Hello!

CS 537: INTRO TO OPERATING SYSTEMS

Louis Oliphant and Shivaram Venkataraman

Fall 2024

WHO AM I?

Six years as a 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 → textbook, lectures

- Design and implement system libraries and kernel calls → programming assignments

- Assess system performance

- Explain the impact of algorithms and data structures

ASSESSMENTS

Exams (40%)

Three midterm exams, all in-person.

→ Midterm 1
Midterm 2
Midterm 3

Quiz (5%)

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

Assess OS concepts, abstractions discussed in class

Best 20 (out of ~25 lectures)

Projects (50%) + Code review (5%)

~6 Programming Projects → graded automatically

Programming projects done on CS Linux labs

Code review for 1 project → feedback

FORMAT

Lecture

Section 1 (Prof. Louis Oliphant)
Tue and Thu, 9:30AM - 10:45AM
AB20, Geological Sciences

Section 2 (Prof. Shivaram)
Tue and Thu, 11AM - 12:15PM
Location: 132 Noland Hall

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

Discussion → led by TAs

Wednesdays
Multiple sections

Explain programming projects
Practice for exams

PERSONNEL: TWO SECTIONS

Instructors: Louis Oliphant, Shivaram Venkataraman

Teaching assistants: Seunghyun An, Vojtech Aschenbrenner, Leshna Balara, Aditya Das Sarma, Fariha Islam, Robert Nagel, Omid Rostamabadi, John Shawger

22 course staff!

Peer mentors: Laura Kuo, Lucas Abreu Sernik, Arnav Jhingran, Nikhil Sethuram Thenmozhi, Shangyuan Yang, Mengze Teng, Zeren Yang, Dhruv Desai, Samad Syed, Naman Sogani, William Xia, Thi Dao

OFFICE HOURS

My office hours

Thursday 3pm-4pm at CS 7367

TA/Peer Mentor office hours

At CSL labs

Check Piazza, Course Website

IMPORTANT LINKS

Course website

http://pages.cs.wisc.edu/~shivaram/cs537-fa24/

Piazza

https://piazza.com/wisc/fall2024/cs537

↓

Best way to communicate!

CS 537 Intro to Operating Systems - UW Madison, Fall 2024

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, Fall 2024, UW Madison, 4 units.
- Instructors:
  - Louis Oliphant, Office hours: TBD
  - Shivaram Venkataraman, Office hours: Tuesday 3-4pm in CS 7367
- Teaching Assistants and Peer Mentors
- Lecture
  - Section 1
    - Time: Tuesday and Thursday, 9:30AM - 10:45AM
    - Location: AB 20 Weeks Hall for Geo Sciences
  - Section 2
    - Time: Tuesday and Thursday, 11:00AM - 12:15PM
    - Location: 132 Noland Hall
- Discussion
  - DIS 311 Wed 8:50AM - 9:40AM CS 1325
  - DIS 312 Wed 9:55AM - 10:45AM CS 1263
  - DIS 313 Wed 11:00AM - 11:50AM CS 1263
  - DIS 314 Wed 12:05PM - 12:55PM CS 1257
  - DIS 315 Wed 1:20PM - 2:10PM EH 2540
  - DIS 316 Wed 2:25PM - 3:15PM EH 2540
  - DIS 321 Wed 8:50AM - 9:40AM EH 1213
  - DIS 322 Wed 9:55AM - 10:45AM EH 1213
  - DIS 323 Wed 11:00AM - 11:50AM EH 2309
  - DIS 324 Wed 12:05PM - 12:55PM Noland 119
  - DIS 325 Wed 1:20PM - 2:10PM EH 1213
  - DIS 326 Wed 2:25PM - 3:15PM EH 2255
- 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.

MATERIALS

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!

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

Slip days: Maximum of two slip days for first 3 projects. → individually

two slip days for the last 3 projects. → group

Once you have used all your slip days,

- 100% of points if turned in on or before the deadline,
- 90% if turned in a day late, 80% if 2 days late, 70% if 3 days late

Maximum 3 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

WHAT ABOUT CHAT-GPT?

Ok to use ChatGPT / LLMs for general understanding and support
(similar to how you might use Web search, Stack Overflow etc.)

However

- You may not turn in code written by these generative AI
- You are responsible for any code you write and share with course staff (including in office hours etc.)
- Sharing code among students is still discouraged. (We will use code similarity checkers and do code review)
- Code review with TA for one project

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

WAITLIST

~196 students

~47 students

Drop the course if you are not planning to take it!

CS enrollment office manages 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

CSL machine

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!

QUIZ 0

https://tinyurl.com/cs537-fa24-quiz0

WHAT IS AN OPERATING SYSTEM ?

EXAMPLES OF OPERATING SYSTEMS

Linux ≈ server / laptop

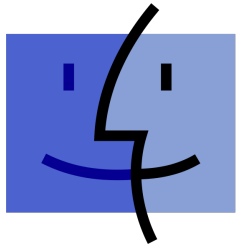
Windows

iOS → phone

Temple OS ?!?

Free RTOS → real time OS
embedded devices

simple
Unix - 6 → xv6 → educational purposes



Mac OS

OS X Yosemite

Microsoft® Windows® xp

Applications

memory

save this to disk

Operating Systems

how to do this

Hardware

WHAT DOES OS PROVIDE: ROLE #1

Abstraction: Provide standard library to access resources

abstraction

API

What is a resource?

Anything valuable (e.g., CPU, memory, disk)

Examples of abstractions OS typically provide?

CPU: Process, threads

Memory: Heap → malloc allocate memory

Stack

Disk: file, directories

permission

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

across diff hardware

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? → How is it done
- What are the correct policies? → What should be expected behavior

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 → abstractions
   make it look like every user has their own computer
2. Concurrency
   ↳ share
3. Persistence
   ↳ Abstraction
   Reliability

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!

↳ CPU or data access from memory

ADVANCED TOPICS

Virtualization

Concurrency

Persistence

Advanced Topics

- Virtual Machines
- SSDs
- Distributed Systems

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

NEXT STEPS

Register on Piazza

later today

First programming assignment out ~~tomorrow~~!

- Due in one week!
- More details in discussion sections next week.

Welcome to CS 537!