CS 537:
INTRODUCTION TO OPERATING SYSTEMS

Kai Mast
University of Wisconsin-Madison
Fall 2022

WHO AM I?

Research Area: Cloud Computing, Databases, Blockchains

Name Pronunciation:
Kai *(like the "Ky" in "Kyle")* Mast *(like the American English word "must")*

Position: Post-doctoral Researcher + Associate Lecturer

TODAY'S AGENDA

1. **What will you do in this course?**
   - **How will you be successful?**

2. What is an operating system and why do we need one?

3. Why should you care?

1) ATTEND LECTURES

- Synchronous in-person course @ Tuesday/Thursday
- Attendance and participation is *highly encouraged*!
- Homework available every Friday
  - Simple quizzes related to material covered in the class
  - 5% of your final grade
- Lectures (of both Remzi and me) will be recorded and available online

2) READ THE TEXTBOOK

Free

Operating Systems: Three Easy Pieces

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau (University of Wisconsin-Madison)

NEW: Security Chapters by Peter Reiher (UCLA)

Blog: Why Textbooks Should Be Free

Quick: Free Book Chapters - Hardcover - Softcover (Lulu) - Softcover (Amazon) - Buy PDF - EU (Lulu) - Buy in India - Buy Stuff - Donate - For Teachers - Homework - Projects - News - Acknowledgements - Other Books

Welcome to **Operating Systems: Three Easy Pieces** (now **version 1.00** -- see book news for details), a free online operating systems book! The book is centered around three conceptual pieces that are fundamental to operating systems: **virtualization, concurrency,** and **persistence.** In understanding the conceptual, you will also learn the practical, including how an operating system does things like schedule the CPU, manage memory, and store files persistently. Lots of fun stuff! Or maybe not so fun?

This book **is and will always be free** in PDF form, as seen below. For those of you wishing to **BUY** a copy, please consider the following:

- Lulu Hardcover (v1.00): this may be the best printed form of the book (it really looks pretty good), but it is also the most expensive way to obtain *the black book* of operating systems (a.k.a. *the comet book* or *the asteroid book* according to students). Now just: **$38.00**
- Lulu Softcover (v1.00): this way is pretty great too, if you like to read printed material but want to save a few bucks. Now just: **$22.00**
- Amazon Softcover (v1.00): Same book as softcover above, but printed through Amazon CreateSpace. Now just: **$25.90** (but works with Prime shipping)
- Downloadable PDF (v1.00): this is a nice convenience and adds things like a hyperlinked table of contents, index of terms, lists of hints, tips, systems advice, and a few other things not seen in the free version, all in one massive DRM-free P
latest version. Just: **$10.00**
- Kindle: Really, just the PDF and does not in

https://pages.cs.wisc.edu/~remzi/OSTEP/

3) START PROJECTS PROMPTLY

5-6 programming projects done on CS Linux labs

- All in C (there will be a C tutorial on Sunday!)
- Some use xv6 (toy OS)
  - Gain hands-on experience
  - Build your own OS functionalities
- 50% of your final grade
- One project partner allowed (except for first project); can change partners

Take significant amount of time (about 3 weeks each)

- Specifications are longer than you may be used to
- We provide test cases and grade based on how many you pass
- 5 slip days allowed; at most 2 per project

4) DO NOT CHEAT: ACADEMIC INTEGRITY

It is okay to:

- discuss project or specification in general terms (e.g., “when to return an error?”)
- discuss how different library routines/system calls work
- ask peer mentors, TAs, and professor for help

It is NOT okay to:

- use code samples for similar problems you may find online
- bug someone else for a lot of help
- share your code directly with other people/project groups
- post your code in a public place

We will run tools to check for similar code across individuals

5) ATTEND DISCUSSION SECTIONS

12 sections on Wednesday

- Approx. 30 students each
- Highly recommended to attend
- Lead by TAs
- Will go over practice problems and discussion solutions

6) PREPARE FOR EXAMS

Exam (45% of final grade)

- Assess OS concepts discussed in class
- One in-class midterm (10/20) and one final exam (12/21)
- Cumulative exams (final will have an emphasis on the second half)

Pick resources that best fit your need

- Lecture notes
- Review sessions
- Piazza discussion
- Group study

7) ASK FOR ADDITIONAL HELP

Teaching Assistants

- Himanshu Sagar
- Abby Matthews
- Jinlang Wang
- Suyan Qu
- Sambhav Satija
- Tingjia Cao
- Evan Wireman
- Hardik Chauhan
- Chahak Tharani

Peer Mentors

- Yuxiao Qu
- Adam Schmidt
- Nevindu Batagoda
- Sarvesh Tandon
- Zhaoyang Li
- Zihan Zhao
- Hongyu Fu
- Wenxin Zhang
- Shivangi Mishra
- Win San
- Ethan Hood
- Dana Schneck
- Zelong Wang
- Leping Li
- Yuhan Zhang
- Eric Gjerde
- Jianbang Sun

TODAY'S AGENDA

1. What will you do in this course?

2. **What is an operating system and why do we need one?**

3. Why should you care?

LEARNING OUTCOMES

1. High-level understanding of an Operating System and role it plays
2. Understand how an OS virtualizes physical resources, the abstractions an OS provides and how they are efficiently implemented w/ current hardware
3. Create correct multi-threaded applications using synchronization primitives
4. Understand how OS ensures information persists despite power outages, crashes, and failures
5. Implement open-ended programming projects (alone and w/ partner)
6. Use existing system calls and add functionality to a simplified OS

EXAMPLES

macOS

Windows 11

archlinux™

OPERATING SYSTEM DEFINITION

Users

Applications

Operating System

Hardware

Software that *manages resources* and provides easy-to-use *abstractions* for its applications and users.

WHAT DO OPERA TING SYSTEMS PROVIDE: ROLE #1

Abstraction: Provide standard library or interface for **resources**

What is a **resource**?

- Virtual or physical hardware of limited capacity (e.g., CPU, GPU, memory, disk)
- What abstraction does modern OS typically provide?
  - CPU: processes and threads
  - Memory: virtual addresspace
  - Disk: files, directories, links

WHY SHOULD OS PROVIDE ABSTRACTIONS?

Advantages

- Provide higher-level or more useful functionality than raw device
- Allow applications to reuse common facilities
- Make different devices look the same to the application

Challenges

- What are the correct abstractions?
- How much of hardware should be exposed?

WHAT DO OPERA TING SYSTEMS PROVIDE: ROLE #2

Resource management – Share resources well

What are we sharing resources among?

- Multiple users of the system
- Multiple applications running on the same system

WHY SHOULD OPERATING SYTEMS MANAGE RESOURCES?

Advantages

- Protect applications from one another at a common layer
- Provide efficient access to resources (cost, time, energy)
- Provide fair access to resources

Challenges

- What are the correct policies? (which? when?)
- What are the correct mechanisms? (how?)

SUMMARY: OPERATING SYSTEM ROLES

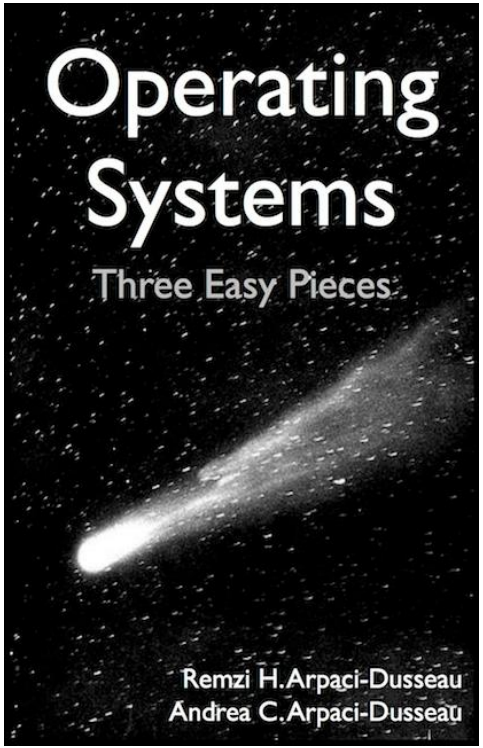
Two main roles

- Abstraction
- Resource management

Common layer between applications and hardware

Number of design, implementation challenges

THREE EASY PIECES



Structure of this lecture and the text book

1. Virtualization

2. Concurrency

3. Persistence

1) VIRTUALIZATION

Make each application believe it has each resource to itself

OS needs to

- Allow multiple applications to use the same resource
- Provide a (mostly) fair scheduling mechanism
- Support mutually distrusting applications

Example: Memory virtualization

2) CONCURRENCY

Events occur “simultaneously” and may interact with one another

- E.g., reading and updating a variable in memory

OS needs to

- Hide concurrency from independent processes
- Provide synchronization primitives for interacting processes (and threads) to use
  - E.g., locks, semaphores, condition variables

Example: Concurrency

3) PERSISTENCE

Why do we need persistence?

- Lifetime of data is longer than the lifetime of any one process
- Machine may lose power or crash unexpectedly

OS needs to

- Provide storage abstractions: Files, directories (folders), links
- Ensure crash consistency: Failures may occur during a write
- Manage performance: Disks are very slow!

4) ADVANCED TOPICS

Virtualization
Concurrency
Persistence
Advanced Topics

- Other Storage Devices: Flash-based SSDs
- Network and Distributed File Systems
- Cloud Computing (if we have time)

TODAY'S AGENDA

1. What will you do in this course?
2. What is an operating system and why do we need one?
3. **Why should you care?**

WHY STUDY OPERATING SYSTEMS?

1) Learn to build, modify, or administer an operating system

2) Fun and challenging to understand large, complex, concurrent systems

3) Behavior of OS impacts applications and every other layer of a system

4) You are guaranteed to encounter content of this class in your future career!

NEXT STEPS

Check Canvas pages and Piazza regularly!

Do not hesitate to ask questions

- During class, after class, at office hours, or on Piazza
- My office hours (for now) are Tue/Thu 3pm-4pm in CS4226
- Please only email Remzi or me directly for urgent or confidential matters

Welcome to CS 537!