ProTech *"Charting the Course ...*

protechtraining.com

... to Your Success!"

Developing Applications for z/OS UNIX

Course Summary

Description

Students who complete this course will be able to create and maintain programs written to run in the z/OS UNIX environment and that use callable z/OS UNIX System Services as well as C functions. They will also be able to construct and use makefiles to support their applications.

Objectives

After taking this course, students will be able to:

- Create programs written in COBOL, PL/I, C, or Assembler that:
  - are compiled and bound under z/OS batch or a z/OS UNIX shell
  - run in batch or from a shell script and that can ...
    - work with z/OS files and files in the Hierarchical File System (HFS)
    - interact with users at a z/OS UNIX terminal
    - dynamically call subroutines that are located in a z/OS library or an HFS directory
- Compile and bind C programs using c89, or Assemble and bind programs using c89, Assemble programs using the as command, or compile and bind COBOL programs using cob2 or compile and bind PL/I programs using pli
- Bind programs using the ld command
- Code programs in C, COBOL, PL/I or Assembler that invoke common C functions to accomplish work, when that is the best way to get the task done
- Code programs in C, COBOL, PL/I or Assembler that invoke common kernel services to accomplish work, when that is the best way to get the task done
- Code programs in C, COBOL, PL/I or Assembler that create, set, access, and update environment variables, and that access the parm data passed to a main program
- Build and use makefiles to manage an application.

Topics

- Introduction to z/OS UNIX applications
- File access in z/OS UNIX application
- Using QSAM to process HFS files
- Interacting with the user at the OMVS terminal
- Basic printf() and scanf() functions
- Calling C functions from COBOL programs
- Calling C functions from PL/I programs
- Calling C functions from Assembler programs
- Compiling and binding C programs from the shell: c89
- Assembling and binding Assembler programs from the shell: c89
- Assembling using the as command
- Compiling and binding COBOL programs from the shell: cob2
- Compiling and binding PL/I programs from the shell: pli
- Binding programs using the ld command
- Callable z/OS UNIX services (BPX1...)
- Callable LE services CEE3PR2, CEEENV, CEE3INF
- Parms and environment variables under the shell
- Make and makefiles

Developing Applications for z/OS UNIX

Course Summary (cont'd)

Audience

This course is designed for programmers experienced with working in a z/OS environment who will be designing and coding applications that are to be run using z/OS UNIX.

Prerequisites

Before taking this course, fundamental knowledge of z/OS UNIX shell scripting such as might be obtained from attending course PT5643: "Shell scripting in z/OS UNIX", and familiarity with at least one of these programming languages: COBOL, PL/I, C, LE-compliant Assembler is required.

Duration

Three days

ProTech "Charting the Course ...
protechtraining.com
... to Your Success!"

Developing Applications for z/OS UNIX

Course Outline

I. Introduction to the class
- A. Applications for z/OS UNIX
- B. Setting the stage: A Level Set
- C. Setting the stage: Skills to Acquire
- D. The Ubiquitousness of C
- E. Computer exercise: Class Lab Set Up

II. File Access in z/OS UNIX Applications
- A. What We Already Know
- B. C functions for accessing QSAM and HFS files
- C. COBOL - QSAM access
- D. COBOL - native access to HFS files
- E. PL/I - accessing QSAM and HFS files
- F. Assembler - accessing QSAM and HFS files
- G. Computer exercise: Accessing HFS files under OMVS

III. Interacting with the user at the OMVS terminal
- A. COBOL - Using DISPLAY and ACCEPT
- B. PL/I - Using PUT LIST and GET EDIT
- C. C printf() and scanf() functions - an introduction
- D. The CALL interface
- E. Function references
- F. Interpreting C language descriptions
- G. Computer exercise: Using printf() and scanf() - C programmers only

IV. Calling C functions from COBOL
- A. General notes
- B. fopen(), fread(), fwrite(), fclose(), printf(), scanf()
- C. Computer exercise: Using printf() and scanf() - COBOL programmers only

V. Calling C functions from PL/I
- A. General notes
- B. fopen(), fread(), fwrite(), fclose(), printf(), scanf()
- C. Computer exercise: Using printf() and scanf() - PL/I programmers only

VI. Calling C functions from Assembler
- A. General notes
- B. fopen(), fread(), fwrite(), fclose(), printf(), scanf()
- C. Computer exercise: Using printf() and scanf() - Assembler programmers only

VII. Compiling / Assembling, and binding Under OMVS
- A. Compiling and binding under OMVS
- B. Archive libraries
- C. Shell commands: ar
- D. C370LIBs
- E. Shell commands: c89
- F. Computer Exercise: Using c89 to compile and bind
- G. Computer Exercise: Using c89 to Assemble and bind

VIII. Assembling - a new alternative
- A. The as command
- B. Computer Exercise (Optional): Using as to Assemble a program

IX. Compiling COBOL and binding executables
- A. Shell commands: cob2
- B. Computer Exercise: Using cob2 to Compile and bind

X. Compiling PL/I and binding executables
- A. Shell commands: pli
- B. Computer Exercise: Using pli to Compile and bind

XI. Binding: the ld command
- A. Shell commands: ld
- B. Computer Exercise (Optional, for Assembler programmers): Bind a module

Developing Applications for z/OS UNIX

Course Outline (cont'd)

XII. Introduction to Callable UNIX services
- A. Dynamic calls
- B. Callable UNIX services
- C. The BPX1LOD service
- D. Assembler calling BPX1LOD
- E. COBOL calling BPX1LOD
- F. PL/I calling BPX1LOD
- G. C calling BPX1LOD
- H. A selection of callable services
- I. BPX1... services, concluded
- J. Computer Exercise: Dynamic calls

XIII. Parms and Environment Variables
- A. How the PARM field is set up
- B. Accessing the PARM field - Assembler
- C. Accessing the PARM field - COBOL
- D. Accessing the PARM field - PL/I
- E. Accessing the PARM field - C
- F. Accessing the PARM field using CEE3PRM and CEE3PR2
- G. Parms for subroutines
- H. The PARM set up under the shell
- I. Accessing the parm from a program run under the shell
- J. Determining the environment (CEE3INF)
- K. Using Environment Variables Under the Shell
- L. C functions clearenv(), getenv(), putenv(), setenv()
- M. Using the CEEENV callable LE service
- N. Computer Exercise: Working With Environment Variables

XIV. Managing Applications: Scripts and make
- A. Application management
- B. Using shell scripts for application management
- C. make - the big picture
- D. Introduction to makefiles
- E. Makefiles by example
- F. Computer Exercise: Basic Makefiles

XV. Archive files and make syntax
- A. Archive files and make
- B. Target attributes
- C. Designing makefiles
- D. Recursive make
- E. Include files for make
- F. Special target directives
- G. The syntax for make
- H. Computer Exercise: Combining makefiles

XVI. More on make
- A. Target lines: rule operators
- B. Runtime macros
- C. Command line prefixes
- D. Group recipes
- E. Special target directives, revisited
- F. Macro modifiers
- G. Conditionals
- H. Conclusion

XVII. Appendices
- A. Source programs
- B. HFS File I/O and C Functions
- C. Kernel functions for working with HFS files
- D. HFS File I/O and integrity
- E. Make and inference rules

XVIII. Index