

Compiler Construction 2011

Introduction

In this course you will learn about compiler construction: the main phases in a compiler like scanning, parsing, static-semantic analysis, and code generation; tools like parser generators and static aspect-oriented programming; formalisms like context-free grammars; program representations like abstract syntax trees and intermediate code; and run-time systems that handle programs during execution. In the course project you will use Java to implement a compiler that compiles your own small procedural language into Intel assembly code. The techniques you will learn are general and useful for a variety of applications, not just compilers.

Prerequisites

The course requires knowledge of algorithms and data structures, good knowledge of Java and object-oriented programming. Formal requirements:

EDAA01 (Programming, Second course).

Teachers

Lennart Andersson, office E:2189, 2nd floor, north stairwell. Telephone: 22 23045. E-mail: Lennart.Andersson@cs.

Emma Söderberg, office E:4132b, 4th floor, south stairwell. Telephone: 22 23075. E-mail: Emma.Soderberg@cs.

Secretary

Lena Ohlsson, office E:2179, 2nd floor, south stairwell. Office hours 9:30-11:30, 12:45-13:30. Telephone 22 28040.

Information

The web pages, cs.lth.se/EDA180, will be updated during the course.

Examination results will be sent to the participants by email.

Leftover material distributed at the lectures will be placed on the shelf outside room E:2180.

Literature

The following text book will be used:

A. W. Appel with J. Palsberg: Modern Compiler Implementation in Java, 2nd Edition, Cambridge University Press, 2002, ISBN: 0-521-82060-X

The following chapters are part of the course: 1-8, 10-11, 13-14. More detailed reading instructions will be given during the course. The textbook is also available as an e-book at Elin, <http://elin.lub.lu.se/elin>, using your STIL identity. It can be read on-line or downloaded to be used for one or three days.

In addition to the textbook, the course will use additional material such as lecture slides, programming assignments, seminar exercises.

Course contents

During study period vt1:

- 14 lectures, 2 hours each
- 6 programming assignments (compulsory)
- 5 seminar exercises, 2 hours each
- exam, 5 hours (compulsory)

During study period vt2:

- project (compulsory)

Lectures

Lectures are held during study period vt1 on Tuesdays, 13-15 and Thursdays, 10-12. The first lecture is held on Tuesday, January 18.

Preliminary schedule:

- F1** Introduction
- F2** Scanning and regular expressions
- F3** Context-free grammars and parsing.
- F4** LL parsing.
- F5** More LL parsing.
- F6** Abstract syntax trees
- F7** Modularization of compilers.
The visitor pattern.
Static aspect-oriented programming.
- F8** Static-semantic analysis

- F9** Run-time systems.
- F10** Intermediate code generation
- F11** LR parsing.
- F12** Short introduction to optimizing compilers.
(Guest lecturer: Jonas Skeppstedt)
Attribute Grammars.
(Guest lecturer: Emma Nilsson-Nyman)
- F13** Automatic memory management.
(Guest lecturer: Roger Henriksson)
- F14** Assembly code generation.
About the project. About the exam.

Programming assignments

The programming assignments are compulsory and done in groups of two students. You have to do some preparations and you may complete the assignment before the laboratory session.

There are laboratory session each Friday during weeks 2-7 of study period vt1. The first session is on Friday, January 28. There are two alternative session slots:

- Friday 8-10, Gamma
- Friday 13-15, Mars

Sign up for a session at <https://sam.cs.lth.se/LabsSelectSession?occasionId=123>.

You must complete the assignments in order to be allowed to take the written exam and start the project.

Seminars

Seminars are held on Fridays, 10-12. The seminars take place in E:2116. The first seminar is on January 21.

During the seminars problems will be solved that illustrate concepts covered in the lectures. You should try to solve the problems yourself before each seminar. Active participation in the seminars will give you extra points at the written exam the same year (see the course web pages for more details).

Project

The project is performed in groups of two students during study period vt2. There will be several deadlines for partial submissions. Details will be available at the end of study period vt1 at the course website. There is a final deadline, Friday June 3, 2011, when all parts of the project must have been checked and approved by your project supervisor.

Examination

There is a written exam given during the exam period after study period vt1 (i.e., before the project). To be allowed to write the exam you need to have completed all the programming assignments.

The regular examination is held on March 11. The next examination opportunity is April 29 (1 week advance registration is required).

To obtain a final grade you must have passed the written examination and completed the programming assignments and the project. The grade is based on the written exam and seminar activity.