WHO AM I?

First 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
TODAY'S AGENDA

What will you do in this course?

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

Why study operating systems?
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 (50%)
- Midterm and final exams
- Closed book
- Assess OS concepts, abstractions discussed in class

Projects (50%)
- Five programming projects done on CS Linux labs
- Gain hands-on experience, Build your own OS system calls!
- Measure, understand performance
Lecture
Tue and Thu, 2:30PM - 3:45PM
Location: 1125 BioChem
Lecture notes, in-class discussion
Links to textbook chapters

Discussion
Thu 5:30PM-7:00PM
Location: 105 Psychology
Hands-on code walk through
Explain programming projects
Instructor: Shivaram Venkataraman

Teaching assistants: Saurabh Agarwal, Ram Alagappan, Alex Anderson, Setareh Behroozi, Yunang Chen, Varsha Pendyala, Yufei Wang

Peer mentors: Arpit Jain, Anshu Verma, Xianjie Zheng, Siddhant Bhagat, Benjamin Yin, Youmin Han, Varun Ravipati, Yingdong Chen
IMPORTANT LINKS

Course website
http://pages.cs.wisc.edu/~shivaram/cs537-sp19/

Piazza
https://piazza.com/wisc/spring2019/cs537
Operating Systems: Three Easy Pieces

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

Blog: Why Textbooks Should Be Free

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

Slip days: Maximum of three slip days through the semester. Use with care!

Academic integrity for projects:
It is DEFINITELY OK to:
- discuss the project in general terms (what do they mean by a pipe?)
- 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 already done!)
- share your code directly with other people/project groups
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?
Operating System

Users

Applications

Hardware
WHAT DOES OS PROVIDE: ROLE #1

Abstraction: Provide standard library for resources

What is a resource?
   Anything valuable (e.g., CPU, memory, disk)
What abstraction does modern OS typically provide?
   CPU: process and/or thread
   Memory: address space
   Disk: files
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?
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

Common layer to implement roles
Number of design, implementation challenges
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:** Virtualize CPU and memory
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
  Virtual Machines
  Network File Systems
  SSDs
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, please email

enrollment@cs.wisc.edu
NEXT STEPS

Register on Piazza

First programming assignment out by tonight!
   Due Jan 29th, (next Tuesday) at 11.59pm
   More details in discussion on Thursday

Welcome to CS 537!