About me

• PhD in CSE, Penn State University
• Research interest: Systems and Software Security
  – Enjoy hacking kernels, compilers and machine learning
• Rich industry experiences:
  – IBM Watson Research Center
  – NEC Lab America
  – Yahoo
  – Symantec
• Office hours: T and Th 3-4pm at 2243 Storey
• Questions and feedbacks are highly encouraged
Teaching Assistant

• Zhixin Zhang
• Office hours: Monday 1-3pm, 1209 Storey
  – Visits out of the hours require appointments
• Email: zhexin@email.sc.edu
Job hunting

Reference
(Not reference letters)

Online coding (maybe)

Phone interview: mainly coding

Onsite Interviews:
- 3 to 4 Coding interviews
- 1 Behavioral Interview
- 1 System Design interview
Tips for job hunting

• Contact schoolmates and friends to get internal references
• Coding: leetcode, topcoder, careercup
• Stay in the Bay Area during job hunting; you may get onsite interviews directly (without going through online and phone interviews)
• Learn this course well
Course prerequisites

• CSCE 240 (Advanced Programming Techniques)
• CSCE 210 (Computer Hardware Foundations) or CSCE 212 (Computer Architecture)
• C or C++ Programming is a must
Course website

- [https://cse.sc.edu/~zeng1/csce311-f18/](https://cse.sc.edu/~zeng1/csce311-f18/)
- Please check this website frequently for updates of assignments, readings, and slides
- Readings ahead of classes are required
Textbooks

• Required

• Optional
Grading

- Midterm: 20%. Closed-book; covers materials up to the date of the exam
- Final: 25%. Closed-book; covers the whole course
- Lab projects: 45%. Late submission will lead to a grade of 0 for that project
- Quizzes: 10%. Note that quizzes will generally NOT be announced in advance. There is no makeup for missed quizzes and the grade for a missed quiz is 0. The lowest quiz grade will be dropped
Policies

• No cheating
  – Do not share your solutions; do not copy solutions online. These will lead to a course grade of F

• No late submission
  – Late submissions will be rejected
    – Submission via dropbox.cse.sc.edu

• No make-up projects/quizzes/exams
Course goals

• **What** are the subsystems of an OS?
  – The functions and purpose of each subsystem
• **Why** have they been built this way?
  – What are the trade-offs?
  – Can the ideas be applied to your system design?
• **How** to invoke important system services?
  – E.g., how to create threads; how to synchronize processes; how to speed up file read;
Why is an operating system needed? RIPS

• **Resource management**
  – Processor, memory, disk, etc.

• **Inter-process communication (IPC)**
  – One process talks with another

• **Protection**
  – Isolation and access control

• **Services to user programs**
  – Typically, hundreds of system calls
Computer structure
Three major subsystems of an OS

- Process management
  - Processes, threads, synchronization
- Memory management
  - Physical memory, virtual memory
- Device management
  - File systems, networks, display, keyboard, …
Layered design: User program -> libs -> core kernel -> Kernel I/O subsystem -> driver -> controller -> device
What is the difference between a process and a thread?
How do processes share a CPU?
What is a segmentation fault?
How to share memory between processes?
What happens upon a keystroke?
How to optimize your programs?
Summary

• Job hunting tips
• Syllabus
• Computer structure
• OS structure