CS4100/5100: Compiler Design

Class Schedule: TTh, 3:05-4:20pm ENGR 109
Final Exam Schedule: Thur May 14th, 12:40-2:40pm ENGR 109

Instructor: Qing Yi (qyi@uccs.edu)
Associate Professor, Department of Computer Science
University of Colorado at Colorado Springs
Office: ENG 176
Office Hours: TTh: 2-3pm, by appointment
Office Phone: 719-255-3066

Textbook

Engineering a Compiler by Keith Cooper and Linda Torczon, Morgan-Kaufmann, 2011. ISBN: 978-0-12-088478-0

Reference Books

Compilers: Principles, Techniques, and Tools, by Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Addison-Wesley, 1986 (or the newer edition published in 2006).

Overview

This class is a study of the underlying theory and the practice of developing compilers, which play a central role in automatically bridging the gap between high-level programming languages, such as C/C++/Java, and low-level architecture details, such as register allocation and instruction scheduling. Topics to be studied include lexical analysis, top-down and bottom-up parsing, syntax directed translation schemes, intermediate and machine-level code generation, and classic code optimization formulations and algorithms. An emphasis will be put on compiler analysis and optimizations to improve the quality of automatically generated machine code.

Class Objective

Students will learn to apply knowledge from their prior studies of algorithms, data structures, automata theory, and concepts of programming languages to solve concrete problems in building modern compilers. The beginning of the class will focus on reviewing and applying the automata and language theory to automatically generate parsers for key concepts of modern programming languages, while the rest of the class focusing on classic problem formulations, algorithms, and data structures used within compilers to automatically generate binary object code and to improve the quality and performance of the generated code. Students will learn how to apply the acquired knowledge to formulate and solve problems, to implement significant algorithms and data structures via a sequence of programming assignments, and to communicate their solutions through a term project report.

Prerequisites

CS 2160, CS 3160, CS 4700/5700

Grading

30%: projects (programming assignments);
20%: homework (paper) assignments;

5%: class participation and in-class exercises;
45% midterm and final exams;

Requirements Each student will be required to work on regular homework assignments to practice applying the knowledge learned in lectures to solve concrete problems. Additionally, a three-phase project will be assigned where each student will implement the parser, type checking, and code generation components of a compiler that supports a small subset of C. An open-ended term project will also be assigned where the students can propose their own projects and implements their projects by extending their compiler built for the class. The term project can be individual or collaborative, and each team is required to present an initial project proposal within the first month of the semester and a final project report by the end of the semester.

Attendance Students are responsible for the presented materials and assigned readings in class. Attendance will be taken based on submission of quizzes given at each class (5% of overall grade).

Collaboration Policy Unless specified explicitly otherwise for selected programming assignments, students are expected to work on all the other assignments individually, and must indicate in the submission any assistance they have received. For any collaborative project, the project report must indicate the individual contributions of each team member besides the ideas and work generated collaboratively. In any event, each student must be responsible for understanding and being able to explain all statements in the work submitted individually or collaboratively.

Email Policy Students are encouraged to use the class discussion forum from Blackboard to post questions about lectures, homework problems, projects, and course organization. The instructor will monitor the forum and post answers that the entire class can see. Individual emails may be sent to the instructor for personal requests or grading related issues. Always leave a reasonable time period for any question to get answered. Last minute questions (those sent the night before homework is due) may not be answered in time.