

Lehrveranstaltung: IAM 2006: Compiler

Wechseln zu: [Navigation](#), [Suche](#)

Diese Lehrveranstaltung wird nicht mehr angeboten.

Studiengang	Interaktive Medien (IAM 2006)
Studienabschnitt	Vertiefungsphase
Modul	Wahlpflichtfach Informatik
Name	Compiler
Alternativname	Compiler
Name (englisch)	
Kürzel	COM.WP
Voraussetzungen	Es müssen mindestens 80 Credits in den Studienabschnitten Grundlagen- und Orientierungsphase und Aufbauphase erworben worden sein.
Wird gehalten:	veraltet
Semester	IAM 7
Lehrformen	
Credits	5
SWS	4 (Lehre: 3, Praktikum: 1, Teaching Points: 4)
Workload	Präsenzstudium: 60 h (durchschnittlich 4 h pro Woche) Eigenstudium: 90 h (durchschnittlich 6 h pro Woche)
Notengebung	Kommanote (1,0; 1,3; 1,7; 2,0, 2,3; 2,7; 3,0; 3,3; 3,7; 4,0; 5,0)
Gewichtung (Modulnote):	100 %
Verantwortliche(r)	Jürgen Scholz
Lehrende(r)	Jürgen Scholz
Homepage	https://www.hs-augsburg.de/fakultaet/informatik/studium/wahlpflichtveranstaltung/compiler/index.html

Inhaltsverzeichnis

- [1 Lernziele](#)
- [2 Inhalte](#)
- [3 Literatur](#)
- [4 Prüfungen](#)

1 Lernziele

Die Studierenden erkennen, wo die Einsatzgebiete von Scannern und Parsern liegen.

Die Studierenden sind basierend auf der Theorie der formalen Sprachen in der Lage sein, Scanner und Parser mit aktuellen Werkzeugen zu erstellen.

2 Inhalte

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 von Parsern über Scanner bis hin zu Compilern erarbeitet. Hierbei wird der sinnvolle Einsatz von Werkzeugen basierend auf den theoretischen Grundlagen beschrieben.

Nach einem kurzen Überblick über diverse kleine "Helferlein", die auf aktuellen Entwicklungsplattformen zur Verfügung stehen, werden die theoretischen Grundlagen - die formalen Sprachen und 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 (lex, yacc) erstellt.

Einführung

Sprachentheorie: Grundlagen der Sprachanalyse

Lexikalische Analyse

lex / flex

Die Syntaxanalyse

Semantische Analyse

Der Compiler-Generator YACC / BISON

Zwischencodeerzeugung

Code-Optimierung

3 Literatur

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

Herold, H.: Linux-Unix-Profertools. Adison-Wesley 1999

Jobst, F.: Compilerbau. C. Hanser 1991

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

Lutz, M., Schmitt, F. J.: Vom Prozessor zum Programm, Fachbuchverlag Leipzig 1997

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

Schreiner, A. T., Friedmann G.: Compiler bauen mit UNIX. C. Hanser 1991

Staubach, G.: Unix - Werkzeuge zur Textmusterverarbeitung. Springer 1989

Wilhelm, R., Maurer, D.: Übersetzerbau. Springer 1997

4 Prüfungen

Nummer	Prüfer	Zweitprüfer	Prüfung	Prüfungsart	Prüfungsdetails	Hilfsmittel
1930353	Jürgen Scholz	Gerhard Meixner	nicht mehr angeboten	Klausur	Dauer: 60 min	1 Din-A4-Seite handgeschrieben

Kategorie:
[Lehrveranstaltung](#)

Diese Seite wurde zuletzt am 28. September 2018 um 16:23 Uhr bearbeitet.
Inhalt verfügbar unter [CC BY-SA 4.0](#).

