

CS 577 Modern Language Processors Spring 2008

Instructor: Andrew Tolmach
120-23 FAB (503) 725-5492
email: apt@cs.pdx.edu
Office Hours: MW 1-2pm & by appt.
Course web page: <http://www.cs.pdx.edu/~apt/cs577>

Description

This is a graduate-level course on modern techniques for programming language compilation and interpretation. We will focus on the implementation of virtual machines for Java and similar languages. Topics will include: program representation; efficient interpretation; runtime system architecture; just-in-time compilation; garbage collection; multithreading; verification; and feedback-directed optimization.

Prerequisites

An undergraduate compiler course, such as CS321/322, or equivalent; familiarity with an object-oriented programming language such as Java, C++, or C#; good low-level (C or assembler) programming skills.

Readings

There is no textbook. We will read roughly six research papers (obtainable on the web). A number of other reference sources will also be suggested. Copies of lecture slides will be provided by the instructor.

Requirements

The course will be based around a series of roughly eight programming assignments (in C) in which students will implement various features of a simple virtual machine for a Java subset. The homework will be worth roughly 2/3 of the course grade.

There will also be a take-home final exam that focuses on the assigned readings and those portions of the lectures that do are not directly addressed in the homework. The exam will be worth roughly 1/3 of the final grade.

Although it will not be formally assessed, class participation is strongly encouraged, and may affect borderline grades.

Computing Facilities

Some of the homework exercises will require access to SPARC processors, like those in the PSU CS Unix lab. Otherwise, you are free to use any convenient machine that runs Java and C.

Individual Work

All homework assignments and exams must represent your own, individual work. It is permissible to discuss assignments with other students, but the solutions must be recognizably your own. *Do not, under any circumstances, copy another person's program or text and submit it as your own.* Writing code for use by another or using another's code or text in any form (even with their permission) will be considered cheating. Cheating on an assignment or exam will result in an automatic zero grade for that piece of work, and the initiation of disciplinary action at the University level.

Disabilities

If you are a student with a disability in need of academic accommodations, you should register with Disability Services for Students and notify the instructor immediately to arrange for support services.

(Very) Tentative Schedule

<i>date</i>	<i>topics</i>
Apr 1 & 3	Introduction; Java Virtual Machine and bytecode
Apr 8 & 10	Efficient interpretation
Apr 15 & 17	Quick compilation to native code
Apr 22 & 24	Basic optimizations; SSA form; dataflow analysis
Apr 29 & May 1	Liveness; register allocation
May 6 & 8	Compiling object-oriented languages
May 13 & 15	Bytecode verification
May 20 & 22	Garbage collection
May 27 & 29	Multi-threading
June 3 & 5	Summary
June 12	Take-home Final Exam Due 5pm