

Introduction to IBM zSystems Assembler incl. Debugging

for

- **System Programmers**
- **Application Developers**

based on the EMA Blended Learning Concept

Version 1.8: February 2023

Author(s): Wolfram Greis / Rolf Brüning

European Mainframe Academy GmbH
Am Klostergarten 3
D 78337 Öhningen
Tel. +49-7735-938 8668
education@mainframe-academy.de

European Mainframe Academy AG
Obergass 23
CH 8260 Stein am Rhein
Tel. +41-79-340 64 52
wolfram.greis@mainframe-academy.de

Table of Contents

1	Objectives of the Training Module	3
2	Prerequisites.....	3
3	Content	4
3.1	Assembler incl. Debugging (approx.. 80 hours)	Fehler! Textmarke nicht definiert.

1 Objectives of the Training Module

The aim of the module is to teach participants the machine-oriented programming language **IBM Assembler High Level Assembler (HLL)**. After this module, the participants can understand, interpret and adapt assembler programs and write their own programs in assembler.

Low-level programming in assembler requires that the participants deal with the architecture and the internal structures of the IBM z Systems architecture. Thus, teaching assembler is not only about programming, but also about understanding the architecture and how the various components of the architecture interact

2 Prerequisites

Knowledge of the basic logic for procedural programming is assumed. Also dealing with TSO / ISPF / JCL and SDSF.

For the duration of the training, the participants are provided with access to a current z/OS system from EMA. This access is also available after the end of the training. The EMA guarantees this access for at least three months after the end of the training.

Access is possible via **VIRTEL from SysperTec in France**, which means that no terminal 3270 emulation need to be installed to access the z/OS system. Access is possible via any browser.

For the duration of the course, participants will be provided with a course area on the EMA learning platform (Moodle LMS), through which documents will be made available and in which a forum will be set up so that questions can be asked, and discussions can be held at any time.

EMA documents and slides are used as accompanying material. In addition, the eBook "Assembler Language Programming for IBM System z Servers" in version 2.00 by John Ehrman is used. In addition, of course, the IBM manuals (HLL Assembler Language Reference, HLL Assembler Programmer's Guide and z/Architecture Principles of Operation).

3 Content

Goals

The participants know the design and structure of an IBM mainframe in connection with the machine language. They can write their own assembler programs and analyze and fix program errors.

Content

Introduction

Introduction to assembler language, source, object, load modules. Language syntax, number systems, code conventions, machine, assembler, macro instructions. Structure of a program, START, END, TITLE, EJECT, PRINT. Data field definitions DS and DC, ORG, EQU literals

Maschine-Instructions

Instruction format, logical processing, MVC, MVI, CLC, CLI, Condition Code, BC, BCR

Programs – Data - Addressing

Registers and their special data fields
Register processing instructions
Subprogram technique, BAL, BALR, ST, L, LA, addressing technique.
Dump analysis

Decimal Arithmetic

AP, SP, CP, MP, DP, SRP.
Techniques used in older programs, MVO, MVN, MVZ

Standard Dataset Processing

Definition of a file, processing sequential files, JCL

Macros

Why Macros?
Use of macros
Control block access

Working with fixed point registers

Load and store registers, compare registers, processing tables,
Convert decimal / dual,
address processing, index processing.

Program Segmentation

CSECT, DSECT, EQU
External subprogram technique, CALL, SAVE, RETURN
Base register, building modular programs, register conventions, save area structure, parameter transfer, external and internal subprograms

Interactive Problem Control System (IPCS)

Introduction and main functions
IPCS configuration

Problem Determination

Basics
Diagnose steps
z/OS versions and releases
Problem areas and problem types
Data sources for problem determination

Configuration and Environment

Dump processing
z/OS traces
SYS1.PARMLIB diagnosis parameters
Cancel and dump production