

Im Laufe der Zeit wurden verschiedene Programmiersprachen verwendet.

Maschinencode Programmiersprache 1. Generation

Der Maschinencode oder die Maschinensprache ist eine Programmiersprache, die der Computer unmittelbar versteht. Maschinensprachen sind für Menschen schwer verständlich und unübersichtlich.

Sie bestehen aus binären oder hexadezimalen Zahlen. Das könnte etwa so aussehen:

```
0202 03E8 0203 03E8 0602 1801 10FD
```

Dieses Programm veranlasst einen bestimmten Computer, auf tausend zu zählen.

Assembler

Programmiersprache 2. Generation

Ein Kompromiss sind die Assemblersprachen, in denen die Zahlen durch leicht merkbare Buchstabengruppen dargestellt werden. Die Maschinensprache ist die unterste Stufe der Computerprogrammierung und die einzige Sprache, welche ein Computer ohne Hilfe eines anderen Programms ausführen kann.

Programme in Maschinensprache werden für einen bestimmten Rechner oder Rechnertyp entwickelt und sind nicht damit nicht auf anderen Rechnern ausgeführt werden können. D. h. sie sind nicht portabel.

Beispiele von Programmiersprachen: M68000, 386

Das Assemblerprogramm weist den Computer an, auf tausend zu zählen:

```
LI 2,1000
LOOP DEC 2
JOC END
JMP LOOP
END
```

Die Assembler-Sprache stellt für jeden elementaren Maschinenbefehl einen mnemotechnischen, maschinengebundenen Code bereit. Dabei bedeutet **mnemotechnisch** erlernbar, verständlich, indem z. B. für den Maschinenbefehl 03EF der Assemblerbefehl INC benützt wird, was soviel bedeutet wie addiere 1 zum Inhalt des Rechenregisters

Das Assemblerprogramm oder der Assembler übersetzt den Source-Code oder Quellcode (Assembler - Code) in Maschinencode um, welcher direkt ausgeführt werden kann.

Auch beim Assembler gilt, dass Assembler - Programme für einen bestimmten Rechner oder Rechnertyp geschrieben werden und damit nicht auf anderen Rechnern ausgeführt werden können. Diese Schwachstelle versuchte man mit einem Cross-Assembler zu umgehen, indem der Cross-Assembler Assembler-Code in Maschinen-Code eines anderen Rechners umsetzt. Damit werden Assemblerprogramme (bedingt) portierbar.

Unter **SOURCECODE** wird auch die Datei verstanden, welche die Anweisungen des Programms enthält.

Beispiele von Assemblersprachen sind: MAKRO, MASM, Turbo Assembler

Programmiersprache 3. Generation

Prozedurale Sprachen für spezielle Anwendungsbereiche. Symbolische, problemorientierte Sprachen bei welchen der Ablauf eines Programms durch die Reihenfolge der Anweisungen festgelegt wird (nicht mehr maschinenorientiert, portabel).

INTERPRETER

Ein Maschinenprogramm, das einzelne Befehle einer höheren Sprache in Maschinensprache übersetzen und direkt ausführen kann, also eine Art Simultanübersetzer.

BASIC ist eine Interpreter-Sprache.

COMPILER

Der Sourcecode wird in Maschinencode übersetzt und in Dateien mit Objectcode abgespeichert. Diese Module müssen mit dem Linker zu einem ausführbaren Programm zusammengefügt werden.

LINKER

Der Linker fügt verschiedenen Objectcode mit weiteren Ressourcen (Modulen, Bibliotheken, usw.) zu einem ausführbaren Programm zusammen.

Beispiele von Programmiersprachen der 3. Generation sind:

COBOL, FORTRAN, PASCAL, MODULA-2, usw.

Programmiersprache 4. Generation

Nichtprozedurale Sprachen für unterschiedliche Anwendungen. Nicht - von - Neumann - Sprachen.

Symbolische, problemorientierte Sprachen, bei welchen angegeben wird, was das Programm tun soll, ohne jedoch anzugeben, wie dies im einzelnen zu geschehen hat (Deskriptivität, Objekt - Orientiert)

Beispiele von Programmiersprachen der 4. Generation sind: MySQL, C++, Pascal MODULA-2

Programmiersprache 5. Generation

AI Artificial Intelligence, KI Künstlichen Intelligenz

Beispiele von Programmiersprachen der 5. Generation: LISP, PROLOG, SMALLTALK, LOGO.

RAD

Rapid Application Development

Mit der horrenden Verbreitung graphischer Oberflächen (im Besonderen Windows) wurden vermehrt graphisch und objektorientierte Sprachen bzw. Entwicklung-Umgebungen durch, wie dies z. B. mit Visual Basic von Microsoft auf der einen und Delphi bzw. Cbuilder der Fall ist. Dabei werden alle Windows basierten Teile so gekapselt, dass der Entwickler sich lediglich noch mit seinem Code beschäftigen muss.

Web

Webbasierte Programmentwicklung

Während in den Anfängen Interpreter kaum eine Chance hatten, erleben diese mit immer schnelleren Prozessoren eine ungeahnte Renaissance, indem Internet-Browser prozedurale Programme langsam verschwinden lassen. Grundlagen sind die Seitenbeschreibungssprache HTML und Sprachen wie PHP, Java, usw.

Mittlerweile stehen AMP-Systeme kostenfrei zur Verfügung, um lokal einen Webserver (Apache), ein Datenbanksystem (MySQL) und einen PHP Interpreter zu betreiben

Ebenso stehen verschiedene webbasierte Entwicklungsumgebungen völlig kostenfrei zur Verfügung. Beispiel sind NetBeans, Eclipse, usw. Gleichzeitig stehen immer mehr Programmpakete oder Frameworks zur Verfügung, welche den Programmieraufwand für Web-Anwendungen auf das absolut Wesentliche reduzieren - ein Beispiel ist Laravel.