

Course Information

Course structure for Spring 2003

The class focuses on the important final stages of the programming language compilation process, known as the compiler *back end*. Starting with an intermediate representation fairly close to the machine, we examine how it can be translated to correct and efficient machine code for a modern architecture, such as PowerPC and IA-64. We also examine the design of a run-time system to support the compiled code by providing memory management services and mediating operating system services.

Prerequisites are a familiarity with at least one functional language (such as ML) and at least one object-oriented language (such as Java), and programming maturity (CS 351, 451). It is helpful, but not mandatory, to have been exposed to programming language semantics (CS 550) and programming language implementation (CS 454, 555). A working knowledge of modern computer architecture is *essential* (CS 341, 441). Instructor permission is needed to register.

Class meeting times

Mondays & Wednesdays 5:30-6:45, or by arrangement, room to be announced.

Instructor

Darko Stefanovic, office FEC 345C, phone 2776561, email darko@cs.unm.edu — office hours to be announced.

Teaching assistant

None.

Assignments and Grading

Active discussion in class following the reading of assigned material. Presentation of assigned material. Optional programming projects. Project topics must be cleared with the instructor. Some suggested topics: generating optimized RISC code from a functional core language, generating optimized RISC code from Java bytecode, generating explicitly parallel machine code (IA-64), analyzing and optimizing RISC code, architectural simulation and performance analysis.

Textbooks

The reading material will come mostly from Steven S. Muchnick: *Advanced Compiler Design and Implementation*, Morgan Kaufmann, 1997, ISBN 1-55860-320-4.