Compilers

ENPM617
Fall 2022

Course Overview

The course is intended to cover the underlying techniques of Compiler Construction. The course will introduce the theory and tools that can be employed in order to perform syntax-directed translation of a high-level programming language into an executable code. Topics covered include: lexical analysis; parsing theory; symbol tables; semantic analysis; intermediate representations; runtime environments; code generation; and basic program analysis and optimization. In the optional final project, Students will construct a compiler function for a simple object-oriented language by using LLVM which is a compiler infrastructure, written in C++, and now maintained by Apple Inc..

Learning Outcomes

After successfully completing this course you will be able to:

- Apply programming theory and software development fundamentals to produce computing-based solutions.
- Identify and describe the functions of the major phases of a compiler.
- Identify and evaluate the grammars in the front end of a compiler and techniques for parsing such grammars.
- Define the compiler's role in creating and managing run-time environments.
- Identify and explain the operation of code optimizations.
- Build both the front and back ends of a compiler.

Jerry Wu
jerrywu@umd.edu

Class Meets
Thursday
4:00pm - 6:40pm
JMP 2217

Office Hours
Location TBD
Tu: 3:20-5:00pm; Th: 3:20-5:00pm
and by appointment

Prerequisites
Knowledge of at least one programming language (C or equivalent)

Course Communication
Course communication will be handled through ELMS, including class announcements, homework assignments, grades, etc. Students should make contact through ELMS or by direct email to discuss questions, absences, or accommodations. Extensions may be requested by ELMS message AND assignment comment.

Core Topics: (The is a tentative weekly schedule and subject to change as necessary)

1. Overview of Compilers
2. Lexical analyzer
3. Parser
4. Intermediate code generation
5. Syntax Analysis
6. Top-down and bottom-up Parsing
7. Dataflow Analysis
8. Syntax-Directed Translation
9. Intermediate code generation
10. Type Expression and Register allocation
11. Translation of Boolean Expressions
12. Code Generator
13. Advance Topics

Readings:

Textbook: Compilers Principles, Techniques, & Tools, Alfred Aho, Monica Lam, Ravi Sethi, and Jeffrey Ullman, 2nd Edition.

Project (optional, as a substitute for the final): To be discussed in the class.

Homework and Quizzes: Homework will be bi-weekly or weekly; There may be pop-quizzes.

Prerequisites: Students are required to have at least one programming experience (C or equivalent).

Course Grade: Homework and/or Quizzes (30%; 28/2), Midterm (30%), Final Exam (40%; or a group project). Final grade distribution will be *A*+ = 96-100, *A* = 90-95, *B*+ = 85-89, *B* = 80-85, *C*+ = 75-79, *C* = 70-74, *D*+ = 65-69, *D* = 60-64.

Communication Outside the Classroom: Students may communicate with the Instructor on CANVAS, or by email regarding class cancellation, room change, or other timely announcements. Students may also reserve by appointment, time ahead of class in the Conference Room, or communicate virtually over Zoom to discuss course related content, questions, or concerns. Instructor may hold virtual office hours upon request by students. In-person consultations with the Instructor can be made with prior arrangements.

ELMS Site or Course Webpage: This course will be using the CANVAS course environment. Students can login to their course(s) by going to http://elms.umd.edu. A University online identity and password are required to access CANVAS. Information on your University password is available at http://www.it.umd.edu/password/. CANVAS offers many choices for notification about course activities. It is each student's responsibility to set their communication preferences for their Canvas accounts. Information posted on CANVAS will govern course operation. Lectures, readings, videos, announcements etc. will be put on CANVAS.

No exam make-up will be given except for documented illness
(see Code of Academic Integrity and discuss with Professor).

The University of Maryland, College Park has a nationally recognized Code of Academic Integrity, administered by the Student Honor Council. This Code sets standards for academic integrity at Maryland for all undergraduate and graduate students. As a student you are responsible for upholding these standards for this course. It is very important for you to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. For more information on the Code of Academic Integrity or the Student Honor Council, please visit http://www.studenthonorcouncil.umd.edu/whatis.html .