SSE3044: Operating Systems
Spring 2020

Jinkyu Jeong (jinkyu@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Introduction

• Schedule
  – 10:30 – 11:45 (Tuesday)
  – 09:00 – 10:15 (Thursday)
  – Lecture room: #400118, Semiconductor Bldg.

• Course homepage
  – http://csl.skku.edu/SSE3044S20/
  – Lecture slides, announcements, exam scores, projects, etc.
  – Don’t waste your time in i-Campus

• Q&A site
  – https://piazza.com/sungkyunkwan_university/spring2020/sse3044
  – Access code: sse3044s20
About Me

• Jinkyu Jeong
  – Assistant professor @ SSE and SW Dept.
  – Computer Systems laboratory
  – Research area
    • Operating systems, storage systems, mobile systems, machine virtualization, …
  – Email: jinkyu@skku.edu
  – URL: http://csl.skku.edu/People/Jinkyu
  – Tel: 031-290-7692
  – Office: Semiconductor bldg. #400510 (5th floor)
  – Office hours: Tuesday & Thursday
  – Email contact is preferred
(Awesome) TAs

• Wonsuk Song (송원석)
  – Email: wonsuky@gmail.com
  – Office: #400509 in Semiconductor Bldg.

• Jiwon Woo (우지원)
  – Email: jiwon.woo@csi.skku.edu
  – Office: #400509 in Semiconductor Bldg.
What is OS?

- Computer systems internals
Why Do We Learn OS?

• To graduate?
• To make a better OS or system
  – Functionality
  – Performance/cost
  – Reliability
  – Energy efficiency
• To make a new hardware up and running
• To design OS-aware hardware
• To understand computer systems better
• Just for fun
System Software Track (2019~)

시스템 소프트웨어 트랙 로드맵 2019
Prerequisites

• **Mandatory courses**
  
  • *Introduction to Computer Systems (or System Programming)*
    – SSE2030, CSE2003, or SWE2001
  
  • *System Software Experiment 2 (or System (Unix) Programming)*
    – SSE2033, SWE2007, ICE2015, or CSE3044
  
  • *Computer Architecture*
    – ICE3003, SWE3005, or EEE3050

• **Required skills**
  
  – Fluent C programming skills
  
  – Intel x86 architecture & assembly programming
  
  – Basic knowledge of Unix/Linux systems
  
  – Reading a large, complex program
Textbook

- Operating Systems: Three Easy Pieces
  - Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
  - Arpaci-Dusseau Books
  - September 2015 (ver. 0.91)

- Free Online Book at [http://ostep.org](http://ostep.org)
- Read Remzi's great article at [http://from-a-to-remzi.blogspot.kr/2014/01/the-case-for-free-online-books-fobs.html](http://from-a-to-remzi.blogspot.kr/2014/01/the-case-for-free-online-books-fobs.html)
Why Three Pieces?

"… as Operating Systems are about half as hard as Physics."

A Dialogue on the Book
Chap. 1
Old Textbook

- **Operating System Concepts**
  - 9th Edition
  - Written by A. Silberschatz, P. B. Galvin and G. Gagne
  - Published by Wiley & Sons Inc.
  - 2014
References (1)

• Operating Systems: Internals and Design Principles (8th ed.)
  – William Stallings
  – Prentice Hall, 2014

• Modern Operating Systems (4th ed)
  – Andrew S. Tanenbaum,
  – Prentice Hall, 2014
References (2)

• For Linux:
  – *Understanding the Linux Kernel* (3rd ed.)
  – D. Bovet and M. Cesati,

• For Windows:
  – *Windows Internals* (6th ed.)
  – Mark E. Russinovich, David A. Solomon, and Alex Ionescu,
  – Microsoft Press, 2012

• For Solaris:
  – *Solaris Internals*
  – Richard McDougall and Jim Mauro
  – Sun Microsystems, 2001
Course Plan

• Lectures
  – General operating system concepts
  – Case study: Linux, xv6

• Hands-on projects
  – Using xv6 instructional OS
Lecture Topics

• Virtualization
  – Processes
  – CPU scheduling
  – Virtual memory

• Concurrency
  – Threads
  – Synchronization

• Persistence
  – Storage
  – File systems
xv6 Project

• A teaching OS developed by MIT
  – Port of the Sixth Edition Unix (v6) in ANSI C
  – Runs on multi-core x86 systems

• Why moving on to xv6 (from Pintos)
  – Code inherited from a real, historical OS!
  – Includes working user-level programs and libraries
  – Easier to install on modern Linux systems
  – Easier to extend
  – Easier to understand modern OSes such as Linux
Project Plan (1)

• Initially, the source tree of xv6 has skeleton codes
  – Do nothing but testing the functionality
• You are supposed to fill in the empty code to provide following features
• We are preparing 5~6 projects
• This semester will be tough
  – We are planning new projects (not reusing the projects in the previous semesters)
Project Plan (2)

• Weekly Lab session
  – A separate class with the TA (mandatory)
  – Project announcement
  – Q&A
  – Hints & helps
  – Oral tests
  – Code review
  – …
Project Plan (3)

• Project topics
  – Project 0: booting (1st week, 1 week)
  – Project 1: system call (3rd week, 1.5 weeks)
  – Project 2: CPU scheduling (5th week, 2.5 weeks)
  – Project 3: virtual memory (7th week, 3 weeks)
  – Project 4: page replacement (10th week, 3 weeks)
  – Project 5: file systems (13th week, 2 weeks)

  – Subject to change
Class Policies (I)

• Grading system
  – Class attendance: 10%
  – Exams: 35%
    • Midterm: 15%
    • Final: 20%
  – Projects: 55%
  – Subject to change

• Class attendance policy
  – If you miss any one of the exams, you will fail this course
  – No lateness is allowed
  – Up to four absences will be tolerated
Class Policies (3)

• Cheating policy
  – What is cheating?
    • Copying another student’s solution (or one from the Internet) and submitting it as your own
    • Allowing another student to copy your solution
  – What is NOT cheating?
    • Helping others use systems or tools
    • Helping others with high-level design issues
    • Helping others debug their code
  – Penalty for cheating:
    • Severe penalty on the grade and report to dept. chair
  – Ask helps to your TA if you experience any difficulty
Summary

• You are now taking the most challenging course in the system software track

• This semester will be very tough!

• Keys to success
  – Read textbook exhaustively
  – Think, think, think
  – Begin your project assignments as early as possible
Questions?