Compilerbau

Winter

(engl. Compiler Construction)

Modulnummer: FIN-INF-100382

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
CB	5	Winter	starting at 3.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	
Verantwortung:	Prof. Dr. Michael Kuhn
Dozent:in:	Prof. Dr. Michael Kuhn
Lehrveranstaltungen:	• Lecture *Compilerbau* • Exercise *Compilerbau*
Verwendbarkeit:	- B.Sc. INF: Informatik - Wahlpflicht - B.Sc. INF: Informatik Wahlpflicht (SPO 2027) - B.Sc. CV: Informatik - Wahlpflicht - B.Sc. CV: Computervisualistik Wahlpflicht (SPO 2027) - B.Sc. INGINF: Informatik - Wahlpflicht - B.Sc. INGINF: Informatik Wahlpflicht (SPO 2027) - B.Sc. WIF: Gestalten und Anwenden - Wahlpflicht - B.Sc. WIF: Informatik Wahlpflicht (SPO 2027) - B.Sc. INF (bilingual): Informatik - Wahlpflicht - B.Sc. INF (bilingual): Informatik Wahlpflicht (SPO 2027)

Angestrebte Lernergebnisse:
The participants...

- understand the theoretical foundations of formal languages
- apply algorithms for optimizing regular expressions, grammars and code representations
- understand important concepts of compilers
- create components of a compiler in small teams
- present solution approaches and results to other participants

Inhalt:
Compilers translate programs from a source language into a target language, usually from a higher programming language into a machine language. Among other things, this allows the same application to be translated for and executed on different processor architectures with relatively little effort. A deeper understanding of compilers helps to write more efficient and secure programs. The programs to be translated are also checked and improved by compilers. Number of errors found depends on the strictness of the language. Some languages, such as Rust, make it more difficult to write programs without errors, but offer certain guarantees of correctness. In addition, compilers perform a variety of optimizations in order to use the given computer architecture as efficiently as possible. The lecture is based on the flipped classroom principle. The relevant chapters of the book “Introduction to Compilers and Language Design” will be read independently and then discussed in depth during the lecture. The concepts are taught in the lecture, while the exercises are used for practical application and implementation of the acquired knowledge.

Arbeitsaufwand:

- 56h attendance
- 94h independent work

Prüfungsvorleistungen:	**Studien-/Prüfungsleistungen:**	**Lehrform / SWS:**
• Active and successful participation in the exercises	Oral exam	• Lecture (2 SWS) • Exercise (2 SWS)

Voraussetzungen nach Prüfungsordnung:	**Empfohlene Voraussetzungen:**
none	• Practical knowledge of a system programming language such as C or C++ • Basic knowledge of data structures and computer architectures (e.g. algorithms and data structures, computer engineering)

Medienformen:	**Literatur:**

	• Introduction to Compilers and Language Design (Douglas Thain)

Hinweise:

Revision #1
Created 2026-10-31 09:06:37 UTC by Fakultät für Informatik
Updated 2026-10-31 09:06:37 UTC by Fakultät für Informatik