

CS 671 – Compilers

Spring 2008

Time: Tuesday & Thursday @ 11:00 AM
Location: MEC 215
Instructor: Kim Hazelwood (hazelwood AT virginia.edu)



This course covers the theory and practice of compiler design and implementation. We will cover both the front end (language-dependent) and back-end (machine-dependent) components of modern compilers. We will discuss the impact of language and architectural decisions on compiler design, as well as many of the challenges that remain at the forefront of code generation and optimization research.

Course Topics

- Structure of Compilers
- Lexical Analysis
- Parsers and Abstract Syntax Trees
- Scoping and Symbol Tables
- Type Checking
- Run-time Data Structures
- Intermediate Code
- Instruction Selection
- Data-Flow and Control-Flow Analysis
- Loop Optimizations
- Liveness Analysis and Register Allocation
- Instruction Scheduling
- Interprocedural Analysis and Inlining
- Size and Power Optimizations
- Adaptive Optimization
- Binary Translation

Prerequisites

Strong programming and problem-solving skills will be important to your success in this course.

Grading

- 15% - In-Class Exam 1
- 15% - In-Class Exam 2
- 30% - Homework
- 40% - Term Project

Assignments

Throughout the semester, you will complete 5-6 assignments that consist of written questions and/or implementation tasks. By the end of the semester, you will be familiar with automatic front-end generation tools (flex and bison), a modern compilation infrastructure (Phoenix), and a program analysis tool (Pin). In addition, you will understand the theoretical backing of each phase in a modern compiler.

Exams

We will have non-cumulative in-class exams on February 21 and April 10.

Term Project

In lieu of a final exam, there is a term project. You may work in groups of 1-2. You may implement and demonstrate a significant optimization algorithm, or propose another interesting, relevant project. Project ideas will be posted on the course website. Your project grade will be determined as follows:

- 30% - Project proposal
- 55% - Final conference-quality report
- 15% - Final in-class presentation

Required Text

A. Aho, M. Lam, R. Sethi, and J. Ullman. **Compilers Principles, Techniques, and Tools** (2nd Edition), Addison-Wesley, 2006.

Optional Supplemental Texts

- Andrew Appel. *Modern Compiler Implementation in C*, Cambridge Press, 2004.
- K. Cooper and L. Torczon. *Engineering a Compiler*, Morgan Kaufmann, 2003.
- S. Muchnick. *Advanced Compiler Design and Implementation*, Morgan Kaufmann, 1997.

Late Policy

Late work will result in the following point deduction: $2^{\text{(number of days late)}}$. No exceptions will be made, regardless of the circumstances.

Cooperation

Please feel free to talk with other class members about the algorithms in the programming assignments, but you should **never** even see someone else's code. Any evidence of a violation of this policy will result in an F for the course, and a case with the honor board. It is in your best interest to attend the lectures and to keep pace with the assignments.