ASM01 S/390 Assembly Language For Programmers

Course Description This course provides students with the skills required to code, test and document programs using IBM Assembler language. The schedule allows for many practical exercises covering various problems of progressive complexity.

Who Should Attend Participants should be programmers, systems programmers or programmer/analysts who intend either, writing and maintaining IBM Assembler code, or who require deeper insight into the internals of other high level languages.

Pre-Requisites Previous programming knowledge is not necessary, but at least six months practical experience should have been gained in the IT applications environment. A working knowledge of TSO/ISPF, or similar would be of benefit during practical sessions but is not strictly necessary.

Duration 10 Days

Machine/Program Architecture

Computer Components
CPU
Storage (main / auxiliary)
Program Execution Overview
The Program Status Word

Introduction To The Assembler

The Assembly Process
Source / Object Code
The Assembler Listing

Defining Storage Areas and Constants

Data Types

Character
Bit
Packed Decimal
Binary Halfword/Fullword
Zoned Decimal
A-cons
V-cons

Registers

Types Of Register
Common Register Functions

Base & Displacement/Relocatability

The Csect Statement
The Using Statement
The Location Counter
The Org Statement

The Instruction Formats

SS1
SS2
SI
RX
RS
RR

The Instruction Set (see over)

Macros

TIME
ABEND
WTO
WTOR
WAIT

Sequential Record I/O

DCB
Open
Close
Get (move/locate)
Put (move/locate)

Dsects

The Using Statement Review
Locate Mode I/O
The Getmain Macro

Save Areas

Standard Register Usage
Save Area Format
SAVE
RETURN
User macros

The Call Macro

Parameters

Moving Instructions

MVC : Move Character
MVI : Move Immediate
MVN : Move Numeric
MVZ : Move Zones
MVO : Move With Offset
IC : Insert Character
ICM : Insert Character Under Mask
STC : Store Character
STCM: Store Character Under Mask
LR : Load Register
LA : Load Address
L : Load
LH : Load Halfword
LM : Load Multiple
ST : Store
STH : Store Halfword
STM : Store Multiple
ZAP : Zero & Add Packed

Comparing Instructions

CLC : Compare Logical Character
CLI : Compare Logical Immediate
C : Compare
CH : Compare Halfword
CR : Compare Register
CP : Compare Packed
TM : Test Under Mask
LTR : Load & Test Register
TRT : Translate & Test

Branching Instructions

Extended Mnemonics
BC : Branch On Condition
BCR : Branch On Condition Register
BAL : Branch & Link
BALR : Branch & Link Register
BAS : Branch & Save
BASR: Branch & Save Register
BCT : Branch On Count
BCTR: Branch On Count Register
BXLE : Branch On Index Low or Equal

Conversion Instructions

CVB : Convert To Binary
CVD : Convert To Decimal
Pack : Pack
UNPK: Unpack
ED : Edit
EDMK: Edit & Mark
TR : Translate

Arithmetic Instructions

AR : Add Register
A : Add
AH : Add Halfword
AP : Add Packed
SR : Subtract Register
S : Subtract
SH : Subtract Halfword
SP : Subtract Packed
MR : Multiply Register
M : Multiply
MH : Multiply Halfword
MP : Multiply Packed
DR : Divide Register
D : Divide
DP : Divide Packed

Boolean Instructions

Truth Tables
N : And
NI : And Immediate
NR : And Register
NC : And Character
O : Or
OI : Or Immediate
OR : Or Register
OC : Or Character
X : Exclusive Or
XI : Exclusive Or Immediate
XR : Exclusive Or Register
XC : Exclusive Or Character

Miscellaneous Instructions

SLL : Shift Left Logical
SLDL : Shift Left Double Logical
SRL : Shift Right Logical
SRDL : Shift Right Double logical
Algebraic Shifts
SVC : Supervisor Call
EX : Execute

Floating Point Instructions (optional)

Floating Point Registers
AE : Add Short Float
AER : Add Short Float Register
SE : Subtract Short Float
SER : Subtract Short Float Register
ME : Multiply Short Float
MER : Multiply Short Float Register
DE : Divide Short Float
DER : Divide Short Float Register
CE : Compare Short Float
CER : Compare Short Float Register
LE : Load Short Float
LER : Load Short Float Register
STE : Store Short Float