CS 153
Design of Operating Systems

Fall 19

Lecture 1: Course Introduction

Instructor: Chengyu Song

Slide contributions from

Nael Abu-Ghazaleh, Harsha Madhyvasta and Zhiyun Qian

Teaching Staff

- Chengyu Song
  - I am an Assistant Professor in CSE
    - » 4th year at UCR, **feedback is appreciated!!**
  - Office hours Monday and Friday 1pm-2pm or by appointment
    - » Hope to meet many of you during office hours

- 2 TAs for 3 lab sessions
  - Gaurav Ranjan Basu and Ali Nowraiz Khan
  - Office hours TBA
  - Leads for Labs

Class Overview

- Check class webpage for information
  - https://www.cs.ucr.edu/~csong/cs153/19f/
  - Will send out link to webpage

- Lecture slides, homeworks, and projects will be posted on class webpage
- Assignment turn-in through iLearn
  - Digital only, no paper copy
  - Announcements through iLearn and posted on class webpage
- Piazza for discussion forums; link on website
  - Use these please
  - Stay on top of things – falling behind can snowball quickly into trouble

Textbook

- Apraci-Dessau and Apraci-Dessau**, OS, 3 easy pieces** (required + free!)
  - Please read, most chapters are only a few pages

- Other good books:
  - Anderson and Dahlin, ***Operating Systems: Principles and Practice (recommended)***
  - Silberschatz, Galvin, and Gagne, ***Operating System Concepts***, John Wiley and Sons, 8th Edition **(recommended)**

Class Overview

- Grading breakdown
  - Projects (40% total)
    - Xv6 Operating system
    - Book uses examples from it
    - 4 projects (used to be 2, splitting into halves)
  - 4 homeworks (20% total)
  - Mid-term (16%)
  - Final (24%)
  - Engagement/extra credit (4%)
    - Includes attendance in lab and lecture
    - You learn much better if you are interested and engaged

Objectives of this class

- In this course, we will study typical problems that an OS to address and the corresponding solutions
  - Focus on **concepts** rather than particular OS
  - Specific OS for examples

- Practice your engineering skills
  - Abstraction and implementation
  - The projects are very close to real projects in industry

- Develop an understanding of how OS and hardware impacts application performance and reliability

Projects

- Project framework this time: xv6
  - Projects are in C
  - Very good debugging support
  - Used in OS class at several other universities

- Start to get familiar immediately
  - We will start labs. next Friday
  - Go over the xv6 documentation (on the course web page)
  - Optional Lab 0 to help get familiar with what xv6 is

Projects are HARD!

- Probably the hardest class you will take at UCR in terms of development effort
  - You must learn gdb if you want to preserve your sanity! ☺
- Working on the projects will take most of your time in this class
- Biggest reason the projects are hard: legacy code
  - You have to understand existing code before you can add more code
  - Preparation for main challenge you will face at any real job

Project Recommendations

- Easier if you are engaged/excited
- Find a partner that you like/trust
- Do not start working on projects at last minute!
  - A lot of the time will be spend understanding the code
  - Debugging is integral process of development

- Make good use of help available
  - Post questions on piazza
  - Take advantage of TA office hours
  - Come prepared to Labs
  - Again, learning to debug

Project Logistics

- Projects to be done in groups of two
  - When you have chosen groups, send your group info to your TA
    - Ask TA for permission if your partner is in another lab session
  - Use the find a partner feature in piazza
    - email if unable to find partner and we'll form groups
  - Option to switch partners after project two

- First step is to conceptually understand the project
  - Then come up with implementation plan
    - Fail and fail again
    - Debug, debug, debug (systems are unforgiving)
    - →success!!

Homeworks and Exams

- Four homeworks
  - Assigned on first day of each section, due the first day of the next section
  - Can expect similar questions on the exams
- Midterm (tentatively Oct 30)
  - In class
- Final (Dec 11, 7:00pm-10:00pm)
  - Covers second half of class + selected material from first part
    - I will be explicit about the material covered
    - Because first midterm is short (50 minutes)
- **No makeup exams**
  - Unless dire circumstances

Submission Policies

- Homeworks due on ilearn by the end of the day (will be specified on ilearn)

- Code and design documents for projects due by the end of the day (similarly will be specified on ilearn)

- Late policy (also on course webpage):
  - 10% penalty for every late day (rounded up in days)
  - Maximum penalty is 50%

Recipe for SUCCESS in CS153

- **Start early on projects!!!**
- Attend labs and office hours
  - Take advantage of available help
- Be engaged, interested, curious
- Try to read textbook material before class
- Make sure you can finish the homeworks by yourself
- Ask questions when something is unclear
  - 4% participation and extra credit – may bump up your grade if on borderline. Face recognition ☺

How Not To Pass CS 153

- Wait until the last couple of days to start a project
  - We’ll have to do the crunch anyways, why do it early?
  - The projects cannot be done in the last few days

  - Repeat: **The projects cannot be done in the last few days**

  - Each quarter groups learn that starting early meant finishing all of the projects on time…and some do not

- Do not finish the homework

Questions for today

- Why do we need operating systems course?
- Why do we need operating systems?
- What does an operating system need to do?
- Looking back, looking forward

Why an OS class?

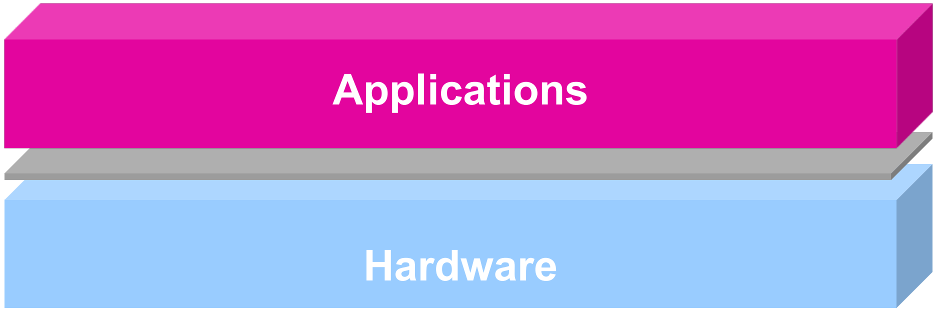
- Why are we making you sit here today, having to suffer through a course in operating systems?
  - After all, most of you will not become OS developers
- The concepts/problems are very general
  - We also encounter these problems in our daily life
  - Many abstractions like threads and synchronization are used pervasively in computer science
- Learn about **complex** software systems
  - Many of you will go on to work on large software projects
  - OS serve as examples of an evolution of complex systems
- Understand what you use (and build!)
  - Understanding how an OS works helps you develop apps
  - System functionality, debugging, performance, security, etc.

Questions for today

- Why do we need operating systems course?
- Why do we need operating systems?
- What does an operating system need to do?
- Looking back, looking forward

Why have an OS?

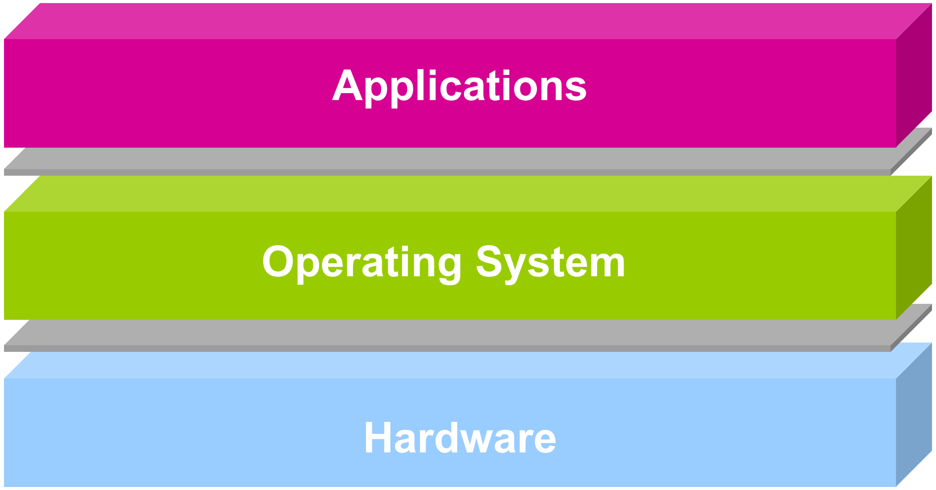
- What if applications ran directly on hardware?



- Problems:
  - Portability → OS Task 1: abstraction
  - Resource sharing → OS Task 2: multiplexing

What is an OS?

- The operating system is the software layer between user applications and the hardware



- The OS is "**all the code that you didn't have to write**" to implement your application

Questions for today

- Why do we need operating systems course?
- Why do we need operating systems?
- What does an operating system need to do?
- Looking back, looking forward.

Fundamental Issues

- The fundamental issues/questions in this course are:
  - Management: how to allocate and schedule resources?
  - Performance: how to do better?
  - Protections: how to make sure things won't go wrong?
  - Security: how to create a safe environment?
  - Communication: how to enable collaboration?
  - Reliability and fault tolerance: how to mask failures?
  - Usability: how to enable the users/programs to do more?

Basic Roles of an OS

- **Abstraction**: defines a set of logical resources (objects) and well-defined operations on them (interfaces)

- **Virtualization**: isolates and multiplexes physical resources via spatial and temporal sharing

- **Access Control**: who, when, how
  - Scheduling (when): efficiency and fairness
  - Permissions (how): security and privacy

- **Persistence**: how to keep and share data

Other Questions to Ponder

- What is part of an OS? What is not?
    - Is the windowing system part of an OS? Browser? Java? Apache server? Compiler? Firmware?
- How different/similar between OS?
    - Windows, Linux, macOS, Android, iOS, etc.
- What are the drivers of OS change?
    - Performance, functionality, usability, security, etc.
    - The UNIX Operating System
- What are the most compelling issues facing OS today?

For next class...

- Browse the course web (especially xv6 docs)

  https://www.cs.ucr.edu/~csong/cs153/19f/

- Read module 2 in textbook

- Start …
  - … tinkering with xv6
  - … attempting lab 0
  - … finding a partner for project group