

# Linux System Programming

## Course 102 – 40 Hours

### Overview

This course explores the system-level programming interfaces provided by Linux for developing applications and Embedded Linux Systems.

The course covers all the services provided by the C-Library to the application layer: accessing and handling files, creating processes and threads, IPC and many more.

### Course Objectives

- Understand the Linux programming model
- Create and manage processes and threads
- Using synchronization objects and IPC effectively
- Debugging applications

### Who Should Attend

The course is designed for developers who need to take their first steps in Linux development.

### Prerequisites

Students should have a working knowledge of the C programming language

### Course Contents

#### **Linux for Programmers**

- Linux standards and compatibility
- System calls
- Utilities and system limits

#### **Working with files**

- File descriptors
- Opening and closing files
- Reading and writing files
- File status and the inode
- Mapping files
- Creating directories
- Symbolic links
- Poll/select/epoll

### **Introduction to Processes**

- What is in a process?
- Process IDs and parent relationship
- Resource limits
- The process address space
- Virtual and physical memory
- The Linux scheduler
- Priorities

### **Posix Threads**

- Thread overview
- Creating and controlling threads
- Threads attributes
- Canceling threads
- Thread local storage

### **Synchronization Objects**

- Posix mutex
- Mutex attributes
- Priority inheritance
- Using reader-writer locks
- Using condition variables
- Avoiding deadlocks
- POSIX semaphores
- POSIX Shared memory

### **Pipes and FIFOs**

- Creating pipes
- Working with pipes
- Named pipes (FIFOs)
- Working with named pipes

### **Signals**

- What is a signal?
- Reacting to a signal
- Signal masks and signal sets
- Writing a signal handler
- Alarm, and interval timers

**Sockets**

- Domain and socket types
- Using datagram sockets
- Using stream sockets
- UNIX domain socket APIs
- Internet domain sockets
- Hosts, addresses, and ports
- Byte ordering
- IPv4 and IPv6
- Internet domain socket APIs
- Raw sockets
- Direct access to network device

**Application debugging**

- Pre-processing info
- Symbols, name mangling
- Debugging information
- ELF format and tools
- Using GDB
- Debugging with eclipse
- DDD
- Remote debugging
- GDB automation
- Debugging with core dumps
- Writing fault handlers
- Other tools for debugging, tracing and profile