Using z/OS Assembler Training

BILGINC
IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

Duration: **5 Day**

Overview

This is the definitive Assembler course. It is suitable for all systems and application programmers who need to understand Assembler, either to install and maintain systems software or to maintain and amend application programs or packages written in Assembler. This course describes and explains how Assembler actually works, and teaches how to read, interpret and modify Assembler routines.

This course is also available for one-company, on-site presentations and for live presentation over the Internet, via the Virtual Classroom Environment service.

Prerequisites

Experience of using z/OS, including using TSO/ISPF and submitting batch jobs. For programmers it is an advantage to have a good knowledge of either a high level (e.g. COBOL or PL/I) or a procedural language (e.g. CLISTs or REXX).

What You Will Learn

- describe the format of machine instructions
- demonstrate an understanding of addressing and base registers
- explain how control sections and dummy sections are used
- describe how to use a working set of machine instructions
- explain the difference between machine and Assembler instructions
- define constants and storage
- code instructions which will perform: arithmetic, data manipulation, program control, file handling and table handling
- explain how condition codes and masks are used
- read dumps
- explain how module linkage works
- use BASSM and BSM instructions
- understand the concepts of 31 & 64 bit addressing and its effects on existing machine instructions.

Outline

Introduction

Programming Languages; the assembly process and program structure; Assembler directives; macro instructions and Assembler instructions; instruction formats; storage locations and addresses; machine instructions and symbolic statements; hexadecimal and the EBCDIC code.

Handling Character Data

Defining fields; Define Storage & Constants; Examples of DS & DC; Defining areas with zero repetition; Move instructions; Move Long instruction; Literals, constants and immediate data; Dummy Sections (DSECTs); Equates (EQU).

Using z/OS Assembler Training

BILGINÇ IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

Duration: **5 Day**

Comparisons, Branching & Structure

Comparison instructions; Branching Instructions; Branch Mnemonics (BC instruction); BCT/BCTR for looping; Program skeleton; Executable code; The location counter; Addressing; Register convention; Save area chaining; Relative addressing; Indexing; Declarative part; Instructions to the Assembler; Subroutines; Branch and Link Instructions; BASSM & BSM instructions; Branch Register instruction; BXH / BXLE For looping; Execute; Program linkage with BAKR; Linkage Stack; Linkage Stack operations.

Binary Instructions

Signed binary arithmetic instructions; Types of Binary Instruction; Loading and Storing; Load and Store Instructions; The Binary Add Instructions; Binary Subtraction Instructions; Binary comparisons; Convert to Binary; Convert to Decimal; Multiplying Fullwords; Binary Division; Bit shifting instructions.

Macros & Input/Output

Macros; File handling; Building a control block; The DCB; DCB tied to File; The DCB explained; DCB parameters explained; The OPEN macro; The GET macro; The PUT macro; The CLOSE macro; Printed output data sets; Clearing the print line.

Packed Decimal Arithmetic

Summary of 'Decimal Manipulating Instructions'; Summary of decimal instructions; Packed Decimal; Add Packed instruction; SP and ZAP instructions; Pack and Unpack instructions; Move Zone Instruction; Move Numeric Instruction; Decimal Condition Codes; Editing Packed Decimal Data; Constructing patterns; EDIT example; EDMK instruction; Packed Decimal multiplication; Packed Decimal division.

Translations and Bit Manipulations

Summary of bit manipulating instructions; 'OR'ing; 'AND'ing; Exclusive 'OR'ing; Testing the bits; Translate instruction; Translation Tables + 'ORG'; TRT- Translate and Test instruction.

MVS Error Reporting, Dumps & Binder

System error reporting; MVS dumps; Stand-Alone Dump (SADUMP); SVC Dumps; User ABEND Dumps; Generating a User ABEND Dump; System Generated ABEND dump; Snap dumps; Symptom dumps; Program Interruption Codes (PICs); Explanations of the Assembler program listings; The External Symbol Dictionary (ESD); The Source and Object program listing; The Relocation Dictionary; The Symbol and Literal cross-reference (HLASM); The Diagnostic Cross Reference and Summary (HLASM); Options Summary (HLASM); Dump labs.

24, 31 & 64 Bit Programming

Impact of 31-bit addressing; Addressing Memory - when and where?; Extended Addressing; 31-bit Virtual Addressing; Dynamic Address Translation; Addressing-mode control; Instruction differences; ; different addressing; Setting the mode when branching; Branch-And-Save-and-Set-Mode (BASSM); Branch-and-Set-Mode (BSM); Coding examples; The z/Architecture differences; z/Architecture - the bottom line; Tri-modal addressing mode; Memory boundaries; Registers in a z/Architecture machine; 64-bit registers and instructions; z/Architecture PSW; The 64-bit virtual address; Changing to 64-bit mode; Setting the mode when branching; Indicating 64-bit mode in pointers; Branch-And-Save-and-Set-Mode (BASSM); A z/OS address space; Characteristics of Memory Objects.

Using z/OS Assembler Training

BILGINÇ
IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

Duration: **5 Day**