Compilerbau

WS 2011/2012

Dr. Jutta Göers

Do, 8:15 – 9:45 Uhr, Raum E05 (17.11.: 449a)
Rechnerraum 369
Klausur: voraussichtlich 9.2.2012
http://www-lehre.inf.uos.de/cb/ws1112

Literatur

- ***Compiler: Prinzipien, Techniken, Werkzeuge*** von A.V.Aho, M.S. Lam, R. Sethi, J.D. Ullman, Pearson, 2008
- ***Übersetzerbau*** von R. Wilhelm, D. Maurer, Springer, 1997
- ***Compilerbau*** von F. Jobst, Hanser, 1992
- ***Compiler bauen mit UNIX*** von A.T. Schreiner, G. Friedmann, Hanser, 1985
- ***lex und yacc*** von H. Herold, Addison-Wesley, 1995
- ***lex & yacc*** von J.R. Levine, T. Mason, D. Brown, O‘Reilly, 1992

Motivation

- nur selten neue Compiler
- aber kleine Teilaufgaben:
  - Interpreter für kleine Kommandosprache
  - Benutzereingaben auf Vollständigkeit /Korrektheit prüfen
- praktische Anwendung der theoretischen Informatik
- viele weitere Techniken: Stringverarbeitung, Suchverfahren, Optimierung, …

Compilerbau

Inhalte (geplant):

1. Einführung:
   1. Programmiersprachen
   2. Sprachprozessoren
2. Allgem. Aufbau von Compilern
3. Sprachkonzepte und ihre Übersetzungen
4. lexikalische Analyse (lex, flex)
5. syntaktische Analyse (yacc, bison)
6. semantische Analyse
7. Optimierungen
8. Codeerzeugung

1. Einführung

Sprachgenerationen:

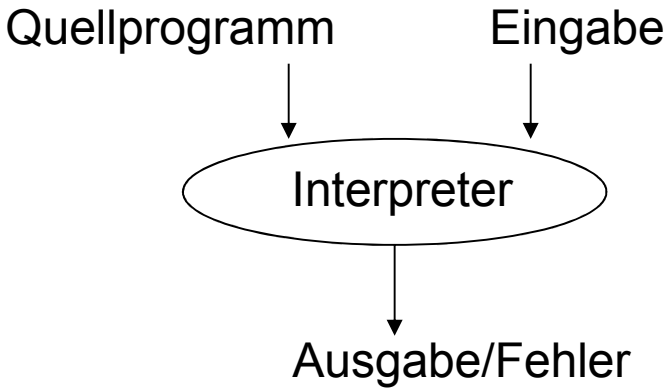
(1) Maschinensprachen
(2) Assembler
(3) Höhere Programmiersprachen
- funktionale
- logische
- prozedurale
- objektorientierte

(4) Anwendungssprachen

1. Einführung

Sprachprozessoren:

(1) interpretierende



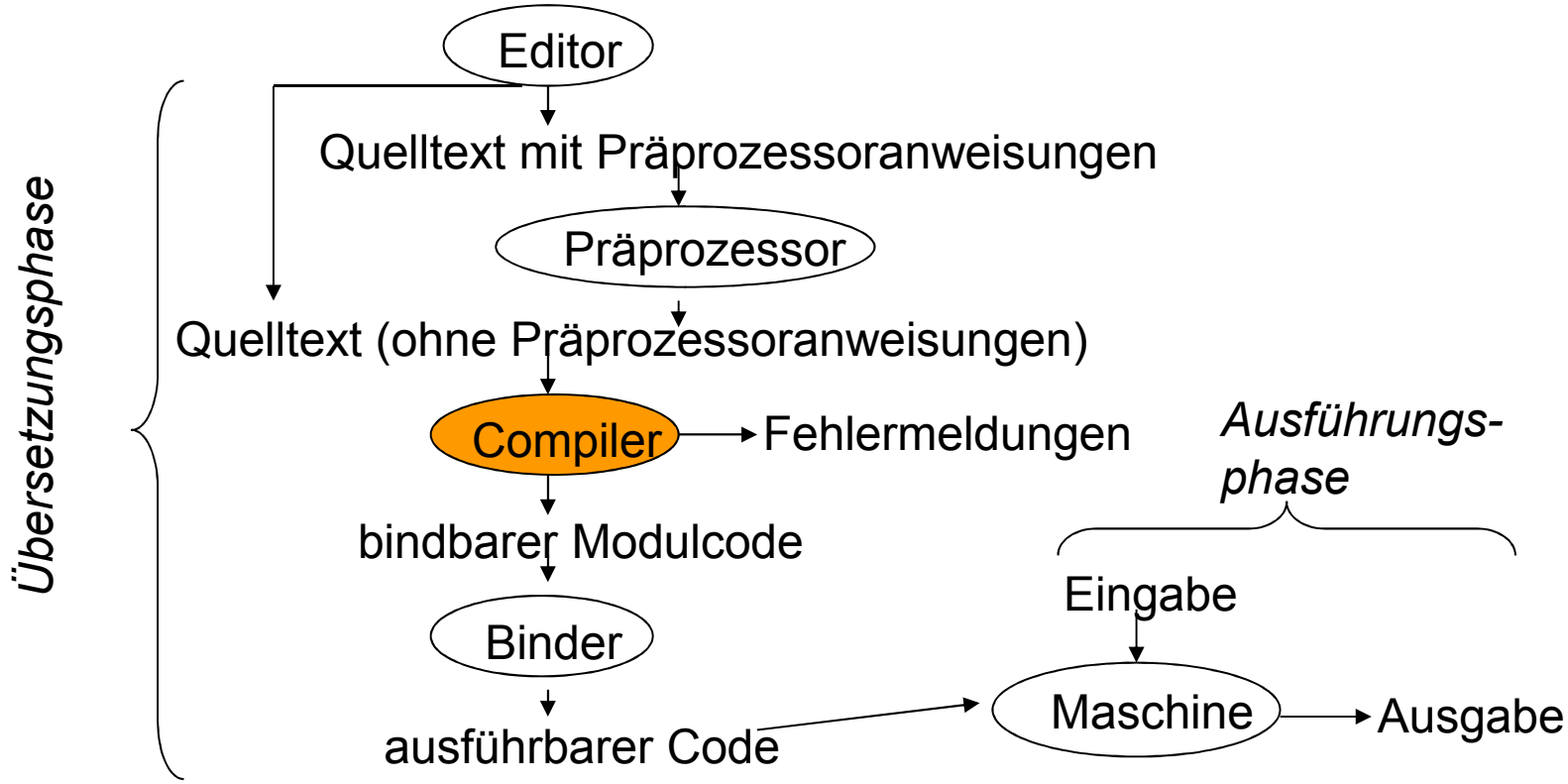
Kennzeichen:

- bearbeitet Programm und Eingabe zur gleichen Zeit
- nutzt keine von Eingabedaten unabhängigen Informationen

1. Einführung

(2) übersetzende

- Vorverarbeitung des Quellprogramms unabhängig von Eingabedaten
- Zielprogramm ermöglicht effizientere Ausführung mit konkreten Eingabedaten



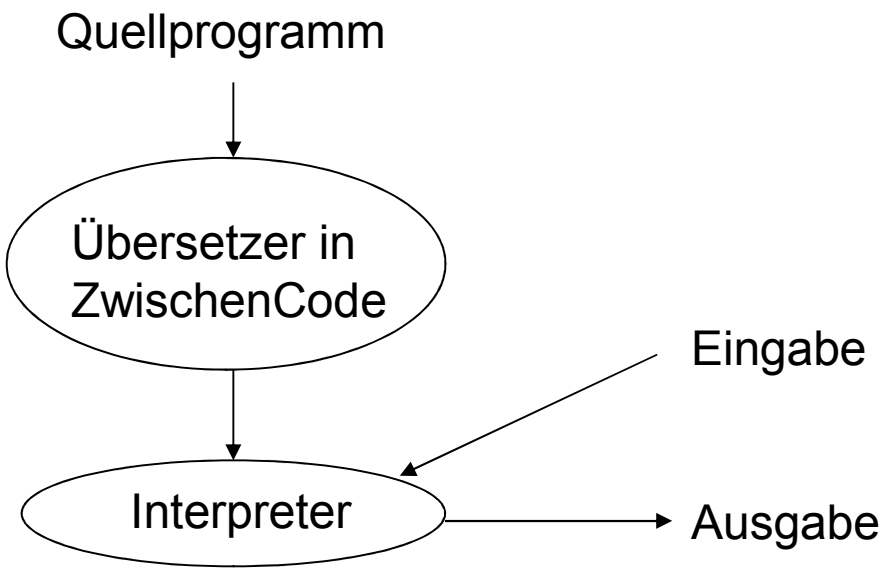
1. Einführung

Übersetzer:

- Compiler-Entwicklung seit den 50er Jahren
- Compiler selbst ein Programm
- übersetzt Quellprogramm (in Quellsprache) in ein Zielprogramm (in Zielsprache)
- wichtige Teilaufgabe: Fehler melden
- viele verschiedene Quellspr. → viele Compiler
- viele verschiedene Zielspr. → noch mehr Compiler
- verschiedene Arten von Compilern: Ein-Pass-, Mehr-Pass-, optimierende, mit Testhilfen, …

1. Einführung

(3) hybride Systeme



- Quelltext durchläuft Analyse-Part des Compilierens
- Zwischensprache wird interpretiert

1. Einführung

Fragestellungen zu Kapitel 1:

- Welche Generationen von Programmiersprachen (und welche Paradigmen darin) gibt es?
- Welche Arten von Sprachprozessoren gibt es?
- Wie arbeiten die Sprachprozessoren?
- Was sind die jeweiligen Vor- und Nachteile der Sprachprozessoren?
- Was versteht man unter einem Ein-Pass-Compiler, was unter einem Mehr-Pass-Compiler?