SSE2030: Introduction to Computer Systems
Spring 2019

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

• Schedule
  – 13:30 – 14:45 (Monday), 12:00 – 13:15 (Wednesday)
  – Lecture room #400126, Semiconductor Bldg.

• Course homepage
  – http://csl.skku.edu/SSE2030S19/
  – Lecture slides, announcements, programming assignments, exam scores, …
  – Don’t waste your time in i-Campus
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: Monday & Wednesday
  – Email contact is preferred
(Awesome) TAs

• Donghyun Kim (김동현)
  – Email: wadong100@gmail.com
  – Office: #400509, Semiconductor Bldg.

• Jiwon Woo (우지원)
  – Email: jiwon.woo@csi.skku.edu
  – Office: #400509, Semiconductor Bldg.
System Software Track (2019~)
Goal of this Course

How does the computer system work?

or

How does your C program run?
Computer Systems
Computer Systems Internals

Software

System calls

Operating Systems

Application

Architecture

Hardware

CPU

Mem

I/O Devices

SSE2030: Introduction to Computer Systems, Spring 2019, Jinkyu Jeong (jinkyu@skku.edu)
## Levels of Abstraction

<table>
<thead>
<tr>
<th>Application programs</th>
<th>Data structures &amp; algorithms</th>
<th>Programming languages &amp; compilers</th>
<th>Operating system</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microarchitecture</td>
<td>Hardware description languages</td>
<td>Digital logic</td>
<td>VLSI layout</td>
<td>Processing, Fabrication</td>
</tr>
<tr>
<td>Chemistry, Physics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Course Plan

• Program structure and assembly programming
  – Data types: integers, floating points, complex data types
  – Arithmetic and logical operations
  – Control flow

• Running programs
  – Processor architecture
  – Memory hierarchy
  – Linking
  – Operating systems
  – Performance optimizations
Course Components

• Lectures
  – Backgrounds
  – Concepts

• Projects
  – The heart of this course!
  – Provide in-depth understanding of an aspect of systems
  – C/Assembly programing on Linux platform
  – Design, implementation, measurement, optimization
  – Each project must be done individually
Prerequisites

• Courses
  – Basis and Practice in Programming or equivalents
  – Logic circuits: ICE2001
  – Data structures and algorithms or equivalents: SSE2029 or ICE2002

• Required skills
  – C programming
  – Basic knowledge of Unix/Linux systems
Textbook

- Computer Systems: A Programmer’s Perspective
  - Randal E. Bryant and David R. O’Hallaron
  - Pearson Education, Inc.
  - [http://csapp.cs.cmu.edu](http://csapp.cs.cmu.edu)
References: C

• The C Programming Language

  – Brian W. Kernighan and Dennis M. Ritchie (a.k.a K&R book)
  – Prentice-Hall
References: x86 Assembly

• x86-64 Assembly Language Programming with Ubuntu
  – Ed Jorgensen
  – Version 1.0.34
  – March 2016
  – http://www.egr.unlv.edu/~ed/

• The Art of Assembly Language Programming
  – Randall Hyde
  – http://webster.cs.ucr.edu
Reference: x86 Architecture

- Intel Architectures Software Developer’s Manual
  - Volume 1: Basic Architecture
  - Volume 2: Instruction Set Reference
  - Volume 3: System Programming Guide
Grading Policy

• Grading system
  – Class attendance: 10%
  – Exams: 50%
    • Midterm: 20%
    • Final: 30%
  – Projects: 40%
  – Subject to change

• Class attendance policy
  – If you miss any of the exams, you will fail this course
  – No lateness is allowed
  – Up to four absences will be tolerated
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 (F) and report to dept. chair
  – Ask helps to your TA if you experience any difficulty!
Summary

• This course was sometimes known to be tough, but it won’t in this semester 😊

• You will gain systems-level perspective, which is required whether or not you are on the system software track

• SSE2030 serves as a foundation for upcoming courses:
  – System Software Experiment 2 (SSE2033)
  – Computer architecture, Operating systems
  – Programming languages/compilers
  – Microprocessor systems, Embedded systems, …