SSE3044: Operating Systems

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

• Schedule
  – 10:30 – 11:45 (Mon.)
  – 09:00 – 10:15 (Wed.)
  – Lecture room: #400126, Semiconductor Bldg.

• Course homepage
  – http://csl.skku.edu/SSE3044F16/
  – Lecture slides, announcements, exam scores, projects, etc.
  – Don’t waste your time in i-Campus

• TA
  – 이규선, lgs0409@naver.com
  – #85533 in Corporate Collaboration Center
About Me

• Jinkyu Jeong
  – Assistant professor @ SSE 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. #400626 (6th floor)
  – Office hours: Monday & Wednesday
  – Email contact is preferred
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

<table>
<thead>
<tr>
<th>1학년</th>
<th>2학년</th>
<th>3학년</th>
<th>4학년</th>
</tr>
</thead>
<tbody>
<tr>
<td>1학기</td>
<td>2학기</td>
<td>1학기</td>
<td>2학기</td>
</tr>
<tr>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>성균프레시멘 콘赜</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>정의적 공학설계</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>공학전자</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>프로그래밍 기초설습</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>문제해결방법</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>자료구조 및 알고리즘</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>이산수학</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>시스템SW실습1</td>
<td>랜리학</td>
<td>랜리학</td>
<td>랜리학</td>
</tr>
<tr>
<td>공학수학1</td>
<td>공학수학2</td>
<td>공학수학3</td>
<td>공학수학4</td>
</tr>
<tr>
<td>컴퓨터SW</td>
<td>컴퓨터SW</td>
<td>컴퓨터SW</td>
<td>컴퓨터SW</td>
</tr>
<tr>
<td>SW실습</td>
<td>SW실습</td>
<td>SW실습</td>
<td>SW실습</td>
</tr>
<tr>
<td>SW실습</td>
<td>SW실습</td>
<td>SW실습</td>
<td>SW실습</td>
</tr>
<tr>
<td>디지털 시스템설계</td>
<td>디지털 시스템설계</td>
<td>디지털 시스템설계</td>
<td>디지털 시스템설계</td>
</tr>
<tr>
<td>시스템SW실습2</td>
<td>시스템SW실습2</td>
<td>시스템SW실습2</td>
<td>시스템SW실습2</td>
</tr>
<tr>
<td>마이크로 전자화</td>
<td>마이크로 전자화</td>
<td>마이크로 전자화</td>
<td>마이크로 전자화</td>
</tr>
<tr>
<td>SoC 설계</td>
<td>SoC 설계</td>
<td>SoC 설계</td>
<td>SoC 설계</td>
</tr>
<tr>
<td>컴퓨터 시스템 설계</td>
<td>컴퓨터 시스템 설계</td>
<td>컴퓨터 시스템 설계</td>
<td>컴퓨터 시스템 설계</td>
</tr>
<tr>
<td>컴퓨터 네트워크 설계</td>
<td>컴퓨터 네트워크 설계</td>
<td>컴퓨터 네트워크 설계</td>
<td>컴퓨터 네트워크 설계</td>
</tr>
<tr>
<td>임베디드 시스템 설계</td>
<td>임베디드 시스템 설계</td>
<td>임베디드 시스템 설계</td>
<td>임베디드 시스템 설계</td>
</tr>
<tr>
<td>멀티코어 시스템 설계</td>
<td>멀티코어 시스템 설계</td>
<td>멀티코어 시스템 설계</td>
<td>멀티코어 시스템 설계</td>
</tr>
<tr>
<td>컴퓨터 시스템 설계</td>
<td>컴퓨터 시스템 설계</td>
<td>컴퓨터 시스템 설계</td>
<td>컴퓨터 시스템 설계</td>
</tr>
</tbody>
</table>

SSE3044: Operating Systems, Fall 2016, Jinkyu Jeong (jinkyu@skku.edu)
Prerequisites

• Courses
  – Introduction to Computer Systems
    • SSE2030, CSE2003, or SWE2001
  – System Programming Experiment
    • SSE2033, SWE2007, or CSE3044
  – Computer Architecture
    • ICE3003

• 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

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

• What is Pintos?
  – An instructional operating system based on Nachos
  – Developed by Ben Pfaff @ Stanford University
  – A real, bootable OS for 80x86 architecture
    • Run on a regular IBM-compatible PC or an x86 simulator
  – Written in C with minimal assembly code
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
  – System call & CPU scheduler
  – Virtual memory I (stack)
  – Virtual memory II (copy-on-write)
  – Thread support
  – Synchronization
Project Plan (2)

• Lab session
  – A separate class with the TA (mandatory)
  – Project announcement
  – Q&A
  – Hints & helps
  – Oral tests
  – Code review
## Project Schedule

<table>
<thead>
<tr>
<th>Project</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project 0 (warming-up)</td>
<td>1 week</td>
</tr>
<tr>
<td>Project 1 (syscall &amp; CPU scheduling)</td>
<td>2.5 weeks</td>
</tr>
<tr>
<td>Project 2 (stack growth)</td>
<td>2.5 weeks</td>
</tr>
<tr>
<td>Project 3 (copy-on-write)</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Project 4 (thread support)</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Project 5 (synchronization)</td>
<td>2 weeks</td>
</tr>
</tbody>
</table>
Class Policies (1)

• Grading policy (subject to change)
  – Class attendance: 10%
  – Exams: 35% (mid-term & final)
  – 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
    • There will be a (small) bonus for students who attend all classes
Class Policies (2)

• 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

• This semester will be very tough for you and me
  – Moving to a brand-new textbook and projects

• You are now taking the most challenging course in the software track
  – Lectures will be easy, but projects will not
  – Please make sure if you are ready to take this course

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