Compiler

- Class web page
  - Lecture slides
    - http://arcs.skku.edu/Courses/Compilers
  - Project handouts, ...
    - http://www.icampus.ac.kr

- Instructor: Hwansoo Han

- TA
  - Grad Students @ARCS (#85565)
Prerequisites

- **Prerequisite Courses**
  - Data Structure
  - System Programming
  - Algorithms
  - Automata Theory
  - Computer Architectures
  - Programming Languages

- **Programming Skill**
  - C/C++
  - Programming in Linux environment (gcc, gdb, vi/emacs, ...)

Textbook & References

- **Textbook**
  - Engineering a Compiler (2\textsuperscript{nd} Ed.)

- **References**
  - Compilers – Principles, Techniques, and Tools (2\textsuperscript{nd} edition)
  - Flex & bison (linux version of Lex & Yacc)
    - Online manual (URLs in class webpage)
Grading

- **Exams**
  - Midterm 30%
  - Final 30%

- **Programming Assignments** 30%
  - Front-End
  - Data Flow Analysis

- **Misc.** 10%
  - Homework & quiz
  - Attendance
Grading Policy

- Must – otherwise you will get ‘F’ grade
  - Take midterm and final exams
  - Submit all two phases of programming assignments
  - Attend 13 weeks and more

- May
  - Be absent from 3 classes (No-Need-To-Ask)
    - From the 4th absence, it will be reflected on your final grade
  - Not submit homework, but it will be reflected on your final grade
  - Not take quizzes, but it will be reflected on your final grade
Topics

- Overview
- Scanning & Parsing
- Context Sensitive Analysis
- Intermediate Representation
- Inner Workings of Compiled Code
- Optimizations
- Data flow Analysis
- Instruction Selection & Scheduling
- Register Allocation
Why Study PLs and Compilers?

- Learn system software intimately interconnected with
  - Architecture, Systems,
  - Programming methodology, and Language design
- Expose to practical algorithmic & engineering issues
  - “Theory and practice are not mutually exclusive; they are intimately connected. They live together and support each other.”
  - [D. E. Knuth, 1989]
What is a Compiler?

- What is a **compiler**?
  - *executable* program $\rightarrow$ *executable* program
    - in another language
    - improve the program *in some way*
Many Forms of Translation

- **Interpretation**

- **Hybrid of compilation and interpretation**
Compiler Technology

- Compiler Technology = Off-Line Processing
  - Goals:
    - Improving performance
    - Making it practical to use the full power of the language
  - Trade-off: preprocessing time vs. execution time (or space)
  - For end-users:
    - Acceptable performance of both compiler and application
To-do-list

- Get your programming project partners
  - Team up with 2 students in a group

- Get programming environment
  - Linux machines
  - Check if “flex” and “bison” installed