Advanced UNIX Programming

This course provides in-depth training for software developers on UNIX system programming facilities. With an emphasis on writing portable programs using industry standards such as POSIX, X/Open, and the SUS, programming interfaces to several system services are explained in detail. Students will write and modify many C programs in this class, using system calls and library routines. This course can also be delivered on Linux.

Course Objectives:

- Develop the programming skills required to write applications that run on the UNIX operating system.
- Write portable applications using UNIX standards.
- Develop the basic skills required to write network programs using the Berkeley Sockets interface to the TCP/IP protocols.

Audience: Application developers who will be writing advanced programs on UNIX.

Prerequisites: *Fundamentals of UNIX and C Programming. Strong C programming skills are required for this course.*

Number of Days: 4 days

1. Course Introduction
- Course Objectives
- Overview
- Suggested References

2. UNIX Standards
- Brief History of UNIX
- AT&T and Berkeley UNIX Systems
- Major Vendors
- What is a Standard?
- What is POSIX?
- Other Industry Specs and Standards
- Library vs. System-Level Functions

3. Files and Directories
- Basic File Types
- File Descriptors
- The open() and creat() Functions
- Keeping Track of Open Files
- File Table Entries
- The v-node Structure
- The fcntl() Function
- The fcntl() Function – with F_DUPFD Command
- File Attributes
- The access() Function
- link(), unlink(), remove(), and rename() Functions
- Functions to Create, Remove, and Read Directories

4. System I/O
- Standard I/O vs system I/O
- System I/O Calls
- File and Record Locking

5. Processes
- What is a Process?
- Process Creation and Termination
- Process Memory Layout
- Dynamic Memory Allocation
- Accessing Environment Variables
- Real and Effective User IDs

6. Process Management
- The Difference Between Programs and Processes
- The fork() System Function
- Parent and Child
- The exec System Functions
- Current Image and New Image