

2.8 Compiler

Informationen über das Modul

Modulbezeichnung	Compiler
Titel in Englisch	Compiler
Prüfungsnummer	IN 3970320, 2970776 TI 3976573, 2976515 WI 3975696 IIS 9775107
Modulkürzel	COM4.WP
Modulverantwortlicher	Prof. Dr. Jürgen Scholz
Fakultät	Fakultät für Informatik
Modulart	Wahlpflichtmodul
Dauer des Moduls / Häufigkeit des Angebots des Moduls	Das Modul wird unregelmäßig bzw. auf Nachfrage angeboten.
Lehrveranstaltungen in den Modulen	Compiler (3 SWS) zugehöriges Praktikum (1 SWS)
Lehrsprache	Das Modul wird in deutscher Sprache unterrichtet.
Lehr- und Lernmethoden des Moduls	Seminaristischer Unterricht und begleitendes Praktikum zur Anwendung und Vertiefung der erworbenen Kenntnisse. Zusätzlich unterstützt und fördert das Praktikum die Gruppenarbeit und das Selbststudium.
Voraussetzungen für die Teilnahme am Modul	Solide Kenntnisse einer höheren Programmiersprache wie JAVA oder C / C++
Verwendbarkeit des Moduls innerhalb des eigenen, sowie für andere Studiengänge	Wahlpflichtfach für Bachelorstudiengänge
Gesamtarbeitsaufwand und seine Zusammensetzung	SWS: 4, CPs: 5, Präsenzzeit: 60 h, Selbststudium: 90 h, Gesamtaufwand: 150 h
Prüfungsform	Klausur, 60 Minuten, Hilfsmittel: 1 DIN-A4-Seite handgeschrieben
Benotung	Gemäß § 20 der APO in der jeweils gültigen Fassung.

Inhalte des Moduls

Wie oft wird ein mehr oder weniger kleiner Scanner oder Parser benötigt? Häufig wird abenteuerlich auf selbst "erfundene" Scanner und Parser gesetzt. Um hier rechtzeitig den Absprung von ein paar Zeilen Code zu erleichtern, ist die Kenntnis des Aufbaus und der Wirkungsweise von Compilern wichtig.

In dieser Vorlesung wird die Funktionsweise und der von Parsern über Scanner bis zu Compilern erarbeitet. Hierbei wird der sinnvolle Einsatz von Werkzeugen basierend auf den theoretischen Grundlagen beschrieben.

Zunächst werden die theoretischen Grundlagen des Compilerbaus - die formalen Sprachen und die Automaten - erarbeitet. Hier wird ein Schwerpunkt auf CH-2 und CH-3 Sprachen gesetzt, die für Compiler besonders relevant sind. Aufbauend auf der Theorie wird dann die praktische Realisierung des Übersetzerbaus besprochen. Der Weg führt zur Konstruktion von Programmen zur lexikalischen und syntaktischen Analyse. Deren konkrete Realisierung wird an Hand allgemein verwendeter Programme veranschaulicht. Hierbei wird ein Compiler mit Hilfe gängiger Werkzeuge erstellt.

- Einführung
- Sprachentheorie: Grundlagen der Sprachanalyse
- Lexikalische Analyse
- lex / flex
- Die Syntaxanalyse
- Semantische Analyse
- Der Compiler-Generator YACC / BISON
- Zwischencodeerzeugung
- Code-Optimierung

Qualifikationsziele des Moduls, Lernziele und Kompetenzen

Nach erfolgreicher Teilnahme am Modul sind die Studierenden in der Lage:

- zu erkennen wo die Einsatzgebiete von Scannern und Parsern liegen.
- die Funktion und Wirkungsweise von Scannern und Parsern darzustellen.
- einen Compiler bestehend aus Scanner und Parser aufbauend auf der Theorie der Formalen Sprachen für eine gegebenen Aufgabenstellung zu entwerfen und zu erstellen.

Literaturliste

A.V. Aho, R. Sethi, J.D. Ullmann: Compilerbau. Band 1 und 2, Addison-Wesley 1999

A.V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman Compilers: Principles, Techniques, and Tools. Addison-Wesley, 2007.

A.W. Appel modern compiler implementation in java, Cambridge University Press 2004
Download unter:
<http://eden.dei.uc.pt/~amilcar/pdf/CompilerInJava.pdf>

B. Bauer, H. Höllerer: Übersetzung objektorientierter Programmiersprachen: Konzepte, Abstrakte Maschinen Und Praktikum: Java-Compiler; Springer; 4. Auflage; 2013

S.D. Bergmann Compiler Design: Theory, Tools, and Examples; free download: <http://elvis.rowan.edu/~sbergmann/books/cd/java/CompilerDesignBook.pdf> (Computer Science Department, Rowan University), 2016

H. Herold: Linux-Unix-Profitools. Addison-Wesley 1999

D. Grune, K. van Ree, H.E. Bal, C.J.H. Jacobs, K. Langendoen: Springer; 2. Auflage 2012

R.H. Güting, M. Erwig: Übersetzerbau; Springer 1999

A. Kunert: LR(k)-Analyse für Pragmatiker; Humboldt-Universität zu Berlin; Institut für Informatik / ZE Rechenzentrum (CMS) (Dissertation) 2011

Levine, J. R., Mason, T., Brown, D.: lex & yacc; O'Reilly & Associates 1995

A.J.D. Reiss. Compiler Construction using Java, JavaCC, and Yacc; Wiley, 2012.

F.J. Schmitt: Praxis des Compilerbaus; C. Hanser 1992

Wagenknecht C, Hielscher M.: Formale Sprachen, abstrakte Automaten und Compiler, Lehr- und Arbeitsbuch für Grundstudium und Fortbildung, Vieweg Teubner 2009
über Springer Link als download verfügbar!