Advanced Assembler and Macro Usage Eğitimi

Süre: **4 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

This four-day course is designed to build on the skills taught in the 'Using z/OS Assembler' course. It starts with a short review of the basic Assembler instructions before introducing more advanced areas such as general addressability, 31 and 64 bit addressing, re-entrancy, macro usage, debugging and recovery.
The course also covers the Assembler macro language fundamentals and z/OS Control Block usage, essential knowledge required when maintaining and writing z/OS exits.

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

Prerequisite for this course is attendance on **Using z/OS Assembler** course or equivalent experience. It is not mandatory but it would be helpful to have also attended the courses **z/OS System Fundamentals Part 1** & **z/OS System Part 2**.

What You Will Learn

- use advanced addressability techniques
- use data management macros
- receive and pass subroutine parameters and process return codes
- utilise the linkage stack
- apply 31 & 64 bit addressing concepts
- write re-entrant code
- use the basic Assembler macro language
- code Binder control statements
- use MVS control blocks in exit routines
- use supervisor services macros
- debug dumps
- write ESTAE routines
- write re-entrant programs.

Outline

Addressability

Control Sections - CSECTS; CSECT addressability - methods and examples; DROP directive; CSECTS in excess of 4K; Multiple Base Registers; Reference by explicit address; Reference by Dummy Section; DSECT mapping and external storage; MVS control block DSECTs; Explicit reference and internal control blocks; DSECT reference and internal control blocks; DSECT reference and external control blocks.

I/O and Data Management

Sequential files: Fixed Length Records, Variable Length Record; Data Control Block (DCB); QSAM DCB;

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

OPEN macro; CLOSE macro; Using the GET macro in Move mode; Using the PUT macro in Move mode; Using the GET macro in Locate mode; Using the PUT macro in Locate mode; Execute.

Intermodule Communication

Static Load Module structure and the VCON; Static Load Module linkage - 'CALL' macro; Secondary entry points; Establishing addressability for secondary entry points; External references using EXTRN; Save Area conventions; Standard Module linkage; SAVE and RETURN macros; Parameter passing standard convention; PARM keyword on JCL EXEC statement; Parameter passing using Explicit Interface; CALL macro: Explicit Parameter Interface, Implicit Parameter Interface, Implicit Variable Parameter Interface; Return code setting; Return code testing; Manipulating the Linkage Stack with BAKR and PR; Status saving using the Linkage Stack.

Extended Addressability

Virtual Storage; Establishing AMODE and RMODE values; AMODE and RMODE combination rules; Addressing mode sensitivity; Mode setting instructions - BASSM and BSM; Calling and returning using BASSM and BSM (Basic Mode); Mode switching to retrieve data from above 16 MB; Using pointer-defined linkage; BASSM & BSM in 64 bit mode; Using linkage assist routines; Using capping to assist linkage to 24-bit program; Using capping to assist linkage to 31-bit program.

Binder (Linkage editor)

Binder overview; Resolving external references; Object and LOAD module records; External Symbol Dictionary; ReLocation Dictionary; Automatic library call feature; Weak external reference; Binder JCL; Binder module attribute options; Binder module processing and output options; Binder control statements; Binder codes; AMBLIST utility.

Debugging

Problem program dump facilities; ABEND; SNAP; ABEND dump files; ABEND dump default contents; Summary dump; Symptom dump; ABEND macro; Parameter descriptions; SNAP macro; SNAP SDATA options; SNAP PDATA options; Save Area traces; Save Area trace and linkage stack entry.

System Services

Event synchronisation; WAIT macro; POST macro; write to log macro (WTL); write to operator macro (WTO); write to operator with reply macro (WTOR); delete operator message macro (DOM); TIME macro; STIMER macro; TTIMER macro; CPUTIMER macro.

Recovery

Recovery flow; ESTAE macro; System Diagnostic Work Area (SDWA); recovery routine processing; SETRP macro; retry options; lab exercise (write an ESTAE recovery routine).

Advanced Instructions

Move long (MVCL); compare logical long (CLCL); translate (TR); translate & test (TRT); edit (ED); edit & mark (EDMK).

Macro Language Fundamentals

Advanced Assembler and Macro Usage Eğitimi

Süre: **4 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Macro language introduction; macro language feature; defining a macro - basic rules; passing positional parameters; passing keyword parameters; parameter and keyword sublists; common system variables; testing how a field was defined; Lab exercise.

Task Management

LINK; LOAD ATTACH; XCTL.

Re-entrant & Re-usable Code

Re-entrant and re-usable code - objectives; re-usable code (REUS) ; re-entrant code (RENT) - the rules; RENT- how do you do it?; Re-entrant code example; List and Execute forms of macros; Getting the re-entrant benefit; Refreshable; Lab exercise (make a routine re-entrant).