday, please read OS:TEP chapters 1 & 2!