Assembler Language Programming – Intermediate

Duration

5 days

Participants

This course is for application and systems programmers that need to use intermediate level language facilities when using the IBM High Level Assembler.

Objectives

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

- Utilize Boolean logic and logical operations.
- Use advanced data manipulation instructions.
- Transfer control to other programs via CALL, LOAD, and LINK macros.
- Acquire and use additional memory via GETMAIN services and DSECTs.
- Read and write VSAM KSDSs (key-sequenced datasets).
- Code, assemble and linkedit assembler programs utilizing 31-bit addressing.
- Code, assemble and linkedit reentrant and reusable programs.
- Use system provided macros and system variables.
- Write assembler macros using the macro language, including MACRO, MEND, MEXIT, MNOTE, AGO, AIG, AIF, ANOP, etc.

Format

Lecture and hands-on workshops.

Prerequisites

Courses in Introduction to z/ OS, TSO/ ISPF, Basic MVS JCL, and Basic Assembler Programming, or equivalent experience.

Overview

This course teaches participants how to use the intermediate techniques available when programming HLASM (IBM's High Level Assembler).

Topic Outline

Logical Instructions, Bit Manipulation, and Testing

Bit Manipulation
AND
OR
XOR - EXCLUSIVE OR
TM – Test Under Mask
Branching After TM
IC - Insert Character
STC - Store Character

Advanced Data Manipulation Instructions

ICM – Insert (LOAD) Characters Under Mask
STCM – Store Characters Under Mask
CLM – Compare Logical Characters Under Mask
EX – Execute
MVCL – Move Characters Long
MVCL Tips
CLCL – Compare Logical Characters Long
MVO – Move with Offset

Topic Outline

System Macros

TIME
Extracting Month from TIME Macros
SNAP
ABEND Macro
WTO
WTOR Macro
WAIT Macro
DOM Macro
Above the Line, Below the Line, the Bar
Memory Types
Virtual Storage Management
VS Memory Management
Subpools
GETMAIN Macro
FREEMAIN Macro

Using VSAM Datasets

VSAM File Organization
KSDS
VSAM KSDS Example
IDCAMS Utility
IDCAMS – CREATE KSDS
IDCAMS – Load and Print KSDS
VSAM Macros – Building Control Blocks
VSAM Macro – Assessing Control Blocks
VSAM Macros – Data and Requests
ACB Macro
RPL Macro
VSAM Macro Connections
ACB Options
RPL Options
OPEN Macro
CLOSE Macro
GET, PUT, ERASE Macros
Some Macro Combinations
EXLST Macro
SHOWCB Macro
GET Move Mode vs. Locate Mode
Example of VSAM Update

Subroutines and Addressing

Eye Catchers
Linkage Instructions
BAL/ R vs. BAS/ R
Modular Code – Internal Subroutines
Invoking an External Subroutine
Static vs. Dynamic Calls
Preserving Registers
Save Areas
LTR and R15

Topic Outline

Subroutines and Addressing *(continued)*

Link
Parameter Passing Using Register 1
LINK, LTR, WTO Example
DSECTS with I/ O
Parms from JCL Example
Dynamic Calls
Call
Static Call – Linkage Editor and GO
LOAD, CALL DELETE
24-bit vs. 31-bit Addressing
LOAD
CALL
CALL – List Form
CALL – Execute Form
DELETE
ENTRY and EXTERN

Reentrant / Serially Reusable Programs

Concepts
Reentrant
JCL Parms
How to Code Reentrant

Writing User Macros and Assembler Instructions

Assembler Instructions
Macro
Defining a Macro
Comment on Prototype Statement
Macro Prototype Statement
MYADD – Macro ADD Example
Keyword Parameters
Local and Global Symbols in Macros
LCLx and GBLx Example
SETx Example
SETA
SETB
SETC
Macro Statements - Overview
AIF and AGO
EQUREGS Macro Example
Concatenation
FMOVE Macro
CMOVE Macro
&SYSNDX – Local System Symbolic Variable
Parameter Sublists
FADDER Macro Example
Storage Attributes
CNOP
System Variables