CS 537: INTRO TO OPERATING SYSTEMS

Shivaram Venkataraman

Spring 2023

WHO AM I?

Fifth year faculty in Computer Science

PhD Thesis at UC Berkeley:
System Design for Large Scale Machine Learning

Industry: Google, Microsoft Research
Open source: Apache Spark committer

CALL ME

Prof. Shivaram or Shivaram

TODAYS AGENDA

What will you do in this course?

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

Why study operating systems?

COURSE SYLLABUS

COURSE LEARNING OUTCOMES

- Explain the fundamental types of OS abstractions

- Design and implement system libraries and kernel calls

- Assess system performance

- Explain the impact of algorithms and data structures

ASSESSMENTS

Exams (40%)

Three midterm exams, all in-person.

Quiz (10%)

In-class: Bring your computing device (or use paper)!

Assess OS concepts, abstractions discussed in class

Best 20 (out of ~25 lectures)

Projects (50%)

Programming projects done on CS Linux labs

Gain hands-on experience, Build your own OS system calls!

Measure, understand performance

FORMAT

Lecture

Tue and Thu, 1PM - 2:15PM
Location: 1310 Sterling Hall

In-person, synchronous
Lecture notes, in-class discussion

Discussion

Wednesdays
Multiple sections

Explain programming projects
Practice for exams

PERSONNEL: TWO SECTIONS

Instructors: Mike Swift, Shivaram Venkataraman

Teaching assistants: Anjali, Abigail Matthews, Ajay Joshi, Johannes Freischeutz, Keren Chen, Rahul Chundru, Sunaina Krishnamoorthy

17 course staff!

Peer mentors: Leping Li, Jack Xu, Vivian Zhang, Abhay Punjabi, Casilda Lewis, Zihan Li, Xinyu Zhou, Marco Kurzyinski

OFFICE HOURS

My office hours

Thursday 3pm-4pm at CS 7367

TA/Peer Mentor office hours

At CSL labs

Check Piazza, Canvas

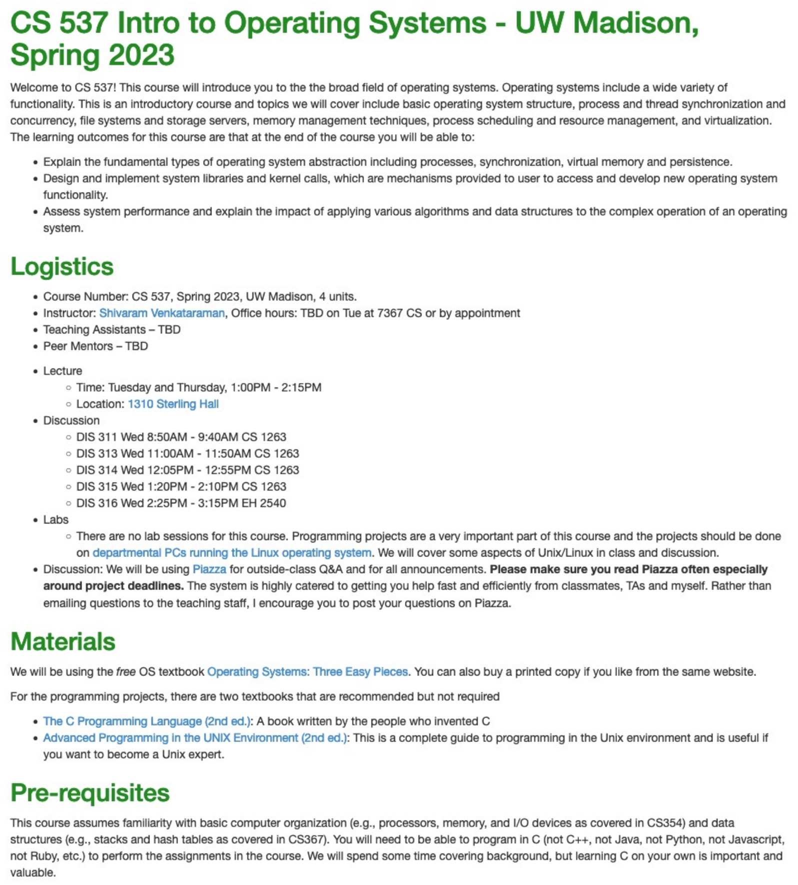
IMPORTANT LINKS

Course website

http://pages.cs.wisc.edu/~shivaram/cs537-sp23/

Piazza

https://piazza.com/wisc/spring2023/cs537



CS 537 Intro to Operating Systems - UW Madison, Spring 2023

Welcome to CS 537! This course will introduce you to the the broad field of operating systems. Operating systems include a wide variety of functionality. This is an introductory course and topics we will cover include basic operating system structure, process and thread synchronization and concurrency, file systems and storage servers, memory management techniques, process scheduling and resource management, and virtualization. The learning outcomes for this course are that at the end of the course you will be able to:

- Explain the fundamental types of operating system abstraction including processes, synchronization, virtual memory and persistence.
- Design and implement system libraries and kernel calls, which are mechanisms provided to user to access and develop new operating system functionality.
- Assess system performance and explain the impact of applying various algorithms and data structures to the complex operation of an operating system.

Logistics

- Course Number: CS 537, Spring 2023, UW Madison, 4 units.
- Instructor: Shivaram Venkataraman, Office hours: TBD on Tue at 7367 CS or by appointment
- Teaching Assistants – TBD
- Peer Mentors – TBD

- Lecture
  - Time: Tuesday and Thursday, 1:00PM - 2:15PM
  - Location: 1310 Sterling Hall
- Discussion
  - DIS 311 Wed 8:50AM - 9:40AM CS 1263
  - DIS 313 Wed 11:00AM - 11:50AM CS 1263
  - DIS 314 Wed 12:05PM - 12:55PM CS 1263
  - DIS 315 Wed 1:20PM - 2:10PM CS 1263
  - DIS 316 Wed 2:25PM - 3:15PM EH 2540
- Labs
  - There are no lab sessions for this course. Programming projects are a very important part of this course and the projects should be done on departmental PCs running the Linux operating system. We will cover some aspects of Unix/Linux in class and discussion.
- Discussion: We will be using Piazza for outside-class Q&A and for all announcements. **Please make sure you read Piazza often especially around project deadlines.** The system is highly catered to getting you help fast and efficiently from classmates, TAs and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza.

Materials

We will be using the *free* OS textbook Operating Systems: Three Easy Pieces. You can also buy a printed copy if you like from the same website.

For the programming projects, there are two textbooks that are recommended but not required

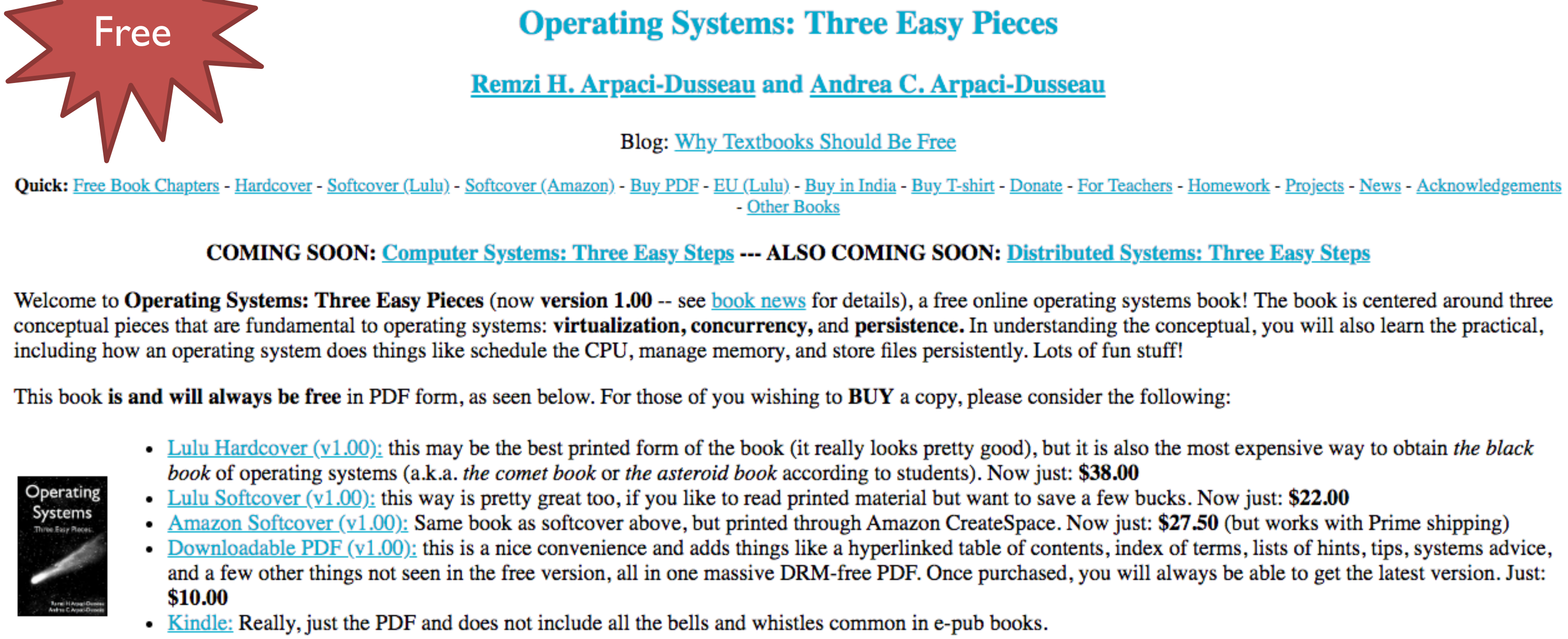
- The C Programming Language (2nd ed.): A book written by the people who invented C
- Advanced Programming in the UNIX Environment (2nd ed.): This is a complete guide to programming in the Unix environment and is useful if you want to become a Unix expert.

Pre-requisites

This course assumes familiarity with basic computer organization (e.g., processors, memory, and I/O devices as covered in CS354) and data structures (e.g., stacks and hash tables as covered in CS367). You will need to be able to program in C (not C++, not Java, not Python, not Javascript, not Ruby, etc.) to perform the assignments in the course. We will spend some time covering background, but learning C on your own is important and valuable.

MATERIALS

Free



Operating Systems: Three Easy Pieces

Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau

Blog: Why Textbooks Should Be Free

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

COMING SOON: Computer Systems: Three Easy Steps **--- ALSO COMING SOON:** Distributed Systems: Three Easy Steps

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!

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: **$27.50** (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 PDF. Once purchased, you will always be able to get the latest version. Just: **$10.00**
- Kindle: Really, just the PDF and does not include all the bells and whistles common in e-pub books.

COURSE POLICIES: TIME MANAGEMENT

Time management is a skill to learn!

Projects are mostly back-to-back – start early

Ask for help (email or OH) if you have any issues

Slip days: Maximum of three slip days.

Once you have used all your slip days,

- 100% of points if turned in on or before the deadline,
- 80% if turned in a day late
- 60% if 2 days late.

COURSE POLICIES: ACADEMIC INTEGRITY

It is DEFINITELY OK to:

- discuss the project in general terms (what do they mean by a file?)
- discuss how different library routines/system calls work
- ask the TA or professor or both for as much help as you need!

It is NOT OK to:

- bug someone else for a lot of help (particularly if they are done!)
- share your code directly with other people/project groups

COURSE POLICIES: INCLUSION

Create an environment where everyone can learn and thrive

Always feel free to ask a question!

Create a climate where we treat everyone with respect

SUMMARY

Course outline

- OS abstractions: Principles + Code
- Exams, programming projects
- Operating system: Three Easy pieces textbook

Action items: Register on Piazza and check course website!

WHAT IS AN OPERATING SYSTEM ?

EXAMPLES OF OPERATING SYSTEMS

MS DOS

Mac OS

OS X Yosemite

Microsoft® Windows® xp™

Applications

Hardware

WHAT DOES OS PROVIDE: ROLE #1

Abstraction: Provide standard library to access resources

What is a resource?

- Anything valuable (e.g., CPU, memory, disk)
- Examples of abstractions OS typically provide?
  - CPU:
  - Memory:
  - Disk:

WHY SHOULD OS DO THIS ?

Advantages of OS providing abstraction?

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

Challenges

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

WHAT DOES OS PROVIDE: ROLE #2

Resource management – Share resources well

What is sharing?

- Multiple users of the system
- Multiple applications run by same user
- Multiple devices for same functionality

WHY SHOULD OS DO THIS ?

Advantages of OS providing resource management

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

Challenges

- What are the correct mechanisms?
- What are the correct policies?

OPERATING SYSTEM ROLES SUMMARY

Two main roles

- Abstraction
- Resource management

Goals

- Ease of use
- Performance
- Isolation
- Reliability
- …

COURSE APPROACH

OPERATING SYSTEMS: THREE EASY PIECES

Three conceptual pieces

1. Virtualization
2. Concurrency
3. Persistence

VIRTUALIZATION

Make each application believe it has each resource to itself

Demo

CONCURRENCY

Events occur simultaneously and may interact with one another

Need to

- Hide concurrency from independent processes
- Manage concurrency with interacting processes

Provide abstractions (locks, semaphores, condition variables etc.)

Demo with threads

PERSISTENCE

Lifetime of data is longer than lifetime of any one process

Machine may lose power or crash unexpectedly

Issues:

- High-level abstractions: Files, directories (folders), links
- Correctness with unexpected failures
- Performance: disks are very slow!

ADVANCED TOPICS

Virtualization

Concurrency

Persistence

Advanced Topics

- Network File Systems
- SSDs
- Cloud Computing

WHY STUDY OS ?

Build, modify, or administer an operating system

Understand system performance

- Behavior of OS impacts entire machine
- Tune workload performance
- Apply knowledge across many layers

Fun and challenging to understand large, complex systems

WAITLIST

If you are on the waitlist

- Keep attending classes
- Start working on projects
- Email enrollment@cs.wisc.edu to check
- Meet instructor in office hours

NEXT STEPS

Register on Piazza

First programming assignment out tomorrow!

- Due on February 1 (one week!)
- More details in discussion sections this week.

Welcome to CS 537!