CS 510 Advanced Programming Language Implementation Spring 2014

Instructor: Andrew Tolmach
120-23 FAB  (503) 725-5492
e-mail: apt@cs.pdx.edu
Office Hours: after class, W 4-5pm, or by appt.
Course web page: http://www.cs.pdx.edu/~apt/cs510apli

Description

This is an advanced course on modern techniques for programming language compilation and interpretation. We will focus on the implementation of higher-order, safe languages such as Java and JavaScript.

Topics will include: language implementation architectures; compilation challenges posed by modern languages and hardware; efficient interpretation; dataflow analysis and classical optimizations; static single assignment form and other program representations; dynamic language compilation; garbage collection; and compiler correctness and verification.

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 two research papers per week. Additional background reading will also be suggested for those who need it. Copies of lecture notes (of varying degrees of formality) will be provided by the instructor.

Requirements

Each student must do a substantial programming project or write a substantial survey paper on a topic relevant to the course. Students choose their project/topic, with agreement of the instructor. It is permissible to work in teams of two on a project (but not on a paper), with a corresponding increase in expectations. The project or paper will count as roughly 2/3 of the course grade.

In addition, there will be weekly homework assignments, to encourage you to keep up with the reading material. Most of these will require only brief written answers. In addition, there will be two or three more substantial assignments involving programming problems. All these are to be submitted by email. Together, these assignments will count as roughly 1/3 of the course grade.

There will be no exams.

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

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

The idea behind this schedule is that we will address each topic over two course meetings: a Wednesday and the following Monday. Wednesday classes will focus on basic background; Monday classes will explore more advanced issues raised by the readings. Homework on the topic will be due on the Monday.

date   topics

Mar 31  Introduction
Apr 2 & Apr 7  Compiler and Interpreter Architectures; Java Virtual Machine
Apr 9 & 14  Compilation challenges for Modern Languages
Apr 16 & 21  Compilation challenges for Modern Hardware
Apr 23 & 28  Efficient Interpretation
Apr 30 & May 5  Basic Compiler Optimizations and SSA; Dataflow analysis
May 7 & 12  Functional IR representations; CPS; Memory Optimizations
May 14 & 19  Garbage Collection
May 21  Dynamic Languages
May 26  Memorial Day – No class
May 28 & June 2  Correctness and Verification
June 4  No class
Wed. June 11  Final exam slot (12:30-2:20pm): Project Presentations

date   project/paper milestone

Apr 25  Draft proposals due
May 2  Final project approval required
May 23  Progress report due
June 11  Project or paper due