



Course 609: Introduction to Red Hat Linux Programming (5 days)

Course Description...

In this course, C programmers learn how to use the APIs available under Linux. The course starts by exploring the tools of the Linux C programmer: gcc, GNU debugger, electric fence, version control, and making libraries. With the tools covered, the course quickly reviews file I/O, then moves to more advanced I/O topics such as multiplexing and memory mapping. Directory manipulation is then covered. The next four modules cover processes: the process model and creating child processes, signals and how to build signal handlers, IPCs, and threads complete this section. The course finishes up with a comprehensive section covering network programming. Here the course explains how to build the different kinds of network servers and daemons. The client side is fully covered. A few minor, but absolutely necessary skills, including building an RPM, manipulating time, and getting a random number, round out the course.

Suggested Prerequisites...

A working knowledge of C. If you are not comfortable with pointers and simple data structures in C you will need to review these topics before attending the class.

Who should attend...

C programmer who need to know how to program using the Linux programming interface.

Learning Objectives...

- Be able to get the most out of gcc
- Understand header files and the Linux header file system
- Be able to use Makefiles effectively
- Be able to work within a version control system
- Use and build static and dynamic libraries
- Effectively use debugging tools such as electric fence, GNU debugger
- Use signals, alarms and build signal handlers
- Build client server system with IPC: share memory, semaphore, message queues
- Understand and effectively use threads
- Build RPM for installing software
- Do memory mapping, and multiplexing I/O
- Manipulate directories and file globbing
- Know how to do socket, and RPC programming



Course Outline...

Introduction and Overview

Course Objectives

Unit 1: Shell Review

- Directory Commands
- Security Commands
- File Commands

Unit 2: The gcc Compiler

- Options
- Header Files
- In line functions

Unit 3: Development Tools

- Editors
- Makefiles
- Version Control

Unit 4: Debugging Tools

- gcc Options
- GNU Debugger
- Electric Fence
- Checker
- mpr and mcheck()

Unit 5: Creating and Using Libraries

- Static vs. Shared
- Using Libraries
- Library design: Compatibility
- Building Libraries

Unit 6: File Handling

- Simple I/O
- File Security
- I/O Mutiplexing
- Memory Mapping
- File Locking

Unit 7: Working with Directories

- Current Working Directory
- Changing the Root Directory
- Reading a Directory
- File Globbing
- Making and Removing Directories



Unit 8: Process Creation and Management

- Process Model
- Process Characteristics
- Simple Children
- fork and vfork
- Job Control

Unit 9: Signal Handlers

- The C Signal Model
- POSIX Signals
- Writing Signal Handlers
- Waiting for Signals

Unit 10: IPC

- Pipes and FIFO
- Semaphores
- Shared memory
- Messages Queues

Unit 11: Threads

- What are they?
- When to use them
- Thread Basics

Unit 12: Network Programming

- Socket Interface
- Introduction to TLI
- RPC Programming
- Making a daemon

Unit 13: Odds and Ends

- Building an RPM
- Working with Time
- Making Random Numbers
- Working with Strings and Regular Expressions
- Managing log files: logrotate
- Working with syslog and other system logs
- Automating tasks with at, cron, and anacron

Please contact your ROI representative to discuss course tailoring!!!