Course Memo

Introduction

Welcome to the course in Compilers and Execution Environments!

In this course, you will learn how a compiler works. You will learn about compiler theory, for instance regular expressions, parsing theory, optimization algorithms, code generation strategies, garbage collection, and more. Moreover, you will gain significant practical experience in designing and implementing a complete compiler from scratch.

Learning Outcomes

The complete syllabus **can be found here (https://www.kth.se/student/kurser/kurs/ID2202?l=en)**. In summary, the learning outcomes are as follows.

After passing the course, the student should be able to

- use methods for lexical, syntactic and semantic analysis
- use methods for generation of machine code
- use methods for optimizing programs
- give an account of common components in execution environments

in order to

- obtain an understanding of how a programming language is implemented as well as for the general theories that are used and how these can be applied.

Modules

The course is divided into three modules. Each module is two weeks long and covers a specific area:

- **Module 1 (weeks 44 and 45): Lexical analysis and parsing.**
  The goal of module 1 is to comprehend, design, and implement the front end of a compiler, including lexing, parsing, and interpretation.
- **Module 2 (weeks 46 and 47): Code generation and runtime environments.**
  The goal of module 2 is to compile complete unoptimized Cigrid programs to x86 machine code.
- **Module 3 (weeks 48 and 49): Program analysis and optimizations.**
  The goal of module 3 is to perform various optimizations of your compiler, including intermediate code optimization and register allocation.

Learning Activities and Examination

Each module consists of the following *learning activities* (marked with LA) and formative *examination tasks* (marked with ET):

1. **Lectures (LA).** Each module starts with three lectures (Mondays and Tuesdays the first week in a module). All lectures will be given physically at KTH Kista, but they will also be live streamed and recorded using Zoom.
2. **Hacking sessions (LA).** Each week, there are two online hacking sessions (please see the schedule). During these online sessions, you will be able to chat with fellow students, and to pose questions to teaching assistants.
3. **Assignments (LA/ET)**. During the 2 weeks module, you will work independently (not in groups) on different design and implementation tasks. You will submit your tasks continuously to the course Git system. The deadline for submitting all tasks is on Friday afternoon 5pm in the last week of the module.
4. **Peer reviewing (LA/ET).** During the beginning of the week after a module, you will be given the task to peer review the code and the solutions of another fellow student. As part of this peer reviewing, you provide comments and written questions about the other student's work.
5. **Seminars (LA/ET)**. The week after each module, after the peer reviewing task, you will participate in a seminar, where you will present your tasks and answer questions from a teaching assistant and the peer reviewing student. There will be three students presenting during each seminar. The seminars will take place over Zoom and there will be individual bookings of these time slots.

The last week of the course (week 50) includes peer reviewing and seminars for module 3, as well as a final summary lecture. During this week, there will also be extra retake opportunities for the previous modules.

Grading

According to the syllabus, there is one course part that is reported into LADOK.

- DAT1 - Computer based exam, 7.5 credits, Grading scale: A, B, C, D, E, FX, F

Note, however, because of the Covid-19 situation, KTH has decided that examination should not (unless motivated by pedagogical reasons) take place as a written exam. For this reason, all examination will be done using: (i) the assignments (ii) peer reviewing, and (iii) seminars. Hence, there is no written exam at the end of the course.

Each of the three modules are graded with one of the following grades:

- *Satisfactory (S): The student has fulfilled the following for the module:*
  - *All assignments marked as (S) have been submitted to the Git correction system and all automatic tests pass.*
  - *A teaching assistant has corrected tasks for (S) that cannot be automatically corrected and the results are pass.*

- *The student has sent in a peer reviewing assessment that has been given the grade pass by a teaching assistant.*
- *The student has participated in a complete seminar, where they have acted as an opponent and asked questions based on the peer review.*
- *The student has presented his/her solution on a seminar, defended the implementation, and explained the solution in a satisfactory manner. The teaching assistant for the seminar has given the student pass on the oral presentation.*

- *Good (G): The student has fulfilled all criteria for (S) as well as submitted solutions to all assignments marked as (G). For all the tasks at level (G), the student has received a pass by either the automatic grading system or manually by a teaching assistant, or both.*
- *Very Good (VG): The student fulfilled all criteria for (S) and (G), as well as submitted solutions to all assignments marked as (VG). For all the tasks at level (VG), the student has received a pass by either the automatic grading system or manually by a teaching assistant, or both.*
- *Failed (F): The criteria for satisfactory (S) have not been fulfilled.*

The final grade (A, B, C, D, E, FX, F) is based on the overall grades of the three modules.

If a student submits solutions that are given pass on time *for all three modules* (meets the deadline of the last Friday at 5pm in each module), then the student receives a *deadline bonus.* The final grade is computed as follows:

- **Grade A** if received VG on all three modules. With a deadline bonus, only two VGs and one G is needed.
- **Grade B** if received two VGs and one G. With a deadline bonus, only one VG and two Gs are needed.
- **Grade C** if received one VG and two G. With a deadline bonus, only three Gs are needed.
- **Grade D** if received at least two Gs and one S. With a deadline bonus, only at least one G and two Ss are needed.
- **Grade E** if received at least three S.
- **Grade FX** if F is received on at most one module.
- **Grade F** otherwise.

During the last week of the course (week 50), students have the possibility to complement their solutions for all three modules, and to receive higher grades on the different modules. A submission must be submitted at latest by December 10, 5pm. Peer reviewing and seminars will take place the week after (week 51). Note that this submission is optional. If complementary work is done during week 50, the deadline bonus is not valid anymore.

Example 1: Suppose a student submitted all assignments on time, and passed seminars and peer reviewing on all modules. Suppose also that they received grades G S G on the three modules. Hence, because of the deadline bonus, the student gets the final grade D.

Example 2: Suppose a student submitted everything on time and hence has the deadline bonus. The grades for the three modules were VG G S. Hence, they get the final grade D. However, the student is not satisfied with this grade and manages to upgrade the S to a G during week 50, resulting in VG G G. The deadline bonus is not valid anymore (after complementary work). As a consequence, the student gets grade C. To get a B during the complementary work, one of the Gs would have to be further upgraded to a VG (i.e., resulting in VG G VG or VG VG G). Note that if the student would have received VG G G on time and received the deadline bonus, the final grade would have been a B.

The grade will be assigned according to what has been reported into Canvas by the end of December 17. If a student receives grade FX, they will have to submit a new solution at latest by December 31. The peer reviewing and the seminar for the FX submission will take place in January on an individual basis.

There will be one retake occasion during the spring. It is then possible to complement your solutions and assignments for all three modules, to achieve a higher grade or to pass the course. The submission deadline is March 1. The peer reviewing and seminars will take place during March or April, and will be assigned on an individual basis. If you have not passed the course after the March deadline, you will have to retake it the next time the course is given.

Note also that the teacher may ask for individual examination and presentation of exercises, if there are unclarities in submitted solutions, unclarities during seminar presentations, or indications of cheating attempts.

Teachers

Examiner and Course Responsible

- **David Broman (mailto:dbro@kth.se)**

Lectures

- **David Broman (mailto:dbro@kth.se)**

Hacking Sessions and Seminars

- **Linnea Ingmar (mailto:lingmar@kth.se)**
- **Viktor Palmkvist (mailto:vipa@kth.se)**
- **Lars Hummelgren (mailto:larshum@kth.se)**

Literature

The following is the recommended course book:

- Keith D. Cooper & Linda Torczon. **Engineering a Compiler.** Second Edition, Morgan Kaufmann, 2012.

The book is available for free online, using your KTH account. **See this link (https://www-sciencedirect-com.focus.lib.kth.se/book/9780120884780/engineering-a-compiler)** .

Disability

If you have a disability, you may receive **support from Funka (https://www.kth.se/en/student/studentliv/funktionsnedsattning)** .

Funka also recommends that you inform the teacher (in this case **David Broman (mailto:dbro@kth.se)**) regarding any needs you may have. Funka does not automatically inform teachers in courses.

Code of Honor

Please make sure that you are aware of and that you understand the **KTH EECS Code of Honor. (https://www.kth.se/en/eecs/utbildning/hederskodex)**

Policy against Plagiarism

Note that all forms of cheating and plagiarism will be reported. Please see KTH's **policy for handling plagiarism (https://canvas.kth.se/courses/17831/files/2446097/download)** .

- **Programming assignments.** You may discuss problems and solutions with anyone. You are not allowed to copy ANY code and submit it as part of your solution. You must create your own solutions from scratch.
- **Individual theory assignments.** You are not allowed to discuss or publish any of the assignments. You are not allowed to get help from anyone or find the solution on Internet.
- **Written text assignments.** You are not allowed to copy, cut, or paste any text into your report that is not produced by you. The only exception is if you quote text properly, and give a citation to the original source.