Computer Systems

Introduction

Han, Hwansoo
❖ **Instructor**
  - Hwansoo Han
  - [http://arcs.skku.edu/People/HwansooHan](http://arcs.skku.edu/People/HwansooHan)
  - Corporate Collaboration Center #85568

❖ **TA’s**
  - TBA
  - [http://arcs.skku.edu/](http://arcs.skku.edu/)
  - Corporate Collaboration Center #85565
Lecture and Labs

❖ Lecture will explain high level concepts.

❖ Lab will teach you details of computer systems.
  - Complete 1 assignment within about 2 weeks
  - 3 programming assignments (project/Lab) total
  - In TA sessions, TA will explain the assignment.
Prerequisites

❖ C/C++ programming is a must!
❖ Data structure would help!

❖ What do we prepare by ourselves?
  ▪ How to use Linux commands
  ▪ How to edit source code (vi), compile (gcc/make) and run
  ▪ How to use debugger (gdb)
  ▪ Details in C library functions, network APIs, and thread APIs
Computer Systems Track

**Fundamental**
- Discrete math.
- Data structures
- Digital systems

**Core**
- Compilers & PL
- Computer organization (architecture)
- Operating systems

**Interdisciplinary**
- Parallel computing
- Embedded systems
- Network computing

**System Programming**
= Intro. to Computer System
Evaluation

❖ **Exams (60%)**
  - Midterm 30%
  - Final 30%
  - If you don’t show up, you will automatically fail this course.

❖ **Programming projects (30%)**
  - 3 separate programming assignments
  - Submitting all assignments is mandatory to earn your grade.

❖ **Participation (10%) - Extra**
  - Quiz
  - Attendance
    - 3 days of absence will not be reflected on your grade
    - 4+ days will be counted in evaluation for your grade
Programming assignments

❖ Work in a group
   ▪ Two students in a group work together.

❖ Late policy
   ▪ 10% penalty per day, up to 5 days
   ▪ 50% of the full score is the maximum, if you delay more than 5 days

❖ Cheating
   ▪ Copying, retyping, looking at, or submitting a copy from others and internet
   ▪ Describing verbally one’s code to another
   ▪ Coaching your friend to write a code line by line
What do we learn?

- **Coverage**
  - Binary representation
  - Machine-level programming
  - Computer architecture
  - Memory hierarchy
  - Virtual memory
  - Performance optimization
  - Linking

- **Programming assignments**
  - FP arithmetic (#FP lab)
  - Binary code disassemble (#bomb lab)
  - Buffer overflow attack (#attack lab)
Computer systems:

A Programmer’s Perspective

- Randal Bryant and David O’Hallaron
- http://csapp.cs.cmu.edu

Introduction to computer systems

- A bit more focus on assembly programming
- One chapter will be covered in about 2~3 weeks
- Read the textbook!
References (1)

- **C Programming**
  - *C Programming Language*, (Second Edition)
    B. Kernighan and D. Ritchie,
References (2)

- **Assembly Programming**
  - The Art of Assembly Language Programming,
    (Second Edition)
    Randy Hyde,

- **Intel Architecture Software Developer’s Manual (SDM)**
  Volume 1: Basic Architecture
  Volume 2: Instruction Set Reference
  Volume 3: System Programming Guide
Action items before the next class

- **Linux Programming Environment**
  - Virtual machine (Oracle Virtualbox)
  - Linux (Ubuntu) installation

- Review the C language introductory classes
Class web page

- [http://arcs.skku.edu/](http://arcs.skku.edu/)
  - Courses – System Programming
  - [http://arcs.skku.edu/Courses/ComputerSystems](http://arcs.skku.edu/Courses/ComputerSystems)

- **Download lecture notes**
  - Lecture notes will be posted before the class

- **Project material will be distributed from icampus**

- **Discussion on class materials and programming assignments**
  - Encourage students to discuss each other within a group
  - TAs will respond to the questions