

# Linux for the Systems Programmer

## Course No. 3303

Type: Hands-on

Duration: 40 hours

#### **Course Overview:**

This hands-on course is a follow on course to the Linux systems administration course. The course covers technical in-depth topics including system programming, file systems, signals, processes, pipes, threads, timers, input-output, sockets, the kernel and scripting.

#### Who should attend?

Linux systems programmers. The course is intended for programmers who are familiar with the C programming language and at least one other operating system.

#### **Prerequisites:**

C programming knowledge. An advantage to introduction To Linux or Linux Fundamentals or equivalent. Linux systems administration. Some basic experience in using Linux, Unix or another operating system.

### **Course Content:**

#### 1. Linux/Unix Overview

- History and philosophy of Unix/Linux and Open Source
- System architecture: from user interface to hardware
- Getting around: shell basics
- Overview: strace-ing "Hello World"

#### 2. System Programming

- Anatomy of a system call: uname()
- /proc your window to the kernel

#### 3. File Systems

- Overview of common file systems: ext2/3, nfs, reiserfs, xfs, vfat
- Kernel file system architecture, from block devices to files & directories
- File related system calls: stat, access, open, close, read, write
- Exercise
- More file related system calls: readv, writev
- Exercise

#### 4. Using Signals

- Overview of signals
- Typical usage
- Gotcha's traps and pitfalls
- Exercise

#### 5. Processes

- What is a process
- Process environment
- Working with processes: fork, exec\* and wait\*
- Exercise

#### 6. Pipes and IPC (inter-process communication)

- Pipe and dup2, popen and pclose system calls
- FIFOs (named pipes)
- Shared memory
- Sockets
- Semaphores
- Exercise

#### 7. Threads

- Time, gettimeofday system calls
- Alarm, setitimer
- Nanosleep
- Exercise

#### 8. Timers

- Using the Unix clock
- Internal clocks

#### 9. Socket programming

- TCP/IP overview
- The socket API
- Datagram v.s. connection-oriented sockets
- Typical client/server examples

#### 10. I/O

- File locking with fcntl
- Asynchronous I/O via select
- Exercise
- 11. The kernel
  - The kernel boot process
  - Building the kernel
  - An overview of module programming
  - Exercise

#### 12. Scripting

- Overview
- Variables
- Functions
- Examples
- Exercise

#### 13. Summary