

Compiler

– 0. Course Introduction –

Jieung Kim

jieungkim@yonsei.ac.kr



Lecture goal

- Compiler
 - A computer program that translates computer code written in one programming into another language.
 - The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a low-level programming language to create an executable program.



Lecture goal

- Why we learn compiler?
 - Build a large, ambitious software system.
 - Learn how to build programming languages.
 - See theory come to life.
 - Learn how programming languages work.
 - Learn tradeoffs in language design.

Lecture goal

- Learn
 - Compiler overview and concepts
 - Simple programming language concepts
 - Purpose of compilers (and comparison with interpreters)
 - Structures of compilers
 - Applicability of compiler concepts
 - For each module in compilers
 - Functionalities and purpose of modules
 - Underlying theories
 - Implementations

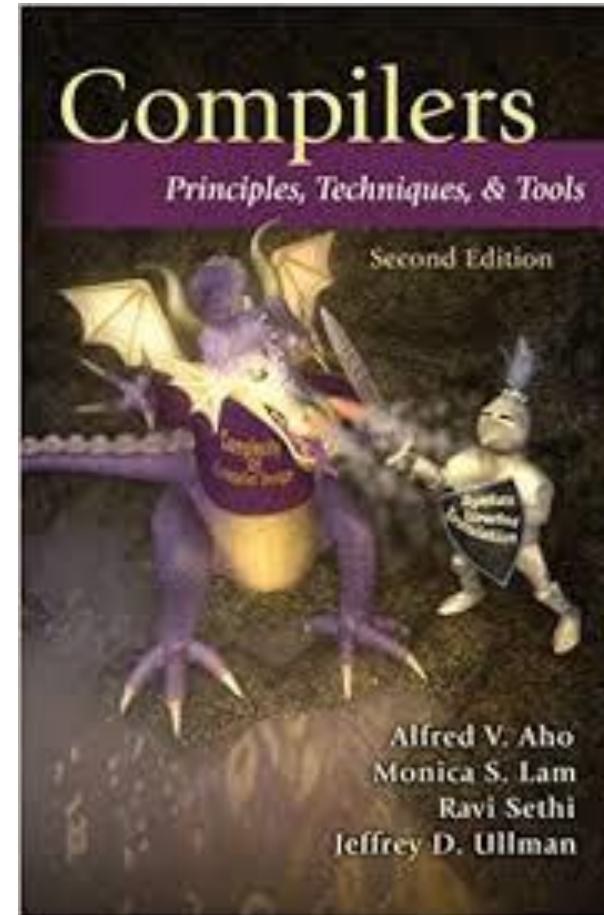


Lecture goal

- Practice
 - Build parts of compilers
 - Exercise theory related problems

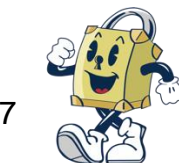
Textbook

- Textbooks
 - Main: “Compilers : *Principles, Techniques, and Tools*” by Aho et. Al.



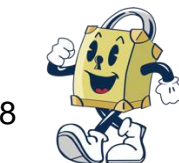
Prerequisites

- Prerequisites
 - Automata theory and formal languages
 - Data structures
 - Programming languages
 - Object-oriented programming
 - Computer programming



Lecture plan

- Lecture notes and programming resources.
 - All lecture notes and additional reading materials will be provided in PDF format.
 - Two types of pages, or entire slides, that will not be discussed during the lecture.
 - Self-study: Those pages, or entire slides, incorporate additional examples or explanations that are beneficial or essential for exam preparation.
 - Supplementary: Those pages, or entire slides, contain additional examples or explanations only for students who hope to study more about those topics.
- Problem sets, quizzes, and exams.
 - 3 problem sets, 2 quizzes, and 2 exams.
 - Problem sets consist of programming questions.
 - Quizzes involve simple pop-up style questions, and they will be conducted via I-Class.
 - Exams consist of all handwritten questions.



Lecture plan

- Providing other useful information.
 - Research topics and areas related to compilers.
 - How do you prepare for tech interviews?
 - How do you enter a graduate school?



Week 1	Intro to the course, compiler concepts	Week 9	Simple LR, more about LR, parser Generator
Week 2	Lexical analysis	Week 10	Symbol table, attribute grammar, syntax directed translation, assignment #2 release
Week 3	Lexical analysis	Week 11	Abstract syntax tree, run-time environment
Week 4	Lexical analysis, role of syntax analysis	Week 12	Intermediate code generation, intermediate code optimization, quiz #2
Week 5	Context free grammar, push down automata, quiz #1, assignment #1 release	Week 13	Intermediate code optimization, Code generation, assignment #3 release
Week 6	Top-down parsing	Week 14	Code generation & optimization
Week 7	Bottom-up parsing	Week 15	Reading period
Week 8	Midterm	Week 16	Final



Lecture plan

- Two lectures are currently scheduled to be canceled due to a holiday and a business trip.
 - 9/18
 - 11/8
- Video lectures will be uploaded as make-up classes.



Grading

- Midterm exam (30%) & Final exam (40%)
 - Offline exams
 - Open book, and using your electronic devices are permitted with no internet connections.
 - Midterm schedule and location: 10/25 Friday 10:00 AM ~ 11:50 AM, D603
 - Final schedule and location: 12/20 Friday 10:00 AM ~ 11:50 AM, D603
- Programming Assignments (3 times, 20% for total)
 - 5% for HW1, 5% for HW2, 10% for HW3
- Quiz (2 times, 10% for total)
 - The exact date will be announced a week ahead.
- Attendance (0%)
 - "F" to the students who commit more than 1/3 times of absence.



Grading

- Grade
 - The course uses absolute grading according to the following grading table.
 - A: 100~80
 - B: 79~60
 - C: 59~50
 - D: 49~30
 - F: 0~29
 - Don't worry, I will calibrate the exam scores if the average is too low.
 - I will adjust the average to 50 if it is lower than 50.
 - If the average is higher than 50, I will use the average score as it is.
 - Scores in other rubrics (assignments, quizzes) will not be calibrated.



Additional rules

- Q/A
 - After class
 - Zoom or offline meeting
 - Send me an email when you hope to arrange a meeting.



Academic integrity

- Cheating and plagiarism are strictly prohibited.
 - Cheated problem sets or exams will be voided.
 - Not OK
 - Copying or otherwise looking at someone else's code.
 - Sharing your code in any way (copy-paste, github, paper and pencil, ...).
 - Using code from previous semesters.
 - Roll calls (대리출석) and sharing online attendance check codes.
 - OK (and encouraged)
 - Discussions of concepts.
 - Discussions of debugging strategies.
 - Verbally sharing experiences.



Questions?

