CS 537:
INTRO TO OPERATING SYSTEMS

Yuvraj Patel
University of Wisconsin-Madison
Summer 2021

Who am I?

6th Year PhD Student (Graduating this summer)
Advisors – Remzi Arpaci-Dusseau & Andrea Arpaci-Dusseau
Research Area – Operating Systems, Storage, Concurrency & Security
Thesis: Process Synchronization as a resource in Concurrent Systems

Teaching for the first time (preparing for being a Professor)
- Learn the nuances of teaching
- Share the knowledge with others
- Teaching Philosophy – Use teaching as a tool to build future

Who am I (contd..)?

Prior to Grad School:

- Spent 9 years writing Operating System and File System code
- Code written by me executes thousands of times every second
  - Bank transactions
  - Car production – BMW, Daimler Chrysler
  - Movie production - Avatar
  - Large Hedron Collider and many more….

Call Me

Yuvraj or Yuvi or UV (but not IR)

Please do not address me as Professor/Prof. Yuvraj

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?

1) Attend Lectures + Take Notes

Cover conceptual ideas

Attendance and "participation" highly encouraged - 10% of final grade

- Synchronous course
- Many zoom polls – Use for attendance
  - Check your understanding
  - Realtime feedback - Alters what I talk about!
- Lecture Notes available ahead of time
  - Annotate; not complete, expect you to fill in details

However, lectures will be recorded and available

- Additional lectures will be provided

2) Read the Textbook

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.

3) Start Projects Promptly

Projects (50% of final grade)

Seven programming projects done on CS Linux labs

- All in C
- Some use xv6 – toy OS
  - Gain hands-on experience, Build your own OS system calls!

Take significant amount of time (about 2 weeks each)

- Specifications are longer than you may be used to
- Provide test cases; grade based on how many you pass
- 3 slip days to use throughout semester

First 2 work on alone

- Last 3 with one project partner (can change)
  - Someone you know or we are happy to match

4) Don’t Cheat: Academic Integrity

It is OK to:

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

It is NOT OK to:

- use code samples for similar problems you may find on-line
- 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

Teaching assistants:

Akshat Sinha
Luke Swanson

Zoom links on Canvas

2 sections on Wednesday …

Explain programming projects
Code walk through/ debugging code
Led by TAs

6) Review Material for Exams

Exam (30% of final grade)

- Assess OS concepts discussed in class
- One evening midterm and a final exam
- Final exam will not include material before midterm
- Multiple choice questions

Learn through

- Lecture Material
- Projects
- Piazza discussion
- Group study
- Encouraged to discuss with others

7) Ask for Additional Help

- Many TA + Peer Mentor Lab Hours
  - Projects
  - Use google form to ask question

- Piazza (10% of final grade)
  - Tends to be active and prompt
  - Learn and help others

- Instructor Office Hours (Zoom)

Peer mentors:
Ritika Mittal

Today's Agenda

1. What will you do in this course?

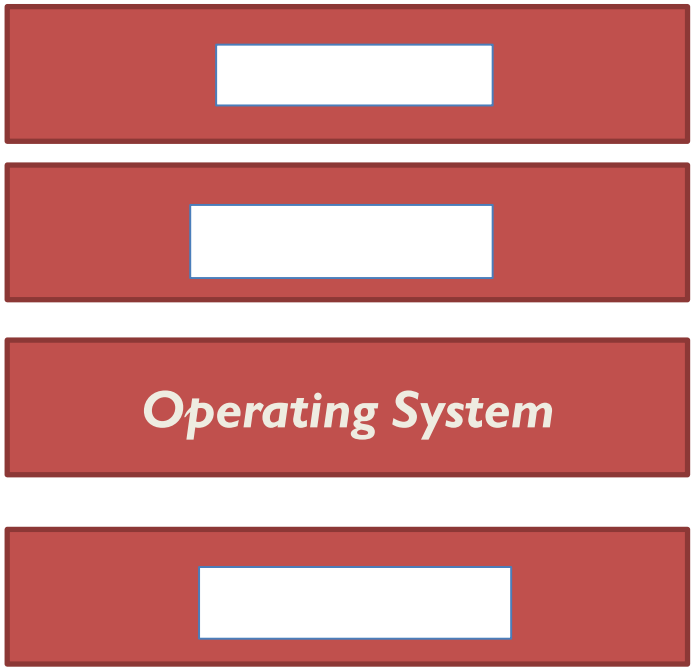
2. **What is an operating system and why do we need one?**

Learning Outcomes

1. High-level understanding of an Operating System and role it plays
2. Understand how OS [ ] physical resources, the [ ] the OS provides and how they are efficiently implemented w/ current hardware
3. Create correct multi-threaded applications using [ ] primitives
4. Understand how OS ensures information [ ] 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

OS Definition



What does OS provide: Role #1

Provide standard library or interface for **resources**

What is a **resource**?

What abstraction does modern OS typically provide?

CPU:

Memory:

Disk:

Why should OS provide Abstractions?

Advantages of OS providing abstraction?

- Provide [ ] than raw device
- Allow applications to [ ]
- Make different devices [ ]

Challenges

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

What does OS provide: Role #2

What is sharing?

- Multiple users of the system
- Multiple applications run by same user

Why should OS Manage Resources?

Advantages of OS providing resource management?

- [ ] from one another at a common layer
- Provide [ ] to resources (cost, time, energy)
- Provide [ ] to resources

Challenges

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

Summary: OS Roles

Two main roles

- Abstraction
- Resource management

Common layer between applications and hardware

Number of design, implementation challenges

Break Time?

- You form a unique community
- Get to know your classmates
- Introduce yourself to people around you
  - Your name
  - What year are you in school?
  - What other classes are you taking?
  - Have you already taken any that they are – if so, advice?

Operating Systems: Three Easy Pieces

1. Virtualization

2. Concurrency

3. Persistence

1) Virtualization

Make each application believe

Demo: Virtualize CPU and memory

2) Concurrency

Events occur "simultaneously" and may interact with one another

- Reading and updating a variable in memory

Need to

- Hide concurrency from independent processes
- Manage concurrency with [ ] processes

Provide synchronization primitives for [ ] processes to use (locks, semaphores, condition variables, etc.)

Demo: Threads

3) Persistence

than lifetime of any one process

Machine may lose power or crash unexpectedly

Issues:

- High-level abstractions:
- Crash consistency:
- Performance:

Demo: Persistence

4) Advanced Topics

Virtualization

Concurrency

Persistence

Advanced Topics

- Other Storage Devices: Flash-based SSDs
- Network and Distributed File Systems (NFS + AFS)

Why Study Operating Systems?

1) 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 system
   - Understand and tune workload performance
   - Apply knowledge across many layers

Next Steps

Check out Canvas pages and Piazza

Project 0 available – Complete it ASAP (no grading)
Project 1 available now
- More details in discussion on Wednesday
- Due soon…
- Office hours of TAs and Peer Mentors start Wednesday

Welcome to CS 537!