

Einführung in IBM Z Assembler inkl. Debugging

für

- **Systemprogrammierer**
- **Anwendungsentwickler**

Version 1.9 vom Januar 2026

Autoren: Wolfram Greis / Rolf Brüning

European Mainframe Academy GmbH
Ernst-Reuter-Platz 6
10587 Berlin
info@mainframe-academy.de

Inhaltsverzeichnis

1	Ziele des Ausbildungsmoduls.....	3
2	Randbedingungen / Voraussetzungen.....	3
3	Inhalt.....	4
3.1	Modul Assembler incl. Debugging (ca. 90 Stunden).....	4

1 Ziele des Ausbildungsmoduls

Ziel des Moduls ist es, Teilnehmern die maschinennahe Programmiersprache **IBM High Level Assembler** beizubringen. Die Teilnehmer können nach diesem Modul Assemblerprogramme verstehen, interpretieren und anpassen sowie eigene Programme in Assembler schreiben.

Systemnahe Programmierung in Assembler setzt voraus, dass sich die Teilnehmer mit der Architektur und den internen Strukturen der IBM z Systems Architektur auseinandersetzen. Somit geht es bei der Vermittlung von Assembler nicht nur um Programmierung, sondern auch um das Verständnis der Architektur und wie die diversen Komponenten der Architektur zusammenspielen.

2 Randbedingungen / Voraussetzungen

Die Kenntnisse der grundlegenden Logik für prozedurale Programmierung werden vorausgesetzt. Außerdem der Umgang mit TSO / ISPF / JCL und SDSF.

Den Teilnehmern wird für die Dauer der Ausbildung ein Zugang auf dem z/OS System der EMA (aktuell: z/OS 3.1) zur Verfügung gestellt. Dieser Zugang steht auch nach Beendigung der Ausbildung zur Verfügung. Die EMA garantiert diesen Zugang für mindestens drei Monate nach Ausbildungsende.

Der Zugang ist über **VIRTEL** möglich, d.h., dass für den Zugriff auf das EMA-System keine 3270-Emulation installiert werden muss. Der Zugriff ist über einen beliebigen Browser möglich.

Den Teilnehmern wird für die Dauer des Kurses auf der Lernplattform der EMA ein Kursbereich eingerichtet, über den Unterlagen zur Verfügung gestellt werden und in dem ein Forum eingerichtet wird, so dass jederzeit Fragen gestellt und Diskussionen geführt werden können.

Als Begleitmaterial werden Unterlagen und Folien der EMA eingesetzt. Ergänzend wird das eBook „**Assembler Language Programming for IBM System z Servers**“ in der Version 2.00 von John Ehrman eingesetzt. Außerdem natürlich die IBM Manuals (HLL Assembler Language Reference, HLL Assembler Programmer's Guide, Callable Services for HLL sowie z/Architecture Principles of Operation).

3 Inhalt

3.1 Modul Assembler inkl. Debugging (ca. 90 Stunden)

- Sitzungen im virtuellen Klassenzimmer
- eLearning
- theoretische Übungen
- praktische Übungen

Ziele

Die Teilnehmer kennen den Aufbau und die Struktur eines IBM Mainframes in Zusammenhang mit der Maschinensprache. Sie können einfache Assemblerprogramme schreiben und Programmfehler analysieren und beheben.

Inhalt

Einleitung

Einstieg in den Assembler, Source-, Object-, Lade-Modul. Sprachsyntax, Zahlensysteme, Codevereinbarungen, Maschinen-, Assembler, Macro-Instruktionen.

Aufbau eines Programmes, START, END, TITLE, EJECT, PRINT.

Datenfelddefinitionen DS und DC, ORG, EQU
Literele

Maschinen-Instruktionen

Befehlsformate, Logische Verarbeitung, MVC, MVI, CLC, CLI, ...

Condition Code, BC, BCR, JC

Das Programm, seine Daten und Adressen

Register und ihre speziellen Datenfelder

Befehle zur Registerverarbeitung

Unterprogrammtechnik, BAL, BALR, ST, L, LA.

Adressierungstechnik.

DUMP Analyse

Dezimalarithmetik

AP, SP, CP, MP, DP, SRP.

Techniken in alten Programmen, MVO, MVN, MVZ

Standard Dateiverarbeitung

Definieren einer Datei. Verarbeiten einer sequentiellen Datei. Job Control

Macros

Warum Macros?

Macros nutzen

Zugriff auf Control Blocks

Arbeiten mit Festpunktregistern

Register laden und speichern, Register
vergleichen, Tabellenverarbeitung,
Konvertieren Dezimal/Dual, Rechnen mit
Festpunkt-Registern, Adressrechnungen,
Indexrechnung

Programmunterteilung

Programmgestaltung, Segmentierung, CSECT,
DSECT, EQU.
Externe Unterprogrammtechnik, CALL, SAVE,
RETURN,
Basisregistertechnik, Konstruktion von
modularen Programmen,
Registerkonventionen, SAVE-AREA-Aufbau,
Parameterübergabe, extern und interne
Unterprogramme

Interactive Problem Control System (IPCS)

Funktionen
Konfiguration von IPCS

Problemdiagnose Überblick

Grundlagen der Problemdiagnose
Diagnoseschritte
z/OS Versionen und Releases
Problembereiche und -typen
Datenquellen für die Problemdiagnose

Konfiguration und Umgebung, Hardware

Dumpverarbeitung, z/OS Traces
SYS1.PARMLIB Diagnoseparameter
Canceln und „Dump ziehen“
Neue Hardware mit neuen Instruktionen