

PROBLEM 1:**a) Implizite Register-Adressierung mit Flag**

Ein Befehl bezieht sich implizit auf ein Register. Bsp.: ``DIV _``

b) Unmittelbare Adressierung

In einem Befehl wird ein Wert direkt angegeben. Bsp.: ``MOV _, 1``

c) Direkte Speicheradressierung

Die Speicheradresse wird direkt angegeben. Bsp.: ``[100]``

d) Register-indirekte Speicheradressierung

Die Speicheradresse liegt in einem Register. Bsp.: ``[RDX]``

e) Register-relativ-indizierte Speicheradressierung

Die Speicheradresse wird relativ zu einem Indexregister angegeben. Bsp.: ``[RBX+4]``

f) Register-relativ-indirekt-indizierte, zweistufige Speicheradressierung

Die Speicheradresse wird relativ zu einem Indexregister angegeben, dabei liegt der Offset Wert auch in einem Register. Bsp.: ``[RBX+4*RCX]``

g) Programmzähler-relativ-indirekte, zweistufige Speicheradressierung

Die Speicheradresse wird relativ zum Programmzähler (RIP) angegeben, dabei liegt der Offset Wert auch in einem Register. Bsp.: ``[RIP+RCX]``

PROBLEM 2:**a) Pseudoinstruction**

Pseudoinstruction sind Befehle, welche vom Assembler nicht in ausführbaren Maschinencode umgewandelt werden, aber trotzdem Teil der Assembler Sprache sind, wie ``db`` oder ``equ``.

b) Linker

Ein Linker ist ein Computerprogramm, welches kompilierten Code in ein ausführbares Programm umwandelt. Dabei linked der Linker durch Symbole spezifizierte Programmodule aus einer Library (des Betriebssystems), die für die Ausführung benötigt werden.

c) Dynamic Link Library

Bei dynamischem Linken werden manche Symbole erst während der Programmausführung aufgelöst. Diese sind dann in der Dynamic Link Library (DLL) gespeichert, was den Vorteil hat, dass sie separat geupdated werden können.

d) Macro

Ein Macro ist ein Codeblock (mit Parametern), welcher an verschiedenen Stellen des Programms aufgerufen werden kann. Macros werden dabei beim assemblieren direkt an allen Stellen, an denen sie aufgerufen wurden, in den Bittcode eingefügt.

e) Koroutine

Koroutinen sind eine Verallgemeinerung von Subroutinen, die es erlauben zwischen zwei Codeabschnitten hin und her zu springen. Sie erlauben also mehrere Entry- und Return-Stellen.