SWE2001-44: System Programs
Spring 2020

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

• Schedule
  – 16:30 – 17:45 (Tuesday), 15:00 – 16:15 (Thursday)
  – Lecture room #400112, Semiconductor Bldg.

• Course homepage
  – http://csl.skku.edu/SWE2001S20/
  – Lecture slides, announcements, programming assignments, exam scores, …
  – Don’t waste your time in i-Campus

• Q&A website
  – https://piazza.com/sungkyunkwan_university/spring2020/swe200144
  – Access code: swe2001s20
About Me

• Jinkyu Jeong
  – Associate 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: Tuesdays & Thursdays
  – Email contact is preferred
(Awesome) TAs

• Sunghwan Kim (김성환)
  – Email: wadong100@csi.skku.edu
  – Office: #400509, Semiconductor Bldg.

• Jiwon Woo (우지원)
  – Email: jiwon.woo@csi.skku.edu
  – Office: #400509, Semiconductor Bldg.
This course is equivalent to the course “Introduction to Computer Systems (컴시개)”
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
## Levels of Abstraction

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

• Caution
  – This course is identical to ”Introduction to Computer Systems”
  – Those who took ”컴시개” should not take this course
Textbook

• Computer Systems: A Programmer’s Perspective

  – Randal E. Bryant and David R. O’Hallaron
  – Pearson Education, Inc.

  – 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
Class Attendance Policy

• Each of you will have a designated seat (aka 고정 좌석제).
  – Take a seat you like this Wednesday (3/7) and that seat will be yours for the first half of the semester.
  – We will do the same thing after the midterm.
  – Instructor reserves the right to reassign seats as necessary.

• TA will check the attendance by marking empty seats
  – Don’t be late; he may check the attendance at the beginning of the class (or at any random time)
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

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

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