IBM Mainframe Assembler – Introduction

Duration

10 days

Participants

This course is designed for programmers already experienced in another procedural language such as COBOL. Non-programmers should first learn program logic. A knowledge of JCL is recommended.

Objectives

Upon successful completion of this course, you will be able to:

- Code, link and execute structured assembler programs using the fixed point binary instruction set, the packed decimal instruction set, and most of the non-privileged logical character instruction set.
- Read and interpret an assembler listing, and correct diagnostics.
- Debug program interrupt abends, such as S0C7, S0C9, S0C6, S0CB.
- Convert between symbolic and explicit machine instruction code, and interpret instructions in a hexadecimal dump.
- Use DS, DC (including C, X, F, H, D, P, Z, A, and V type constants), START, CSECT, DSECT, ENTRY, EXTRN, USING, DROP, ORG, LTORG, CNOP, and END.
- Format the source listing using TITLE, EJECT, SPACE, PRINT and comments.
- Code the I/ O macros (OPEN, CLOSE, GET, PUT, DCB) for QSAM FB MOVE mode.
- Call a sub-program, pass data and access that data in the sub-program, using standard entry linkage and exit linkage.

Format

Lecture and hands-on workshops.

Overview

Course topics include data definition, use of packed decimal and fixed point binary instruction sets, looping techniques, formatting and editing output reports, modular and self-documenting code, standard entry and exit linkage, writing main and sub-programs with standard linkage and data passing, DSECTs, macros for QSAM move mode I/ O, Boolean instructions for bit manipulation and testing, and debugging techniques. Students code and run many programs and sub-programs to reinforce the lecture material. This assembler is used on zSeries and OS/ 390 mainframe computers, running z/ OS or OS/ 390 MVS operating systems.

Topic Outline

Background

Numbering systems
    Hexadecimal, binary, decimal
Addressability
    Absolute addresses (24 bit and 31 bit)
    Relative addressing - base displacement concept
    Boundaries (H, F, D)

Data Definition

Character / packed decimal / fixed point binary / zoned decimal
DS and DC instructions
Constants - C, X, B, F, H, D, P, Z
Introduction to A and V address constants
Using the IBM Reference Summary Booklet
Redefining fields and records - ORG

Topic Outline

Computer Listing: Overview

- Assembler listing
- Linkage editor listing
- SYSUDUMP layout - control blocks, registers, unformatted dump
- Locating data in the dump

Assembler Fundamentals

- Machine instruction formats (RR, RS, RX, SS, SI, S)
- Operands - explicit code, symbolic addresses, literals
- Basic machine instruction set for fixed point binary instructions
  - Data conversion - PACK, CVB, CVD, UNPK, MVZ
  - Arithmetic - A, AH, AR, S, SH, SR, M, MH, MR, D, DR
  - Moving data - LM, L, LH, IC, STM, ST, STH, STC, MVC
  - Branching - B, BR
- Comments and structured coding considerations
- Basic QSAM macros
  - OPEN, GET, PUT, CLOSE, DCB
  - Using the Data Management Macro Instructions manual
- Basic assembler instruction set - CSECT, USING, END
- Base register - USING, LR
- Entry and exit linkage - concepts and code
- Basic CLG JCL
- Manuals
  - System/ 370 Reference Summary Booklet
  - S/ 370 Principles of Operation
  - Assembler Language
  - Assembler Programmer's Guide

Standard Instruction Set, Moves, Compares, Branches, Modularization

- Move instructions - MVI, MVC, MVZ
- Compare fixed point binary data - CR, C, CH
- Extended mnemonic branching instructions - BR, BH, etc.
- Structured programming
  - Self documented programs
  - Modular programming - BAL, BALR, BR (to return)
  - Formatted source listing - TITLE, EJECT, SPACE, PRINT

Decimal Instruction Set

- Arithmetic - AP, SP, MP, DP
- Moving data - ZAP
- Conditional Branching - CP
- Truncating and rounding techniques - SRP
- Editing reports - ED, EDMK

Introduction to Debugging

- Using the System Messages and System Codes manuals
- Debugging dumps - S0C7, S0CB, S0C6, S0C9

Topic Outline

Linking Multiple Programs

Address constants - A and V
Loading Addresses - LA (MVS/ XA considerations)
Linkage editor
Program linkage and CALL statements
DSECTs

Logical Instruction Set

And - turn bits off - NI, NC, NR, N
Or - turn bits on - OI, OC, OR, O
Exclusive or - flip bits and zero out fields— XI, XC, XR, X
Testing bits - TM
Conditional branch - extended mnemonic instructions

Looping and Introduction to Table Handling

Defining one dimensional tables
Addressing using RX instructions and indexing
Addressing using SS instructions
Controlling loops - BCT, BCTR, BXLE

Introduction to Advanced Techniques - Optional

Multiple base registers
Placement of literals - LTORG
Conditional no-ops - CNOP
Data shifting - SLA, SRA, SLDA, SRDA, SLL, SRL, SLDL, SRDL
Testing and validating data - TRT