

# **CSE221**

# **Principles of Computer Operating Systems**

Prof Yuanyuan (YY) Zhou

Lecture 1



# 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?
- Undergraduate vs Graduate?

# 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?

# Why are you here?

- Fulfill the requirement
- Prepare for other courses
  - Network
  - Distributed systems
  - Security
  - Embedded systems

# 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

# My Double Career 😊

## Academia



- **UIUC, 2002-2009**
  - 2002-2006 Assistant professor
  - 2006-2009 Tenured associate professor
- **UCSD, 2009-Now**
  - Chair Professor



## Startup/Industry



- **Emphora, 2000-2002**
  - Database storage
  - Acquired by a public company
- **Pattern Insight 2007-2012**
  - System data pattern analysis
  - Acquired by VmWare in 2012
  - “Log Insight” is now used in thousands of data center via VmWare
- **Whova: Nov 2013-Now**
  - Conferences/events
  - Used by 15, 000+ conferences/events in 93 countries



# A little bit about my former & Current Students

- I have graduated 18 Ph.Ds so far
  - Currently still advising 8 Ph.D students
- 6 former Ph.D students are now professors
  - FQ (2006), Tenured professor at **Ohio-State University**
  - \*SL (2008), tenured professor at **University of Wisconsin-Madison**, and later recruited as a chair professors at **Univ of Chicago**
  - \*TL (2009), Tenured professor at **Purdue University**
  - DY (2012), Tenured professor at **University of Toronto**
  - RH (2015), Tenure-track ass. professor at **John Hopkins University**
  - TX (2017), Tenured-track ass. professor at **University of Illinois, Urbana-Champaign (UIUC)**

**\* Women Students**



# My Former Ph.D students Who Joined Industry (big companies)

- ZC (2005), Google Brain, TensorFlow co-author
- QB(2007), Senior manager @Google
- YP(2009), retired from Facebook
- JT(2008), Director@Amazon
- SP(2008), VP@splunk
- XM(2012), Director@Splunk
- JZ (2016), Oracle
- XC (2021), Google Infrastructure
  
- WJ (2008), VP @ 拼多多
- ZY(2011), VP@ 美菜网

# My Former Ph.D students Who Co-Founded Startups

- [Pin Zhou](#), Ph.D 2006, now founding engineer at Datas IO Inc
- [Zhenmin Li](#), Ph.D 2006, co-founder of [PatternInsight](#), now VmWare
- [Qingbo Zhu](#), Ph.D 2007, co-founder of [PatternInsight](#), now CEO and co-founder of More Technology (更多科)
- [Spiros Xanthos](#), Ph.D student, co-founder [PatternInsight](#), now CEO and Co-founder of [EzHome](#)
- [Weiwei Xiong](#), Ph.D May 2013, co-founder of [Whova: Event Mobile App](#)
- [Soyeon Park](#) former post-doc, co-founder of [Whova: Event Mobile App](#)
- [Tianwei Shen](#) former post-doc, co-founder of [Whova: Event Mobile App](#)
- [Jiaqi Zhang](#) Ph.D 2014, Founding engineer of [Whova: Event Mobile App](#)

# Any questions for 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: Thursday 4pm-5:30pm
  - More office hours may be added close to final exams
- *Teaching Assistants*
  - **Tianyi Shan** ([tshan@eng.ucsd.edu](mailto:tshan@eng.ucsd.edu))
  - **Eric Mugnier** ([emugnier@ucsd.edu](mailto:emugnier@ucsd.edu))

## Class Information

- Reading list
  - On Canvas
- Reference Textbook if you need to catch some basic background  
Operating System Concepts, by Silberschatz, Galvin and Gagne. 9th Edition, Wiley & Sons.
- Final: TBA
- Canvas page: <https://canvas.ucsd.edu/courses/29524>
- If you are enrolled, the system should automatically add you into Canvas

# Why CSE221? (1/2)

## 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

# Why CSE221? (2/2)

- 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
  - Quizzes 15%
  - Homeworks 20%
  - Project 30%
  - Final exam 35%
- 
- Letter grades are curved based on distribution



# Read papers in reading list

- How do I make sure that you will read the papers?
  - ~~7 in class pop-quizzes.~~ COVID-19
    - 4 *asynchronous* quizzes on Canvas 😊
    - Each quiz has only 20min, but you can have 24 hours to do the quiz
    - Quiz questions are randomized
    - **Your top 3 quiz scores** are chosen for your final grade (That is, you can skip/fail 1 quiz 😊 )
- Quizzes will count for 15% 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

# How to Read Papers

- You should not read these papers as "truth"
  - You should have your own judgment
- 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 2-3 on the project
  - Start looking for partners now
- Topic: performance evaluation *Topic?*
  - 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

DATE?

# 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 lectures or watch podcast (if you miss lectures)
- 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?

# Warm-up Questions

- 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 2

- 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

*review*

*Are these topics?*

# 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 process 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 Canvas page
- Try the practice quiz (doesn't count toward your final grade)
- Homework 0
- Read 2 papers for next Tuesday's 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!