

CS 153

Design of Operating Systems

Fall 21

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
 - » 6th year at UCR, **feedback is appreciated!!**
 - ◆ Office hours MWF 3pm-4pm or by appointment
 - » In person and online
- 2 TAs for 3 lab sessions
 - ◆ Nathan Brennan and Mingjun Yin
 - ◆ Office hours and links on iLearn and Canvas
 - ◆ Leads for Labs

Class Overview

- Check class webpage for information
 - ◆ <https://www.cs.ucr.edu/~csong/cs153/21f/>
- Lecture slides, homework, and projects will be posted on class webpage, iLearn, and Canvas
 - ◆ All lectures will be live streamed and recorded
- Assignment and exams turn-in through GradeScope
- Piazza for discussion forums (visit through Canvas)
 - ◆ Please use these tools, we will try to answer question ASAP
 - ◆ 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 + 5% bonus)
 - » xv6 operating system
 - » Book uses examples from it
 - » 4 projects (w/ bonus)
 - ◆ 4 homework (24% total)
 - ◆ Mid-term (15%)
 - ◆ Final (16%)
 - ◆ Engagement (5%)
 - » Attendance in lab and lecture
 - » Participation on Piazza
 - » You learn much better if you are interested and engaged

Why taking this class?

- It is mandatory (part of degree requirement)
- Why do we require you to take the OS class?
 - ◆ It is hard (at least the rumor says)
 - ◆ You are unlikely to become OS developer
- Knowledge
 - ◆ Know what you're using everyday
 - ◆ Problems and solutions
- Skills
 - ◆ The ability to **understand** complex code base
 - ◆ The ability to **debug** complex programs

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 a 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

Objectives of this class

- Three levels of understandings
 - ◆ Lv1: the concept/idea → easy
 - ◆ Lv2: the algorithm to realize the idea → medium
 - ◆ Lv3: implement the algorithm on real hardware → hard
- An example
 - ◆ Concept: synchronization
 - ◆ Algorithm: semaphore
 - ◆ Implementation: xv6

Objectives of this class

- Is CS153 a difficult class?
 - ◆ Lectures: mostly concepts → easy
 - ◆ Homework and exams: have algorithms → medium
 - ◆ Labs: implementations → hard
- My expectations
 - ◆ Lv1: remember you had learned the concepts and solutions
 - ◆ Lv2: know how to apply the solutions to solve real problems (not necessarily computer problems)
 - ◆ Lv3: can realize the solution and make it work

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
 - ◆ 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**
 - ◆ **Think in abstractions**
- 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 homework
 - ◆ Assigned on first day of each section, due the first day of the next section
 - ◆ Can expect similar questions on the exams
- Midterm (tentatively Nov 1)
 - ◆ In class
- Final (Dec 6, 8:00am-11:00am)
 - ◆ 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 with dire and documented circumstances

Submission Policies

- Homework due on GradeScope by the end of the day (23:59pm)
- Code and design documents for projects due by the end of the day (will be specified on iLearn and Canvas)
- 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 homework by yourself
- Ask questions when something is unclear
 - ◆ 5% participation 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

- What is “operating”?
- What is an operating system?
- Why do we need operating systems?
- How does an operating system function?

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/cs153/21f/>
- Read module 2 in textbook
- Start ...
 - ◆ ... tinkering with xv6
 - ◆ ... attempting lab 0
 - ◆ ... finding a partner for project group