SSE3044: Operating Systems

Jin-Soo Kim (jinsookim@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu
Introduction

- **Schedule**
  - 16:30 – 17:45 (Tuesday & Thursday)
  - Lecture room #400112 (Semiconductor Bldg.)

- **Instructor**
  - Jin-Soo Kim (jinsookim@skku.edu)
  - Computer Systems Laboratory (http://csl.skku.edu)
  - Office: Semiconductor Bldg. #400630 (6th floor)
  - Tel: 031-299-4593
  - The best way to contact me is via email.
System Software Track

<table>
<thead>
<tr>
<th>2학년</th>
<th>3학년</th>
<th>4학년</th>
</tr>
</thead>
<tbody>
<tr>
<td>2학기</td>
<td>1학기</td>
<td>2학기</td>
</tr>
</tbody>
</table>

- 자료구조 및 알고리즘
- 컴퓨터시스템 개론
- 소프트웨어 공학
- 컴퓨터구조
- 프로그래밍 언어와컴파일러
- 시스템 운영체제
- 컴퓨터 네트워크
- 마이크로프로세서
- 마이크로프로세서 실험
- 소프트웨어 공학
- 컴퓨터구조
- 프로그래밍 언어와컴파일러
- 시스템 운영체제
- 컴퓨터 네트워크
- 마이크로프로세서
- 마이크로프로세서 실험
- 시스템 운영체제
- 컴퓨터 네트워크
- 마이크로프로세서
- 마이크로프로세서 실험
- 시스템 운영체제
- 컴퓨터 네트워크

3SSE3044: Operating Systems | Fall 2010 | Jin-Soo Kim (jinsookim@skku.edu)
What is OS?

- Computer systems internals

Software

Operating Systems

Application

Architecture

Hardware

CPU

Mem

I/O Devices
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!
Prerequisites

- **Prerequisites**
  - SSE2030 (Introduction to Computer Systems) or CSE2003 (System Programming)
  - ICE3003 (Computer Architecture): Recommended

- **You should be familiar with the followings:**
  - Basic computer organization
  - Process/thread concepts
  - How to write multi-process/multi-threaded programs
  - How to read from/write to files or networks
  - Shells and basic Unix/Linux commands
  - C programming skills
Textbook

- Operating System Concepts
  - Avi Silberschatz, Peter B. Galvin, and Greg Gagne,
    8th Edition,
    John Wiley & Sons, Inc.
    2008.
References (1)

- For General Operating System Concepts:
  - Modern Operating Systems
    (Second Edition)
    Andrew S. Tanenbaum,
References (2)

- **For Linux:**
  - *Understanding the Linux Kernel* (Third Edition)
    D. Bovet and M. Cesati,
References (3)

- **For Windows:**
  - **Windows Internals**
    - (Fifth Edition)
    - Mark E. Russinovich and David A. Solomon,
References (4)

- For Solaris:
  - Solaris Internals
    Richard McDougall and Jim Mauro,
References (5)

- For Introduction to Computer Systems:
  - Computer Systems: A Programmer’s Perspective
    Randal E. Bryant and David R. O’Hallaron,
Course Plan

- **Lectures**
  - General operating system concepts
  - Case studies
    - Linux
    - Microsoft Windows
    - Solaris

- **Hands-on projects**
  - Using Pintos instructional OS

- **Course Homepage**
  - [http://csl.skku.edu/SSE3044F10/Overview](http://csl.skku.edu/SSE3044F10/Overview)
Lectures: Topics

- Operating system structure overview
- Processes and threads
- CPU scheduling
- Synchronization
- Deadlocks
- Memory management
- Virtual memory
- I/O systems
- Storage
- File systems
What is Pintos?

- An instructional operating system
- Developed by Ben Pfaff @ Stanford U.
- A real, bootable OS for 80x86 architecture
  - Run on a regular IBM-compatible PC or an x86 simulator
- The original structure and form was inspired by the Nachos instructional OS from UC Berkeley (Java-based)
- A few of the sources files are derived from code used in the MIT’s advanced operating systems course
- Written in C language (with minimal assembly code)
Project Schedule

- **Project 0**
  - Warming-up project (2 weeks, ~10/1)

- **Project 1**
  - Threads (2 weeks, ~10/15)

- **Project 2**
  - User programs (3 weeks, ~11/12)

- **Project 3**
  - Virtual memory (4 weeks, ~12/10)

- This schedule is subject to change
Project Policies

- **Team project (except Project 0)**
  - Two students in a team

- **Slip days**
  - Each team has 5 “slip” days
  - 20% off per day after slip days exhausted

- **Project grading**
  - Automatic grader in Pintos
  - Demos & oral tests

- **Lab session**
Class Policies (1)

- Grading Policy (subject to change)
  - Exams: 45%
  - Projects: 45%
  - Class attendance: 10%
Class Policies (2)

- **Grading**
  - If you miss one or both of exams, you will fail this course.
  - Do not be late! You should be present when I take class attendance.
  - You have four "tokens"; these tokens can be used for unexcused absences and for excused absences as well.
Academic Integrity

- Cheating
  - What is cheating?
    - Sharing code: either by copying, retyping, looking at, or supplying a copy of a file.
  - 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:
    - Anyone who involved in cheating will fail this course and get disciplinary actions from the University.
  - Ask helps to me or TAs if you experience any difficulty!