ID2202 Compilers and Execution Environments 7.5 credits

Kompilatorer och exekveringsmiljöer

This is a translation of the Swedish, legally binding, course syllabus.

Establishment

Course syllabus for ID2202 valid from Spring 2019

Grading scale

A, B, C, D, E, FX, F

Education cycle

Second cycle

Main field of study

Computer Science and Engineering

Specific prerequisites

120 university credits (hp) in engineering or natural sciences and documented proficiency in English corresponding to English A and courses in basic computer organization/architecture, Programming in C or Java, algorithms and data structures. Basic knowledge of C-programming.

Language of instruction

The language of instruction is specified in the course offering information in the course catalogue.

Intended learning outcomes

The course covers techniques for the implementation of programming languages using compilers in both real and virtual execution environments.

The overall aim of the course is to provide an understanding of how a programming language is implemented including common theories and how these theories are applied. The course will cover techniques for reading, understanding, translating, improving, and executing programs.

This understanding means that after the course a student should be able to:

explain the overall structure of a compiler.

describe regular expressions and finite automata and use them unambiguously for lexical analysis;

combine and apply techniques to construct a deterministic finite automaton from regular expressions.

describe context free grammars and apply them to capture common programming language constructs; describe the basic approach of top-down and bottom-up parsing;

construct recursive descent, LL(1), LR(0), and SLR parsers;

identify whether grammars can be used with a given parsing technique.

list and describe tasks carried out during semantic analysis.

describe the design principles behind intermediate program representations;

explain and apply methods for selecting instructions;

describe the organization of activation records and identify the impact of different design decisions for activation records on program execution.

define the liveness of variables and compute liveness information from control flow graphs;

define the principles of conservative approximations for analyzing programs;

construct interference graphs and perform complete register allocation on them reflecting the design of activation records.

name key components in an execution environment;

describe simple techniques for heap management and garbage collection;

identify the amortized costs of different approaches to garbage collection;

list common techniques and explain their properties used in virtual execution environments.

describe common techniques for program optimization;

identify loops in programs using dominators;

describe and apply techniques for optimizing loops and memory access.

describe how characteristics of hardware architectures influence compilation of programs;

give examples of important characteristics.

Course contents

- Reading programs: lexical and syntactic analysis. Finite automata, regular expressions, LL- and LR-parsing.
- Understanding programs: Semantic analysis, type checking. Scope control, declarations and expressions.
- Translating programs: machines and instructions. Intermediate code, instruction selection, stacks and procedure calls.
- Executing programs: virtual execution environments and run-time systems. Memory management, garbage collection, loading and linking, just-in-time compilation.
- Dependencies on hardware architecture.
- Improving programs: optimization. Machine independent optimizations (dataflow analysis, strength reduction, ...)
- Machine level optimization (register allocation, scheduling, prefetching, power consumption...)

Course literature

Modern Compiler Implementation in Java, Andrew W. Appel

Upplaga: Second Förlag: Cambridge University Press År: 2002

ISBN:

Examination

- INL1 - Assignments, 1.5 credits, grading scale: P, F
- TEN1 - Examination, 6.0 credits, grading scale: A, B, C, D, E, FX, F

Based on recommendation from KTH's coordinator for disabilities, the examiner will decide how to adapt an examination for students with documented disability.

The examiner may apply another examination format when re-examining individual students.

If the course is discontinued, students may request to be examined during the following two academic years.

Other requirements for final grade

Approved written exam (TEN1; 6hp) and approved home assignments (INL1; 1.5hp).

The home assignments are evaluated with the grades P/F (pass or fail). The course features three assignments which must be solved and submitted in time (you will have one week to solve all tasks on
an assignment). Each assignment will feature 6 points on tasks. In order to pass the assignment part of the course you have to reach 9 points on all three assignments.
If you submit an assignment in time, the points will serve as bonus points on the exam. That means that you can score at most 18 bonus points for the exam. Note: the bonus points are valid only for this academic year.

The tasks of the exam are worth 200 points.

The grades for the entire course are defined by total points being the sum of the exam points and the bonus points you got on the assignments. You need at least 100 total points to pass the exam. The written exam is evaluated with the grades A-F.

The grades for the number of total points n are as follows:

n >= 180: A
180 > n >= 160: B
160 > n >= 140: C
140 > n >= 120: D
120 > n >= 100: E
100 > n >= 80: Fx
80 > n : F

In case of the grade Fx, completing examination is possible within one month after the original exam. In that case, the course responsible will on demand offer an extra home assignment to be solved by the student within one week.

Ethical approach

- All members of a group are responsible for the group's work.
- In any assessment, every student shall honestly disclose any help received and sources used.
- In an oral assessment, every student shall be able to present and answer questions about the entire assignment and solution.