Übungen zu
Grundlagen der Informatik für Studierende des Maschinenbaus
im Sommersemester 2001
(Dipl.-Inf. Jörg Barner)

Aufgabe 15: Assemblierer

a) Warum muß ein Assemblierer in der Regel ein Assemblerprogramm mehrfach bearbeiten?

b) Welche Rolle spielt dabei die Symboltabelle?

c) Was versteht man unter einer externen Adresse?

Lösung:

a) Wenn Symbole vor ihrer Definition verwendet werden ("Vorwärtsreferenzen"), ist der Wert, der für sie eingesetzt werden muß, noch nicht bekannt. Daher muß der Assemblierer solche Werte in einem zweiten Durchlauf — also nachdem alle Definitionen gesehen worden sind — einsetzen.

b) Aus der Symboltabelle kann der Assemblierer ersehen, ob ein Symbol bereits definiert wurde oder noch nicht.

c) Wenn ein Programm in mehrere Quelltexte aufgeteilt ist, die getrennt übersetzt werden, muß i.a. auch auf Symbole Bezug genommen werden, die in anderen Modulen definiert werden. Deren Wert kann aber nicht während der Übersetzung ermittelt werden, sie müssen deshalb als externe Adressen stehenbleiben und werden später vom Binder aufgelöst.

Aufgabe 16: Binder
Was versteht man unter einem Binder?

Lösung:
Der Binder (Linker) verbindet die durch getrennte Übersetzung entstandenen einzelnen Objektmodule eines Programms und eventuell zusätzlich Bibliotheken, Laufzeitsystem etc. zu einem lauffähigen Programm. Seine Hauptaufgabe ist es, externe Bezüge in den einzelnen Objekten zu finden und aufzulösen.

Aufgabe 17: Lader
Was versteht man unter einem Lader?

Lösung:
Der Lader liest ein ausführbares Programm (die Ausgabe des Binders) in den Speicher und paßt (wenn nötig) absolute Adreßangaben an die tatsächliche

Lage im Speicher an. Schließlich bringt er das Programm zur Ausführung. Der Lader is meist Teil des Betriebssystems.

Aufgabe 18: Compiler/Interpreter

a) Wie unterscheiden sich Compiler und Interpreter?

b) Grenzen Sie Vor- und Nachteile von Compilern und Interpretern gegeneinander ab. Wann benutzt man was?

Lösung:

a) Ein Compiler übersetzt ein Programm, das in einer höheren Programmiersprache geschrieben ist, in Maschinencode, der vom Rechner ohne weitere Hilfsmittel verstanden wird. Ein Interpreter dagegen führt den Quellcode direkt aus, d.h. er interpretiert die Anweisungen der höheren Sprache und führt sie unmittelbar — ohne Maschinencode zu erzeugen — aus.

b) Compiler: Übersetzung in Maschinensprache benötigt Zeit, dafür ist das fertige Programm schneller.

   Interpreter: Nach Änderungen keine Neuübersetzung notwendig, aber weniger effizient, da der Quellcode bei jeder Ausführung übersetzt werden muß, nicht nur einmal.

   Ein Interpreter hat also Vorteile während der Entwicklung eines Programms, da Änderungen ohne lange turn-around-Zeiten schnell getestet werden können. Wenn aber keine Änderungen mehr nötig sind, ist das vom Compiler erzeugte Maschinenprogramm deutlich schneller, weil die Übersetzungsarbeit vorweggenommen wurde und nicht mehr bei jedem Ablauf anfällt.
