Introduction to System Programming Course

2013 Spring
Euiseong Seo
(euiseong@skku.edu)
Overview

- What this course is about
- Who teaches this course
- Why you have to take this course
- What you will learn in this course
- What you will earn in this course
- How to succeed in this course
What This Course is About

- **System Programming**
  - Information representation
  - Assembly language
  - Processor architecture
  - Compilers, linkers, and loaders
Administrative Information

- **Course Code**
  - SWE 2001

- **Class Hour**
  - Monday and Wednesday
  - 13:30 PM ~ 14:45 PM

- **Lecture Room**
  - #22111 (located on 1F in Engineering Bldg. I)
Textbook

- Computer Systems: A Programmer’s Perspective (2nd Ed.)
  - Randal E. Bryant and David R. O’Hallaron
  - 2011 Prentice-Hall
  - Authors’ site: http://csapp.cs.cmu.edu
Course Components

- Lectures
  - Concepts
  - Backgrounds

- Quizzes
  - On topics covered in previous classes

- Projects
  - Mostly on assembly programming
  - Design, implementation, measurement and optimization
Course Web Page

- Check the web site regularly
- Class material, project information and other useful things will be posted
Grading

- Proportion of Activities
  - Participation 10%
  - Projects 30%
  - Midterm 25%
  - Final 25%
  - Quiz 10%

- If you miss any exam, you will fail
- No lateness is allowed
- Up to four absences will be tolerated
Project

- You will work on each project alone
- The submission status will be posted on the course web page
- Only the assignments submitted before the deadline will receive full credit
- 25% of the credit will be deducted for every single day delay
Ethical Code

- No academic misconduct will be tolerated
  - Zero-tolerance policy
  - One who is found guilty will be kicked out of my class immediately
The Lecturer

- Euiseong Seo
  - Associate professor, Software Dept.
  - E-Mail: euiseong (at) skku.edu
  - Office: #85564
  - Phone: (031) 299-4953
Why You Have to Take This

- **Abstraction is Good, But Don’t Forget Reality**
  - Most CSE courses emphasize abstraction
    - Abstract data types
    - Asymptotic analysis
  - These abstractions have limitation
    - In performance optimization
    - In detection and elimination of bugs
Great Reality

**Ints are not Integers, Floats are not Reals**

- **Example 1:** Is $x^2 \geq 0$?
  - Float’s: Yes!
  - Int’s:
    - $40000 \times 40000 = 1600000000$
    - $50000 \times 50000 = ??$

- **Example 2:** Is $(x + y) + z = x + (y + z)$?
  - Unsigned & signed Int’s: Yes!
  - Float’s:
    - $(1e20 + -1e20) + 3.14 = 3.14$
    - $1e20 + (-1e20 + 3.14) = ??$
Great Reality

- When you measure the required time to execute a part of your code, what can you do?
  - Measuring time in clock cycle accuracy is necessary

- Time Stamp Counter
  - A special 64-bit register in Intel-compatible machines
  - Incremented every clock cycle
  - Read with rdtsc instruction
  - Why couldn’t it be possible with a C-function?
What You Will Learn

- A little bit of computer organization
- A little bit of processor internals
- Assembly language
- Code optimization techniques
- A little bit of system software
  - Compiler, linker, loader, OSs and so on
What You Will Learn

Software

Application

Operating Systems

Architecture

Hardware

CPU

Mem

I/O Devices
Application programs

Data structures & algorithms

Programming languages & compilers

Operating System

Architecture

Microarchitecture

Hardware Description Languages

Digital logic

VLSI layout

Processing, Fabrication

Chemistry, Physics
What You Will Earn

- You will become more effective programmers
  - Able to find and eliminate bugs efficiently
  - Able to understand and tune for program performance

- You will be prepared for later “systems” classes in CSE
  - Compilers, Operating Systems, Computer Architecture, Embedded Systems and etc.
Keys to Success

- Think with your butt, not with your brain
- Invest as many hours as possible
- There’s no royal road to become a hacker