CSE 304/504: Compiler Design
Spring 2018

Course Description:
Learning compilers will give you an in-depth knowledge about programming languages, sophisticated techniques for handling text descriptions, and the theories behind them. In this class, we will learn how compilers are designed and implemented. Specifically, we will discuss how to write grammars, how to parse and translate them, and the theories behind them. We will also implement a compiler that generates an x86 assembly code. Through the implementation, we will learn how programming language elements are implemented, how to setup the runtime environments, and an assembly language.

Class hours: TuTh, 10:30am ~ 11:50am
Class Room: B204
Office Hours: TBD

Instructor:
YoungMin Kwon (youngmin.kwon at sunykorea dot ac dot kr)
Office: B420
Office hours: TBD

Textbook and References:
- "Compilers Principles, Techniques, and Tools" 2nd edition by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey Ullman
- "lex & yacc" by John R. Levine, Tony Mason, and Doug Brown

Course Website (temporary):
http://www3.cs.stonybrook.edu/~youngkwon/cse304/

Major Topics Covered in the Course:
- Learn overall compiling steps using important tools
- Lexical analysis (Regular expressions, NFA, DFA)
- Syntax analysis (Context-free grammars, Top-Down parsing, Bottom-Up parsing)
- Semantic analysis (Syntax directed translation, Type checking)
- Runtime environment (Memory allocation)
- Code generation (x86 assembly language, runtime environment, register allocation and assignment)

Course Learning Outcomes:

- An ability to use of formal attributed grammars for specifying the syntax and semantics of programming languages.
- Working knowledge of the major phases of compilation, particularly lexical analysis, parsing, semantic analysis, and code generation.
- An ability to design and implement a significant portion of a compiler for a language chosen by the instructor.

Grading:

- Midterm exam: 20 %
- Final exam: 30%
- Programming assignments: 45%
  - Late submission: for each week, 10% of the total score will be deducted from yours.
- Attendance: 5%
  - Missing more than 20% of the class will automatically fail the course.

Course Schedule

Date	Topic
2/27/2018	Overview
3/06/2018	Simple compiler: Syntax definition, Syntax-directed translation
3/08/2018	Lex & Yacc: Simple calculator, Abstract stack machine
3/13/2018	Lex & Yacc: Translation to abstract stack machine (Part 1)
3/15/2018	Lex & Yacc: Translation to abstract stack machine (Part 2)
3/20/2018	Simple compiler: Top-Down Parsing (Predictive Parsing),
3/22/2018	Simple compiler: Lexical analysis, Symbol table
3/27/2018	Simple compiler: Translation of expressions & statements
3/29/2018	Lexical analysis: Regular Expressions, Transition diagrams
4/03/2018	Lexical analysis: NFA, DFA, Conversion from NFA to DFA
4/05/2018	Lexical analysis: Conversion from Regular expressions to NFA
4/10/2018	Syntax analysis: Context-Free Grammars, Writing a grammar
4/12/2018	Syntax analysis: Top-Down Parsing (Nonrecursive Predictive Parsing)
4/17/2018	Syntax analysis: Bottom-Up Parsing (SLR Parser)
4/19/2018	Syntax analysis: Bottom-Up Parsing (LR Parser, LALR Parser)
4/24/2018	Midterm exam
4/26/2018	Syntax directed translation: Overview, S-atrributed definitions
5/01/2018	Top-Down translation of L-attributed definitions
5/03/2018	Bottom-Up translation of Inherited attributes
5/08/2018	Type Checking
5/10/2018	Runtime environments
5/15/2018	Assembly language
5/17/2018	Code generation (Lex & Yacc, implementation)
5/24/2018	Intermediate code generation (Three address code, Part 1)
5/29/2018	Intermediate code generation (Three address code, Part 2)
5/31/2018	Code Generation (Part 1)
6/05/2018	Code Generation (Part 2)
6/07/2018	Code Optimization
6/15 ~ 6/21	Final exam