Maschinencode

Maschinencode ist die Sprache, die der Prozessor versteht. Jeder Prozessortyp hat seine eigene Variante dieses Codes. Die Grundlagen sind jedoch gleich. Die Maschinensprache setzt sich aus aneinander gereihten Instruktionen zusammen. Jede Instruktion besteht aus einem OP-Code, also der Anweisung was genau getan werden soll und den dazugehörigen Argumenten, die in die Instruktion kodiert werden. Je nach Architektur ist die Größe und die Verteilung der Argumente unterschiedlich.

Ein Beispiel einer Instruktion an Hand des Intel IA 32 Prozessors(1):

|  |  |
| --- | --- |
| Instruktion als Bytes | 48 B8 8877665544332211 |
| Instruktion als Text | MOV RAX,1122334455667788H |
| Beschreibung | Schreibe den Wert von Memory Offset 1122…88H in Register RAX |

(Das Präfix "h" am Ende einer Zahl sagt aus, dass diese in Hexadezimaler Schreibweise ist)

Das Beispiel zeigt deutlich die Struktur einer Instruktion:

48 B8 8877665544332211

OpCode Argument Argument

(Mov) (Register RAX) (Memory Offset 1122..88H)

Die Reihenfolge der Argumente ist je nach Prozessor unterschiedlich. Die meisten Instruktionen benötigen 1 bis 2 Argumente, selten 3. Es gibt auch Ausnahmen, die keine Argumente, zum Beispiel die "ret"-Anweisung, benötigen.

(1) <https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-instruction-set-reference-manual-325383.pdf> (S. 45)