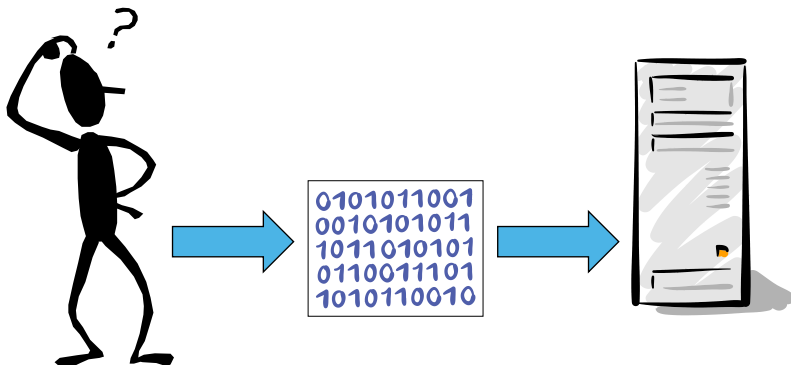


Einführung in Programmiersprachen allgemein

Prof. Dr. Helmut G. Folz

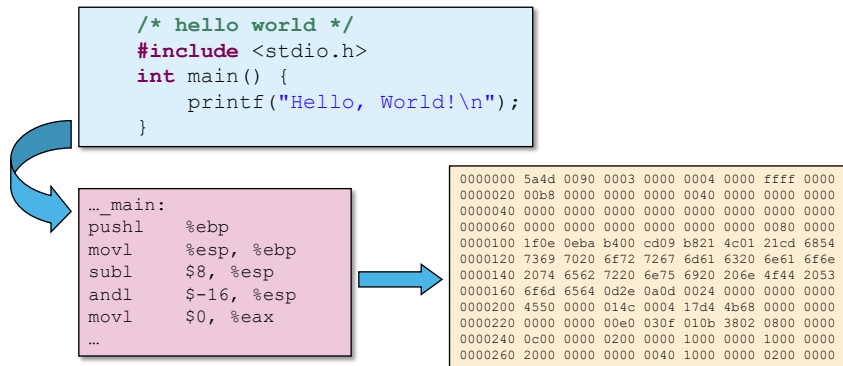


Wie kommuniziert eigentlich ein Mensch mit einem
Computer?



Was ist ein Computerprogramm?

- Ein (Computer-)**Programm** ist eine Folge von Anweisungen und Ausdrücken, die von einem Computer verstanden und ausgeführt werden können.



Was ist ein Computerprogramm?

Mögliche Antworten:

- **Programm** = Text mit Anweisungen für einen Computer;
- Ein **Programm** ist eine Folge von Anweisungen und Ausdrücken, die von einem Computer verstanden und ausgeführt werden können.
- **Informatik-Duden**: Formulierung eines Algorithmus und der zugehörigen Datenbereiche in einer Programmiersprache.

Programmiersprache

- **Programmiersprachen** bilden die wichtigste Schnittstelle zwischen Menschen und Computern.
- Eine **Programmiersprache** ist eine Sprache zur Formulierung von Rechenvorschriften, d.h. von Datenstrukturen und Algorithmen, die von einem Computer ausgeführt werden können.
(*Informatik-Duden*)

Maschinensprachen (1. Generation)

- Die Sprachen der 1. Generation sind **Maschinensprachen**, deren Elemente direkt aus dem Befehlsvorrat bestimmter Computer abgeleitet sind.
- Maschinensprache ist spezifisch für genau einen Prozessor und ein Betriebssystem und damit in höchstem Maße unportabel.
- Maschinenprogramme werden heute nicht mehr direkt geschrieben.

Beispiel: Maschinensprache

```
0000000 5a4d 0090 0003 0000 0004 0000 ffff 0000
0000020 00b8 0000 0000 0000 0040 0000 0000 0000
0000040 0000 0000 0000 0000 0000 0000 0000 0000
0000060 0000 0000 0000 0000 0000 0000 0080 0000
0000100 1f0e 0eba b400 cd09 b821 4c01 21cd 6854
0000120 7369 7020 6f72 7267 6d61 6320 6e61 6f6e
0000140 2074 6562 7220 6e75 6920 206e 4f44 2053
0000160 6f6d 6564 0d2e 0a0d 0024 0000 0000 0000
0000200 4550 0000 014c 0004 17d4 4b68 0000 0000
0000220 0000 0000 00e0 030f 010b 3802 0800 0000
0000240 0c00 0000 0200 0000 1000 0000 1000 0000
0000260 2000 0000 0000 0040 1000 0000 0200 0000
0000300 0004 0000 0001 0000 0004 0000 0000 0000
0000320 5000 0000 0400 0000 4872 0000 0003 0000
0000340 0000 0020 1000 0000 0000 0010 1000 0000
0000360 0000 0000 0010 0000 0000 0000 0000 0000
0000400 4000 0000 01ec 0000 0000 0000 0000 0000
0000420 0000 0000 0000 0000 0000 0000 0000 0000
```

...und noch viele viele weitere Zeilen.

Assemblersprachen (2. Generation)

- **Assemblersprachen** sind Programmiersprachen, in denen Maschinenbefehle durch leichter lesbare Abkürzungen ersetzt sind.
- Die Zielmaschine bestimmt die Struktur der Sprache.

Aus dem Maschinenbefehl

```
10110000 01100001
```

wird so

```
mov al, 61h ;
```

Bedeutung: Lade den Hexadezimalwert 61 in den ersten Teil des Registers a

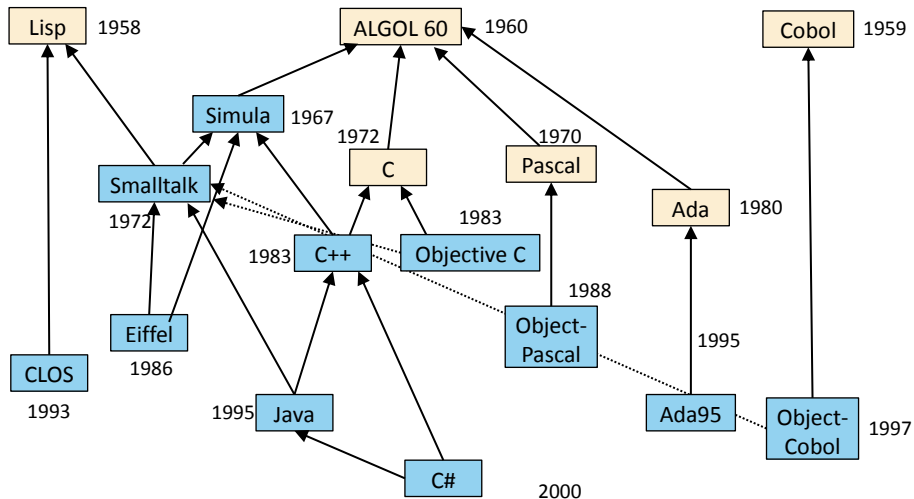
Beispiel: Assembler

```
.file "helloC.c"
.def __main; .scl 2; .type 32; .endef
.text
LC0: .ascii "Hello, World!\12\0"
.globl _main
.def _main; .scl 2; .type 32; .endef
_main:
    pushl %ebp
    movl %esp, %ebp
    subl $8, %esp
    andl $-16, %esp
    movl $0, %eax
    movl %eax, -4(%ebp)
    movl -4(%ebp), %eax
    call __alloca
    call __main
    movl $LC0, (%esp)
    call _printf
    leave
    ret
.def _printf; .scl 2; .type 32; .endef
```

Höhere Programmiersprachen (3. Generation)

- Sprachen der 3. Generation orientieren sich an den zu lösenden Problemen und nicht mehr an den benutzten Maschinen.
- Die meisten zeitgemäßen Programmiersprachen fallen in diese Kategorie.

Historie der Programmiersprachen



Prof. Dr. H. G. Folz

Programmierung 1: Einführung in Programmiersprachen allgemein

-11-

Syntax

- Die **Syntax** regelt die "Rechtschreibung" einer Programmiersprache. Sie lässt sich formal festlegen, z.B. mit Syntaxdiagrammen oder in einer Grammatik.
- Informatik-Duden:
 - ⇒ Eine Sprache wird durch eine Folge von Zeichen, die nach bestimmten Regeln aneinandergereiht werden dürfen, definiert.
 - ⇒ Den hierdurch beschriebenen formalen Aufbau bezeichnet man als ihre Syntax.

Prof. Dr. H. G. Folz

Programmierung 1: Einführung in Programmiersprachen allgemein

-12-

Semantik

- Lehre von der inhaltlichen Bedeutung einer Sprache
- Die **Semantik** regelt die Bedeutung einzelner Sprachelemente und ihr Zusammenspiel. Sie lässt sich viel schwieriger exakt festhalten.
- Es gibt formale Beschreibungsmethoden für die Semantik.
- Üblicherweise werden wir die Semantik von bestimmten Sprachkonstrukten umgangssprachlich erläutern.

Übersetzen von Programmen

- Computer können generell nur Maschinenprogramme ausführen.
- Alle anderen Sprachen müssen erst in ein Maschinenprogramm übersetzt werden, bevor sie ausgeführt werden können.
- Übersetzer oder **Compiler**
 - ⇒ Compiler transformieren Programme direkt oder in verschiedenen Stufen in Maschinensprache
 - ⇒ Compiler sind selbst sehr komplexe Programme

Übersetzen von Programmen

- **Quelltext und ausführbarer Code:**

- ⇒ Der Programm-Text selbst wird "Quelltext" genannt (auf neudeutsch Source-Code).
- ⇒ Das vom Compiler produzierte und dann ausführbare Maschinenprogramm heißt "übersetztes Programm" oder "Binärprogramm" (engl. binary oder executable).
- ⇒ Soweit der Compiler fehlerfrei arbeitet, verhält sich das Binärprogramm genau so, wie es im Quelltext beschrieben steht.
- ⇒ Syntaxfehler im Quelltext führen zu Fehlermeldungen des Übersetzers und normalerweise dazu, dass kein ausführbares Programm erzeugt wird.

Übersetzen von Programmen

- **Interpreter**

- ⇒ Interpreter übersetzen den Quelltext erst beim Ausführen in Maschinensprache. (z. B. bei BASIC oder Skriptsprachen)
- ⇒ Interpretierte Programme sind langsamer als Binärprogramme, da zum Ausführen des Programms noch das Übersetzen hinzu kommt und naturgemäß vom Interpreter nicht so gut optimiert werden kann wie vom Compiler.
- ⇒ Nicht alle Programmiersprachen eignen sich zum interpretieren, z.B. C und C++.

Übersetzen von Programmen

- **Zwischencodes**

- ⇒ Es gibt auch Programmiersprachen wie z. B. Java, die nicht zu Binärcode übersetzt werden, sondern in einen sogenannten Zwischencode.
- ⇒ Zwischencodes sind meist plattformübergreifende Zwischensprachen, die auf der jeweiligen Zielplattform interpretiert werden müssen.
 - bei Java heißt das **Bytecode**
 - bei .NET von Microsoft heißt das **Common Intermediate Language**
- ⇒ Vorteil von Zwischencodes ist, dass ein einmal übersetztes Programm auf allen Rechnerplattformen ablauffähig ist, auf denen ein Interpreter für den Zwischencode vorhanden ist.