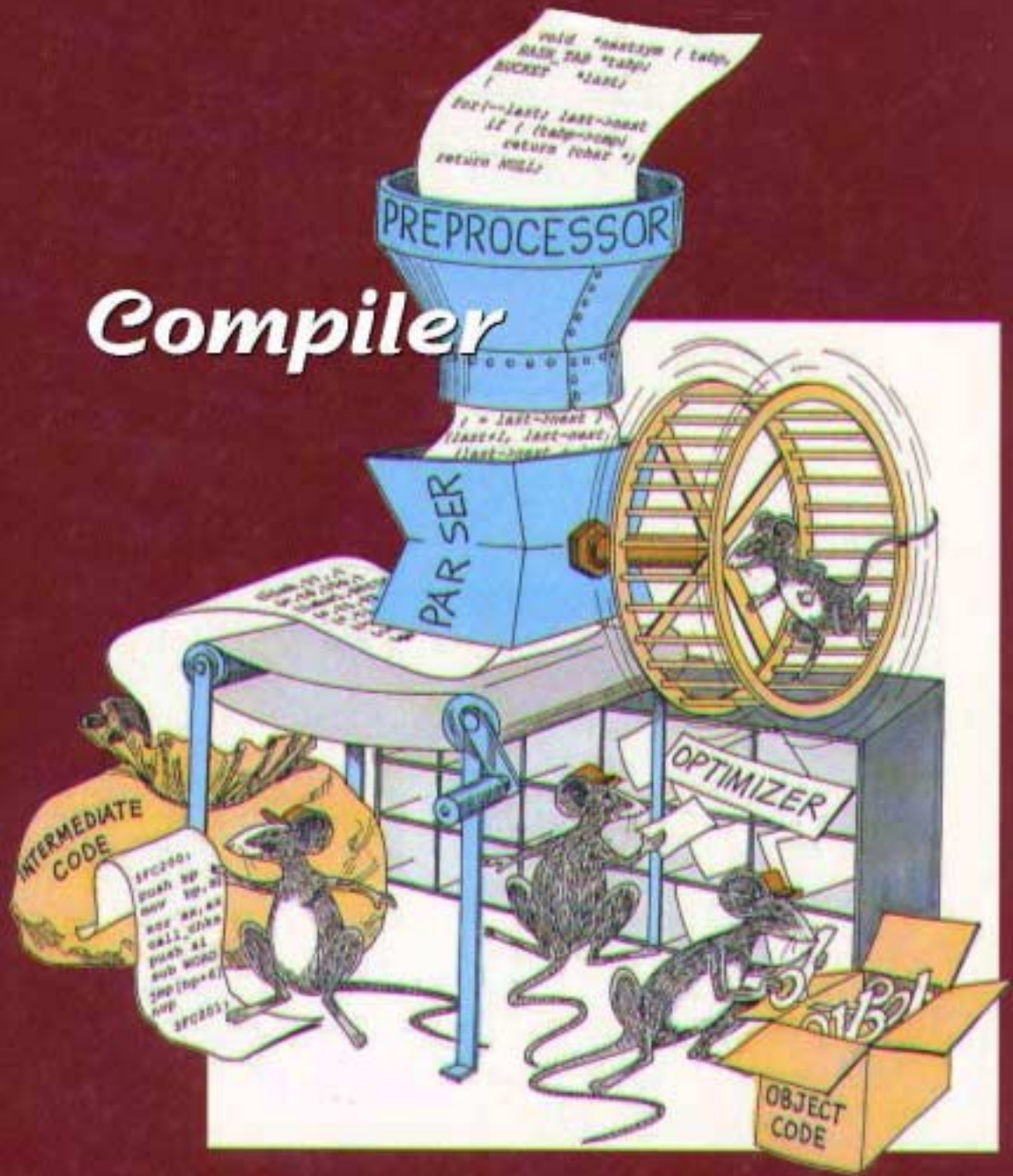


Compiler



Skriptum zur Vorlesung Compiler von H. Weber
8. Auflage - WS 2004/2005

Literatur

- A.V. Aho, R. Sethi, J. D. Ullman*: Compilerbau Teil I und Teil II, Addison-Wesley, Bonn.
- H. Alblas, A. Niemeyer*: Practice and Principles of Compiler Building with C, Prentice Hall, London.
- B. Bauer, R. Höllerer*: Übersetzung objektorientierter Programmiersprachen, Springer-Verlag Berlin.
- J.P. Bennett*: Introduction to Compiling Techniques – A first course using ANSIC, LEX and YACC, McGrawHill, London.
- R.E. Berry*: Programming Language Translation, Ellis Horwood, Chichester.
- C.N. Fischer, R. J. LeBlanc*: Crafting a Compiler, Benjamin/Cummings, Redwood City.
- C. Fraser, D. Hanson*: A Retargetable C Compiler : Design and Implementation, Benjamin/Cummings, Redwood City.
- R.H. Güting, M. Erwig*: Übersetzerbau – Techniken, Werkzeuge, Anwendungen, Springer-Verlag, Berlin
- J. Holmes*: Object-oriented Compiler Construction, Prentice Hall International, London.
- J. Holmes*: Building your Compiler with C++, Prentice Hall, Englewood Cliffs.
- A.I. Holub*: Compiler Design in C, Prentice Hall, Englewood Cliffs.
- R. Hunter*: The Essence of Compilers, Prentice Hall Europe, London
- F. Jobst*, Compilerbau: Hanser-Verlag, München.
- B. Klöppel, M. Paul, J. Rauch, J. Ruhland*: Compilerbau, Vogel-Verlag, Würzburg.
- H. Kopp*: Compilerbau, Hanser-Verlag, München.
- L.C. Louden*: Compiler Construction – Principles and Practice, PWS Boston.
- R. Mak*: Writing Compilers & Interpreters, John Wiley & Sons, New York.
- V. Penner*: Konzepte und Praxis des Compilerbaus, Vieweg-Verlag, Braunschweig.
- M. Rees, D. Robson*: Practical Compiling with Pascal-S, Addison-Wesley.
- F.J. Schmitt*: Praxis des Compilerbaus, Hanser-Verlag, München.
- P. D. Terry*: Programming Language Translation, Addison-Wesley, Wokingham.
- A.T. Schreiner, G. Friedman*: Compiler bauen mit UNIX, Hanser-Verlag, München.
- J. Tremblay, P. Sorenson*: The Theory and Practice of Compiler Writing, McGraw-Hill, New York
- W.M. Waite, G. Goos*: Compiler Construction, Springer-Verlag, Berlin.
- D.A. Watt*: Programming Language Processors, Prentice Hall, New York.
- N. Wirth*: Compilerbau; Teubner-Verlag, Stuttgart.
- H. Zima*: Compiler, Teil 1 und Teil 2, BI-Verlag, Mannheim.

Vorwort

Die 1. Auflage dieses Skriptums zur Vorlesung Compiler (eigentlich müßte es Compilerbau heißen) wurde von meinem damaligen Studenten, Herrn Dipl.-Inf. (FH) Dirk Kiesow nach Vorlesungsmitschriften und meinem Manuskript angefertigt. Dafür sei ihm herzlich gedankt. Mein besonderer Dank gebührt auch meinem Kollegen Prof. H. Kopp von der FH Regensburg für die Überlassung der PL/0-Software, die sich bei der Compiler-Ausbildung am FB Informatik der FH Wiesbaden bewährt hat.

Die 2. Auflage wurde vom Dozenten durchgesehen, von einigen Fehlern befreit und mit einigen Umstellungen versehen. Die 3. Auflage endlich wurde nach MS Word for Windows 2.0 übertragen und neu überarbeitet und ergänzt. Das Kapitel 9 über Werkzeuge ist neu hinzugekommen. In der 4. und 5. Auflage wurden wieder einige anscheinend unvermeidliche Fehler korrigiert und kleinere Ergänzungen vorgenommen. Für die 6. Auflage wurde das Skript von mir neu bearbeitet zwecks Umstellung der Beispiele von Pascal auf C. Meinem Sohn Stefan danke ich für zusätzliche Korrekturarbeiten. In der nunmehr 7. Auflage wurde Ergänzungen eingebaut. Die 8. Auflage enthält ein neues Kapitel 10 über Erweiterung und Portierung.

Eine Garantie für die Fehlerfreiheit kann nicht übernommen werden. Prüfungsrelevant ist in jedem Fall der in der Vorlesung und den Übungen behandelte Stoff und nicht der Inhalt des Skriptums.

Helmut Weber

Inhaltsverzeichnis

Übersicht	II
1 Einleitung	1-1
1.1 Compiler	1-1
1.2 Das Analyse-Synthese-Modell der Compilierung	1-2
1.3 Die Analyse des Quellprogramms	1-2
1.4 Die Phasen eines Compilers	1-4
1.5 Umgebung eines Compilers	1-6
1.6 Zusammenfassung von Phasen	1-7
2 Sprachanalyse	2-1
2.1 Formale Sprachen und Grammatiken	2-1
2.2 Die Chomsky-Hierarchie	2-2
2.3 Eigenschaften von Grammatiken	2-5
2.4 Darstellung von Produktionen	2-7
2.5 Die Modellsprache PL/0	2-9
2.6 Darstellung der Semantik von Programmiersprachen	2-13
3 Lexikalische Analyse	3-1
3.1 Die Rolle des Scanners	3-1
3.2 Implementationsmöglichkeiten	3-3
3.3 Lexikalische Analyse von PL/0 als Beispiel	3-3
4 Syntaktische Analyse	4-1
4.1 Zum Prinzip der Top-Down-Analyse	4-1
4.2 LL(1)-Grammatiken	4-4
4.3 Recursive-Descent-Parser	4-7
4.4 Der PL/0-Parser als Beispiel	4-10
4.5 Tabellengesteuerte Top-Down-Parser	4-11
4.6 Grundlagen der Bottom-Up-Syntaxanalyse	4-13
5 Semantische Analyse	5-1
5.1 Überblick	5-1
5.2 Typ-Überprüfung	5-1
6 Fehlerbehandlung	6-1
6.1 Allgemeine Gesichtspunkte	6-1
6.2 Lexikalische Fehler	6-2
6.3 Syntaktische Fehler	6-2
6.4 Semantische Fehler	6-5
7 Laufzeit-Speicherverwaltung	7-1
7.1 Grundlagen	7-1
7.2 Unterteilung des Laufzeitspeichers	7-2
7.3 Activation Records (AR)	7-3
7.4 Dynamic-Link- und Static-Link-Ketten	7-6
7.5 Verwaltung dynamischer Arrays	7-8

8	Code- und Zwischencode-Generierung	8-1
8.1	Syntaxorientierte Übersetzung	8-1
8.2	Zwischensprachen	8-1
8.3	Semantische Aktionen	8-4
8.4	Die Zwischensprache von PL/0	8-9
8.5	Code-Erzeugung und Interpretation	8-12
8.6	Erzeugung von Assemblercode aus dem Zwischencode.....	8-12
9	Compilerbau-Werkzeuge	9-1
9.1	Überblick	9-1
9.2	Der Scanner-Generator Lex	9-1
9.3	Der Parser-Generator Yacc	9-5
9.4	Problematik der Benutzung von Lex und Yacc	9-7
10	Erweiterung und Portierung	10-1
10.1	Überblick	10-1
10.2	Bootstrapping begl. Teilmengen von Sprachen	10-2
10.3	Portierung	10-4
	Anhang (Quellen des PL/0-Compilers in C).....	A-1