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
CS 450: Operating Systems

Announcements
- Welcome to the Spring 2023 offering of CS 450: Operating Systems!

Calendar
Please note that readings for a given lecture should ideally be reviewed before coming to class, and will likely need to be revisited afterwards. Most readings are from "Operating Systems: Three Easy Pieces" (OS:TEC). All materials can be found online, and are linked below. Lecture slides/notes will be posted after class, when available.

The lecture calendar is tentative and may be updated.

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
<th>Notes</th>
<th>Reading(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 11</td>
<td>Syllabus and Course overview</td>
<td>01-syll</td>
<td>Syllabus</td>
</tr>
<tr>
<td>Jan 13</td>
<td>Operating systems overview</td>
<td>02-what</td>
<td>OS:TEC Chapters 1, 2</td>
</tr>
<tr>
<td>Jan 18</td>
<td>Processes</td>
<td>03-the</td>
<td>OS:TEC Chapters 3, 4, 5</td>
</tr>
<tr>
<td>Jan 20</td>
<td>CPU virtualization</td>
<td>04-cpu-</td>
<td>OS:TEC Chapter 6</td>
</tr>
<tr>
<td>Jan 25</td>
<td>x86 &amp; x86</td>
<td>05-x86</td>
<td>x86 Commentary Chapters 6, 1, Appendices A, B</td>
</tr>
<tr>
<td>Jan 27</td>
<td>x86 code review and demo</td>
<td>05-x8</td>
<td>x86 Source</td>
</tr>
<tr>
<td>Feb 1 - Feb 8</td>
<td>Scheduling</td>
<td>06-sched</td>
<td>OS:TEC Chapter 7 and 8</td>
</tr>
<tr>
<td>Feb 10 - Feb 17</td>
<td>Scheduling</td>
<td>07-sched</td>
<td>Queuing Systems 1, 2, 3</td>
</tr>
<tr>
<td>Feb 22</td>
<td>Virtual memory</td>
<td>08-vm</td>
<td>OS:TEC Chapters 12, 13, 14</td>
</tr>
<tr>
<td>Feb 24</td>
<td>Segmentation and Paging</td>
<td>09-vm</td>
<td>OS:TEC Chapters 15, 16, 18, 19</td>
</tr>
</tbody>
</table>

Course website: http://moss.cs.iit.edu/cs450
<table>
<thead>
<tr>
<th>Topic</th>
<th>Chapter 1</th>
<th>Chapter 2</th>
<th>Chapter 3</th>
<th>Chapter 4</th>
<th>Chapter 5</th>
<th>Chapter 6</th>
<th>Chapter 7</th>
<th>Chapter 8</th>
<th>Chapter 9</th>
<th>Chapter 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 Introduction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 Processors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 Memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 File Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Interrupts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 Memory Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Concurrency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 File Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 File Systems</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

INSTRUCTORS: If you are using these free chapters, please just link to them directly (instead of making a copy locally); we make little improvements frequently and thus would like to provide the latest to whomever is using it. Also, we have made our own class preparation notes available to those of you teaching from this book; please drop us a line at remzi@cs.wisc.edu if you are interested.

HOMEWORKS: Some of the chapters have homeworks at the end, which require simulators and other code. More details on that, including how to find said code, can be found here: [HOMESITES](#).

PROJECTS: While the book should provide a good conceptual guide to key aspects of modern operating systems, we believe (or at least hope) that you will be completing projects. We are in the process of making the projects we use at the University of Wisconsin-Madison widely available; an initial link to project descriptions is available here: [PROJECTS](#). Coming soon: the automated grading framework that we use in our grad projects.

BOOKS NEWS: Lots of new stuff to finally get to version 1.0. Track changes: [NEWS](#)

ACKNOWLEDGEMENTS: These students have greatly contributed to this effort, through endless bug reports and other comments. Your name could go here! (as well as in the printed books) [APRADA](#)

OTHER SYSTEMS BOOKS: Interested in other operating systems books? Good! Of course, we assume some background in The C Programming Language, so that’s a good prerequisite, but we think that [Lyons Modern Computer Organization](#) is a more complete book for the architecture of the bits in which all of our books would be written.

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: $\geq 90\%$
- 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!