Compiler Construction

Programme course

6 credits

Kompilatorkonstruktion

TDDB44

Valid from:

Determined by
Board of Studies for Computer Science and Media Technology

Date determined
2017-01-25

Offered for the last time
Autumn semester 2023

Replaced by
TDDE66

Main field of study

Information Technology, Computer Science and Engineering, Computer Science

Course level

Second cycle

Advancement level

A1X

Course offered for

- Computer Science, Master's Programme
- Computer Science and Engineering, M Sc in Engineering
- Industrial Engineering and Management - International, M Sc in Engineering
- Industrial Engineering and Management, M Sc in Engineering
- Information Technology, M Sc in Engineering
- Computer Science and Software Engineering, M Sc in Engineering
- Computer Science, Master's programme

Specific information

Overlapping course contents: TDDD55

Entry requirements

Note: Admission requirements for non-programme students usually also include admission requirements for the programme and threshold requirements for progression within the programme, or corresponding.

Prerequisites

Data Structures and Algorithms. Formal Languages and Automata Theory, either via that course or studying selected material (check with lecturer). Some knowledge of C++.

Intended learning outcomes

The aim of this course is to give a comprehensive introduction to the theoretical and practical issues underlying the design and implementation of compilers. After the completion of the course you should be able to:

- explain and apply fundamental principles and techniques of compiler design
- explain and use methods for lexical analysis, top-down and bottom-up parsing
- explain and use methods for semantic analysis, syntax-directed translation, code optimization
- explain methods and theory for construction of lexer and parser generators, code generator generators
- explain methods for generation and optimization of code for RISC-based architectures
- construct and implement a top-down parser and a bottom-up parser for a given context-free grammar
- use lexer and parser generators to build a lexical analyzer and a parser for a real compiler
- explain and use methods for machine code generation
- design and implement a complete compiler including: lexer, parser, memory management, semantic analysis, optimization, and machine code generation

Course content

Different types of translators such as compilers and preprocessors. Methods for lexical analysis and syntax analysis. Management of declarations. Different types of internal representations. Memory management and run-time organization. Code generation and code optimization, especially with regards to RISC processors. Methods for handling errors. Compiler construction and compiler generation tools. Language design. A complete compiler for a Pascal-like language is constructed during the laboratory sessions. Certain modules of this compiler will be automatically generated using compiler generation tools, whereas other parts will be implemented by hand in C++.

Teaching and working methods

The theory is presented in the lectures. The seminars prepare for the laboratory assignments, where a complete compiler for a small Pascal-like language is implemented.

Examination

LAB1	Laboratory Work	3 credits	U, G
TEN1	Written examination	3 credits	U, 3, 4, 5
UPG1		0 credits	U, G

The questions in the written exam check how well the student has fulfilled the learning goals of the course. For passing the exam, deficits in fulfilling certain partial goals can be balanced by a better fulfilling of other partial goals.

Grades

Alternative-grade scale, LiU, U, 3, 4, 5

Course literature

Aho, Lam, Sethi, Ullman: Compilers Principles, techniques, and tools, Second edition, Addison-Wesley, 2006.

Department

Institutionen för datavetenskap

Director of Studies or equivalent

Ahmed Rezine

Examiner

Peter Fritzon

Course website and other links

http://www.ida.liu.se/~TDDB44/

Education components

Preliminary scheduled hours: 48 h
Recommended self-study hours: 112 h

Course literature

Aho, Lam, Sethi, Ullman: Compilers Principles, techniques, and tools, Second edition, Addison-Wesley, 2006.
Kompendier, utges av institutionen för datavetenskap.

Common rules

Regulations (apply to LiU in its entirety)

The university is a government agency whose operations are regulated by legislation and ordinances, which include the Higher Education Act and the Higher Education Ordinance. In addition to legislation and ordinances, operations are subject to several policy documents. The Linköping University rule book collects currently valid decisions of a regulatory nature taken by the university board, the vice-chancellor and faculty/department boards.

LiU's rule book for education at first-cycle and second-cycle levels is available at http://styrdokument.liu.se/Regelsamling/Innehall/Utbildning_pa_grund-_och_avancerad_niva.