Content of this lecture

- Course information (personnel, policy, schedule, misc.)
- What is OS? What does it do?
- History of OS
- Summary
Some Survey

- How many MS/PhD graduate students?
  - If >5 graduate students, we will separate graduate student and undergraduates in the final letter grading

- How many CS majors vs. ECE/others?

- How many know Java?

- Why are you taking this class?
About this course...

Principles

- System concepts
- OS design
- Some theory
- Rationale
- Practice

Goals

- Understand OS decisions
- Basis for future learning
- Get hands dirty
Why are you here?

- Fulfill the requirement
- Prerequisite for other courses
  - Network
  - Distributed systems
  - Real-time systems
  - Multimedia systems
- Future plans
  - Graduate schools for CS
  - Employment
What does the class cover?

- Introduce you to operating system concepts
  - Hard to use a computer without interacting with OS
  - Understanding the OS makes you a more effective programmer

- Cover important systems concepts in general
  - Caching, concurrency, memory management, I/O, protection

- Teach you to deal with larger software systems
  - Programming assignments much larger than many courses
  - Warning: Many people will consider course very hard
  - In past, majority of people report 15 hours/week
Why Study OS?

- Operating systems are a maturing field
  - Most people use a handful of mature OSes
  - Hard to get people to switch operating systems
  - Hard to have impact with a new OS
- High-performance servers are an OS issue
  - Face many of the same issues as OSes
- Resource consumption is an OS issue
  - Battery life, radio spectrum, etc.
- Security is an OS issue
  - Hard to achieve security without a solid foundation
- New “smart” devices need new OSes
- Web browsers increasingly face OS issues
Who am I?

- Yuanyuan (YY) Zhou
- Research: operating systems, software reliability, computer architecture, storage systems
- Brief BIO
  - Ph.D, Princeton, 2000
  - NEC Research, 2000-2002
  - UIUC, Professor, 2002-2009 (mostly teach OS)
  - Co-founded 3 companies (2 were acquired)
    - Make me probably the most “industrial” professor in CSE
You are advised to **drop my class**

- You have insufficient pre-requisite
  - CSE 30 (Computer Organization and Systems Programming),
  - CSE 101 (Design and Analysis of Algorithms)
    - CSE100 (data structures)
  - CSE 70 or CSE 110 (Software Engineering)
  - We have a pre-requisite test next Tuesday

- You want an easy class
  - You don’t care about learning or your future job search

- You cannot understand or tolerate my accent
Personnel

- **Instructor:**
  - Yuanyuan Zhou ([yyzhou@cs.ucsd.edu](mailto:yyzhou@cs.ucsd.edu))
  - Office hours (EBU3B 3210):
    - Tu/Thu 3:30am-4:30pm

- **Teaching Assistants**
  - Tianyin Xu ([tixu@eng.ucsd.edu](mailto:tixu@eng.ucsd.edu))
  - Saurabh Bipin Chandra, [sbipinch@eng.ucsd.edu](mailto:sbipinch@eng.ucsd.edu)
  - Office hours: on class website

- **Discussion section:**
  - Friday, 3:00pm - 3:50pm, Center 214
Class Information

- Reference Textbook: 

- Class Website
  http://cseweb.ucsd.edu/classes/fa16/cse120-b/

- Slides, schedule, grading policy, etc. will be available on the class web page.
Course Information

- Course material taught through class lectures, textbook readings, and handouts
- Course assignments
  - Four large programming projects
  - Homework (for exam preparation)
- Discussion sections for asking questions on
  - Lecture material and homework
  - Clarification and supplementary materials
  - *(Oct 14’s discussion will be switched with Oct 28th’s lecture)*
- Piazza(a link from class web site)
Projects

- Nachos is an instructional OS
  - It is a user-level operating system and a machine simulator
  - Will become abundantly clear (or not so clear) very soon

- Programming environment will be Java on Unix (Linux)
  - The projects will require serious time commitments
    - waiting until the last minute is not an option

- You will do four projects using Nachos
  - Installation & Submission
  - Concurrency and synchronization
  - Multiprogramming
  - Virtual memory or FS

- You will work in groups of 1-3 on project 1-3
  - Start looking for partners now
Labs

- We will use the labs in the CSE basement
  - Linux running on PCs
- You can also use your home machine
  - The same project source will work on Linux (but not Windows)
  - Note: We will test and grade on lab machines
  - Be sure to test your projects there as well
Homeworks

- 4 homeworks to reinforce lecture material and help you preparing for exams

- As long as you submit and write some technical OS-related content, you get FULL credit
  - Your choice: you can spend the time to learn something, or you can just waste it
  - Why? Avoid penalizing students who do their work without cheating (but get to taken points off)
Exams

- **Midterm**: Oct 20\textsuperscript{th}, lecture time
  - Covers first half of class + something related to projects
  - So do your project and do NOT copy!
  - They will be different from the other session

- **Final**: Covers second half of class + selected material from first part

- No makeup exams
  - Unless dire circumstances

- No cheating sheet is allowed
Grades

- Midterm: 30%
  - Containing around 5-10% of material related to projects
- Final: 30%
  - Containing around 5-10% of material related to projects
- Projects: 30%
- Homework: 10% (4 homeworks)
Re-grading policy

- Students have 1 week (after the grade for a project/exam is released) to request for re-grading.

- After the re-grading period, no re-grading request will be granted for the project/exams.
Cheating Policy

- Academic integrity
- Your work in this class must be your own - we have a zero tolerance policy towards cheating of any kind and any student who cheats will get a failing grade in the course and reported to campus
- Both the cheater and the student who aided the cheater will be held responsible for the cheating
How NotTo Pass CSE 120

- Do not come to lecture
  - It’s nice out, the slides are online, and the material is in the book anyway
  - TRUTH: Lecture material is the basis for exams

- Copy other people’s projects
  - First, it is cheating.
  - And yes, we do have tools to check for cheating in projects.
  - Well, how can you answer the questions in midterms or final exams?
How NotTo Pass (2)

- Do not ask questions in lecture, office hours, or email
  - It’s scary, I don’t want to embarrass myself
  - Asking questions is the best way to clarify lecture material at the time it is being presented

- 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
Shishi... a secret

- You can talk about many things during my office hours
  - Career, job search, startups

- My office hours for future quarters are always open to you
Any Questions?

- Before we start …

- Do you think this will be a hard class?
Pre-requisite Test

- Does not count toward your grade

- Purposes:
  - For yourself to evaluate whether you are ready to take this class
  - If you perform poorly (in the bottom 15%), we will send you an email asking you to show us a proof of pre-requisite classes (if we cannot find it online)

- If you do not take this test, we will not have your name in our grading system
  - For all students who register later, they are also required to take a pre-requisite test (maybe different versions)
What Is an OS?

Anyone?

What does it do?

Give me a few names of an OS?
  - For desktops?
  - For smart phones?

Is Virtual Machine an OS?
Is Android an OS?
Is Java Virtual Machine (JVM) an OS?
What Is an OS?

“Code” that:

- Sits between programs & hardware
- Sits between different programs
- Sits between different users

But what does it do?

- to provide an orderly and controlled allocation of the processors, memories and I/O devices among the various programs competing for them

Real life analogy:

- Government?
- Mom
OS is...

- 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.
The OS abstracts/controls/mediates access to hardware resources (what resources?)
- Computation (CPUs)
- Volatile storage (memory) and persistent storage (disk, etc.)
- Communication (network, modem, etc.)
- Input/output devices (keyboard, display, printer, camera, etc.)

The OS defines a set of logical resources (objects) and a set of well-defined operations on those objects (interfaces)
- Physical resources (CPU and memory)
- Logical resources (files, programs, names)
Benefits to Applications

- Simpler
  - no tweaking device registers
- Device independent
  - all network cards look the same
- Portable
  - Across Windows95/98/ME/NT/2000/XP/Vista/Windows 7
- Worry less about interference from other applications
What does an OS do?

Resources
- Allocation
- Protection
- Reclamation
- Virtualization

Services
- Abstraction
- Simplification
- Convenience
- Standardization

Makes computers simpler
What Is an OS?

Resources
- Allocation
- Protection
- Reclamation
- Virtualization

Finite resources
Competing demands
Examples:
- CPU
- Memory
- Disk
- Network

Linux or Windows? ➙ Democrat or Republic?
What Is an OS?

Resources
- Allocation
- Protection
- Reclamation
- Virtualization

You can’t hurt me
I can’t hurt you

Implies some degree of
safety & security

Law and order
Government
What Is an OS?

Resources
- Allocation
- Protection
- Reclamtion
- Virtualization

The OS giveth
The OS taketh away
Voluntary at run time
Implied at termination
Involuntary
Cooperative

Government
Income Tax
What Is an OS?

Resources
- Allocation
- Protection
- Reclamation
- Virtualization

illusion of infinite, private resources

Memory versus disk

Timeshared CPU

More extreme cases possible (& exist)

Social security

Government
Some Questions to Ponder

- What is part of an OS? What is not? Is the window system part of an OS? Java?

- Popular OSes today are Windows, Linux, and OS X
  - How different/similar do you think these OSes are?

- Somewhat surprisingly, OSes change all of the time
  - Consider the series of releases of NT, Linux, OS X…
  - What are the drivers of OS change?
    - New hardware, new applications

- What are the most compelling issues facing OSes today?
The Operating System Zoo

- Mainframe operating systems
- Server operating systems
- Multiprocessor operating systems
- Personal computer operating systems
- Real-time operating systems
- Embedded operating systems
- Mobile/Pads operating systems
- IoT (internet of things) operating systems
- Cloud operating systems
## Historical Comparison

<table>
<thead>
<tr>
<th></th>
<th>Mainframe</th>
<th>Mini</th>
<th>Micro/Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System $/ Worker Salary</strong></td>
<td>10:1 – 100:1</td>
<td>10:1 – 1:1</td>
<td>1:10-1:1000</td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>System utilization</td>
<td>Overall cost</td>
<td>Productivity</td>
</tr>
<tr>
<td><strong>Target</strong></td>
<td>Capacity</td>
<td>Features</td>
<td>Ease of Use</td>
</tr>
</tbody>
</table>
Summary

- Course overview
- Policy and requirement
- What is OS?
- OS history

- Next lecture: system overview (chapter 1)
After this lecture...

- Browse the course web
  - http://cseweb.ucsd.edu/classes/fa15/cse120-b/
- Read Chapters 1, 2, and 3
- Start reading Nachos (online)
- Start thinking about partners for project groups
- See me up front if you have any questions
- Let the fun begin!