Assembler Language Coding Workshop: Basic

Duration

5 days.

Participants

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

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 many 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, and Z type constants), CSECT, DSECT, 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, and describe PUT mode using DSECTs.

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, using standard entry and exit linkage, DSECTs, macros for QSAM move mode I/O, and debugging techniques. Students code and run many programs and sub-programs to reinforce the lecture material. This assembler is for zSeries mainframe computers, running z/OS or OS/390 MVS operating systems.

Course Sequence

After taking this course, developers wishing to be more proficient in assembler should take our other Assembler Language Coding Workshops.

Format

Lecture and hands-on workshops.

Prerequisites

Knowledge of program logic, either as a COBOL or PL/I programmer, or by completing our Structured Programming and Logic course. A working knowledge of TSO ISPF and JCL is required.

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)

Topic Outline

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

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
- Executing modules – BAS, BASR, BAL, BALR

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, code, and macros
Basic CLG JCL
Manuals
- Reference Summary Booklet
- Principles of Operation
- Assembler Language
- Assembler Programmer's Guide

Standard Instruction Set, Moves, Compares, Branches, Modularization

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

Topic Outline

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

Conditional Processing Instruction Set

Conditional branch - extended mnemonic instructions
Condition setting, CP, C, arithmetic, etc. instructions
Conditions and masks to choose conditions
BC – the underlying instruction

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
Translating data, byte by byte – TR
Testing and validating data – TRT