C/UNIX® INTERFACE

General:

This course provides experienced C Language programmers with the ability to better utilize the C Programming Language in conjunction with the system calls and standard library functions of the UNIX operating system. The student is prepared to fully utilize the UNIX system calls and standard library functions to deal with processes and perform operations related to input and output.

Objectives:

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

- Make effective use of the C Language preprocessor.
- Use standard C Language library functions to perform operations related to error handling.
- Use UNIX system calls to perform operations related to file input and output.
- Use standard C Language library functions to perform operations related to file input and output, and to manipulate character strings.
- Use UNIX system calls to perform operations related to process creation, execution, and access.
- Use standard C Language library functions to perform operations related to process creation, execution, and access.
- Use UNIX system calls to perform interprocess communications via pipes.

Audience:

Experienced C Language Applications Programmers, and Systems Programmers.

Prerequisites:

At lease six (6) months of C Language programming experience. Some knowledge of the UNIX system tables related to processes and to the file system would also prove helpful.

Duration:

Five (5) days including classroom lecture and lab sessions.

PanTech
908-281-6759

C/UNIX INTERFACE
COURSE OUTLINE

I. C LANGUAGE PREPROCESSOR

A. Preprocessor Relationship to Compiler
B. Directive Syntax
C. Macro Substitution
   1. Token Replacement
   2. Macro Expansion
D. Include Files
E. Conditional Compilation
F. Typedefs

II. SYSTEM CALLS & ERROR HANDLING

A. System Calls vs. Library Routines
B. Error Codes
C. Error Variables
D. System Error Messages (`perror`)
E. Obtaining Core Dumps (`abort`)

III. I/O SYSTEM CALLS

A. UNIX File Types
B. File Definition
   1. System File Information (inode)
   2. File Permission Information
   3. File Descriptors
C. File I/O
   1. Opening and Closing Files (`open, close`)
   2. Reading and Writing Data (`read, write`)
   3. Creating New Files (`creat`)
   4. Positioning within Files (`lseek`)
   5. Linking and Removing Files (`link, unlink`)
D. Processing Directories
   1. Directory structure
   2. Directory Processing Routines
E. Affecting File Control Information
   1. Querying (`stat, access`)
   2. Default permissions (`umask`)
   3. Changing permissions (`chmod`)
   4. Setting Times (`utime`)
   5. Modifying Status (`chown`)
   6. Changing Directory (`chdir`)

IV. STANDARD LIBRARY

A. File Access
   1. `freopen`
   2. `fseek`
   3. `ftell`
   4. `fileno`
   5. `fdopen`
   6. `fflush`
B. Process Communication
   1. `popen`
   2. `pclose`
C. File Status
   1. `feof`
   2. `ferror`
   3. `clearerr`
D. Character Classification
   1. `isalpha`
   2. `islower`
   3. `isupper`
   4. `isdigit`
   5. `isspace`
E. Character Translation
   1. `toupper`
   2. `tolower`
F. String Comparison
   1. `strcmp`
   2. `strncmp`
G. String Handling
   1. `strlen`
   2. `strcpy`
   3. `strncpy`
H. String Manipulation
   1. `strchr`
   2. `strrchr`
   3. `strcat`
   4. `strncat`
   5. `strtok`

PanTech
908-281-6759

V. **PROCESS SYSTEM CALLS**

A. Creating a New Process (`fork`)
B. Executing a New Process
   1. `exec` Variations
   2. Inherited Attributes After `exec`
C. Process Termination (`exit, wait`)
D. Suspending Process Execution (`sleep`)
E. Controlling Process Information
   1. setpgrp
   2. getpid
   3. getppid
   4. getpgrp
F. Signaling a Process
   1. Sending a Signal (`kill`)
   2. Acting on a Signal (`signal`)
      a) Ignoring Signals
      b) Defaulting Signals
      c) Catching Signals

VI. **PROCESS LIBRARY**

A. Process Environment
   1. What is Process Environment
   2. Parts of Process Environment
   3. Obtaining Environment Information
B. Password Information
   1. What is Password File
   2. Obtaining Password File Information
C. Obtaining Time & Date Information
   1. `time`
   2. `ctime`
D. Processing Command Line Options (`getopt`)
E. Dynamic Storage Allocation
   1. `malloc`
   2. `realloc`
   3. `free`

VII. **PIPES**

A. UNIX Internal File Tables
   1. Table Descriptions
      a) File Descriptor Table
      b) System File Table
      c) System Inode Table
   2. Unrelated Processes
   3. Related Processes
B. Interprocess Communication
   1. System Calls
      a) `pipe`
      b) `dup`
      c) `fcntl`
   2. Self Communication
   3. Parent to Child Communication
   4. Child to Parent Communication
C. End of Course Case Study