Content of this lecture

- Some background check
- Course information (personnel, policy, schedule, misc.)
- More background check (optional)
- Summary
Some Survey

- How many CS majors? ECE/others?
- How many have taken CSE120 from UCSD?
- How many have taken an equivalent undergraduate OS class?
Are you ready for this class?

You should be very comfortable with virtual memory (VA translation), processes and threads, synchronization, etc.

Some checkpoints

- What is a privilege instruction? An example?
- Difference between a system call and a function call?
- Give me an example of atomic instructions
- Difference between a semaphore and a condition variable?
- Difference between hardware and software-managed TLBs?
- What could happen at a memory instruction? Can the CPU directly access the cache to get the data?
- What is an inode?
A Prerequisite Test

- Thursday lecture time
- Content: Undergraduate OS materials
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
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, Faculty, 2002-2009 (mostly teach OS)
  - Co-founded 3 companies
Any questions about me?

- Every lecture, in the beginning, you can ask me two questions
  - Can be about anything (not limited to CSE221)
Personnel

- **Instructor:**
  - Yuanyuan Zhou ([yyzhou@cs.ucsd.edu](mailto:yyzhou@cs.ucsd.edu))
  - Office Hour: Tu/Th 10-11am CSE 3210

- **Teaching Assistant**
  - Xinxin Jin (5th year Ph.D, research: OS)
  - Tianyin Xu (5th year Ph.D, research: OS)
  - Liqiong Yang
Class Information

- Reading list
  - On web page

- Reference Textbook if you need to catch some basic background

- Lecture: T/Th 12:30pm - 1:50pm

- Final: TBA

- Class Website
  http://cseweb.ucsd.edu/classes/sp16/cse221-a/
Objective #1: Principles of OS design

- why are our systems designed the way that they are?
- A fundamental issue that a system designer and implementer deals with is complexity.
- Read papers describing different approaches to dealing with complexity
  - Layers
  - Modules
  - messages
  - upcalls
Objective #2: Experience reading research papers
- Applies to any area, not just systems
- After CSE 221, you should feel comfortable picking up a paper in another course or from a proceedings
- Develop intuition for what question/issues are important, which are not

Objective #3: Experience discussing research material
- Expressing opinions and arguing points are essential skills as an effective professional
- Have your own opinions!
- Thinking vs. memorizing
Course Structure

- Material entirely based upon research papers
- In-Class Quizzes 25%
- Homeworks 15%
- Project 30%
- Final exam 30%
Read papers in reading list

- How do I make sure that you will read the papers?
  - 6 in-class Quizzes (your top 5 quiz scores are chosen for your final grade)

- Quizzes will count for 25% of your grades
Class Format

- Discussion based
  - Different from CSE120

- I ask questions
  - Volunteers to answer questions
  - Randomly pick students to answer questions

- Don’t rely on my slides
  - My slides will be questions only
    - No answers
  - I will NOT post my slides on the web
am a Nice Professor?

- First week and final week: no quiz
- hint: the first question for each paper is:
  - what problem/issue did the paper address, i.e. why was it written?
How to Read Papers

- You should not read these papers as "truth"
  - You should have your own judgement

- Critical thinking
  - Papers are arguments based upon research. You are welcome to reject the arguments, criticize the approaches, results, etc.
  - But you will need to back up your criticisms and rejections.
Homeworks

● 4 homeworks to reinforce material and help you catch up backgrounds

● Homework 0 is already released
  – Help you brush up some background you have learned from CSE120 or equivalent

● Late submission will not be accepted without prior approval of the instructor
  – Medical reasons needs doctor’s notes
  – Conference deadlines and conference travels cannot be used as excuses

● You know the deadlines and travel dates as well as the homework deadline in advance, you should know how to plan out your schedule in advance
Projects

- You will work in groups of 1-2 on the project
  - Start looking for partners now

- Topic: performance evaluation
  - More information on the web page

- Why?
  - intuition for performance of standard hardware and operations
  - everyone gets implementation experience
Exam/Quiz

- **Final Exam**
- **No makeup exam/quiz**
  - Unless dire circumstances
- Sorry, no “cheating” sheet
Grades

- Quiz: 25%
- Homework: 15%
- Final Exam: 30%
- Project/report: 30%

- Letter grades are curved based on distribution
Re-grading policy

- Students have **1 week** (after the grade for a homework/project/exam/quiz is released) to request for re-grading
- Re-grading requests need to be in **writing** and submitted after lecture
- After the re-grading period, **no re-grading request will be granted** for the project/exam/quiz
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.
- Both the cheater and the student who aided the cheater will be held responsible for the cheating
How Not To Pass CSE 221

- Do not read papers before the lecture
- Do not come to lecture
- Copy other people’s homework or projects
  - First, it is cheating.
  - And yes, we do have tools to check for cheating in projects
  - How can you pass the final exam?
  - How about quizzes?
Any Questions?

- Before we start …

- Do you think this will be a hard class?
Warmup Questions

How many lines of code do you think each OS has?

- **WinXP:**
  - 50M (http://news.com.com/Old+code+in+Windows+is+security+threat/2100-1001_3-934363.html)

- **Linux (2010):**
  - 13M -- kernel 2.4M, X11 1.8M, browser 2.0M (http://en.wikipedia.org/wiki/Linux_kernel)
Question 2

- What is part of the OS, what is not?
  - Window system part of OS? (Windows vs. Unix/X11, also Mac)
  - Web browser?
    - This went to the supreme court
  - Apache Web server?
  - HTTP protocol?
  - Java?
  - compiler? linker? loader? runtime?
  - device firmware?
Question 3

- What drives an OS design to change?
  - Hardware technology: e.g. multicore,
  - Application demands: e.g. multimedia
  - User-demands: e.g. smartphone, etc
More Background Check/ Warmup Questions

- OS
- Hardware support
  - privilege instructions, interrupt, system call, etc
- Process, threads
- Synchronization
- Memory
  - VM system, TLB, page table, etc
- File systems
  - Disk, File, directory, layout,
- Distributed systems
What Is an OS?

Anyone?

What does it do?

Benefit of OS?

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

Is VmWare an OS? Is Internet Explorer an OS?
Hardware Support

- Kernel vs. user mode?
- What is a privilege instruction? An example?
- Is OS always running on the background?
  - What is the entry to OS?
- Interrupts vs. exceptions?
- What is the interrupt vector?
- What is a fault?
- System call vs. function call?
Process

- What is a process?
- How does an OS support tens of processes running on a machine with only 2-4 processors?
- Process state?
  - What is the ready state?
- Process priority
  - What is it used for?
- Is the address used in the instruction absolute or logical?
Threads

- Threads vs. Process?
- What are shared to multiple threads of the same process? What are not?
- Benefits of threads?
- User level threads vs. kernel level threads?
Scheduling

- What is scheduling?
- Some scheduling algorithms/policies?
  - Which one has a starvation problem?
- What is time slice?
- What is the typical time slice length in Linux?
  Why?
- What is a context switch? Why is it expensive?
- How to favor I/O intensive processes?
Synchronizations

- Why do we need synchronizations in multi-threaded programs?
- What is mutual exclusion?
- How to implement critical sections?
- What are atomic instructions? Examples?
Memory

- Memory hierarchy?
- Virtual memory
  - Address translation, who does it?
  - Pages, page size (why power of 2? Can it be too big, too small?)
  - Page table?
  - TLB? Who handles TLB miss?
  - What happens to TLB in a context switch
  - Page fault and swap space?
- Replacement policies
  - Optimal algorithm?
  - LRU?
  - NRU? Clock?
Disk and File Systems

- Disk performance
- Disk scheduling?
- File, directory hierarchy
  - What is the content of a directory?
  - File system disk layout?
  - What is the problem with contiguous layout? Link-list?
  - Inode?
- File protection
  - Access control list vs. capability list
After this lecture...

- Browse the course web
- Homework 0
- Read 2 papers for Thursday class

- Start thinking about partners for project groups
  - See project page on website for setting up groups.

- See me up front if you have any questions
- Let the fun begin!