Developing Applications for Linux

Course LFD401 Four Days Instructor-led Hands-on

Introduction

Learn how to develop applications for the Linux environment. In this instructor-led course, you'll get hands-on experience with the necessary tools and methods for Linux application development and learn about the features and techniques that are unique to Linux.

This course is taught in partnership with the Linux Foundation.

At Course Completion

Upon successful completion of this course, students will gain the following skills and knowledge as you deploy, test, secure, manage, optimize, and troubleshoot a cloud solution.

- Preliminaries
- How to Work in OSS Projects
- Compilers
- Libraries
- Make
- Source Control
- Debugging and Core Dumps
- Debugging Tools
- System Calls
- Files and Filesystems in Linux
- File I/O
- Advanced File Operations
- Processes - I
- Processes - II
- Pipes and Fifos
- Asynchronous I/O
- Signals - I
- Signals - II
- POSIX Threads - I
- POSIX Threads - II
- Networking and Sockets
- Sockets - Addresses and Hosts
- Sockets - Ports and Protocols
- Sockets - Clients
- Sockets - Servers
- Sockets - Input/Output Operations

Contact ISInc for more information at 916.920.1700 or by visiting our website at http://www.isinc.com

- Sockets - Options
- Netlink Sockets
- Sockets - Multiplexing and Concurrent Servers
- Inter Process Communication
- Shared Memory
- Semaphores
- Message Queues

Prerequisites

To make the most of this course, students should:

Be proficient in the C programming language, basic Linux (UNIX) utilities such as ls, grep and tar, and be comfortable with any of the available text editors (e.g. emacs, vi, etc.) Experience with any major Linux distribution is helpful but not strictly required.

Student Materials

The student kit includes the following:

- Comprehensive digital workbook

Course Outline

Module 1: Introduction

- Objectives
- Who You Are
- The Linux Foundation
- Copyright and No Confidential Information
- Linux Foundation Training
- Certification Programs and Digital Badging
- Linux Distributions
- Platforms
- Preparing Your System
- Using and Downloading a Virtual Machine
- Things Change in Linux and Open Source Projects

Module 2: Preliminaries

- Procedures
- Standards and the LSB

Module 3: How to Work in OSS Projects **

- Overview on How to Contribute Properly
- Know Where the Code is Coming From: DCO and CLA

Contact ISInc for more information at 916.920.1700 or by visiting our website at http://www.isinc.com

- Stay Close to Mainline for Security and Quality
- Study and Understand the Project DNA
- Figure Out What Itch You Want to Scratch
- Identify Maintainers and Their Work Flows and Methods
- Get Early Input and Work in the Open
- Contribute Incremental Bits, Not Large Code Dumps
- Leave Your Ego at the Door: Don't Be Thin-Skinned
- Be Patient, Develop Long Term Relationships, Be Helpful

Module 4: Compilers

- GCC
- Other Compilers
- Major gcc Options
- Preprocessor
- Integrated Development Environments (IDE)
- Labs

Module 5: Libraries

- Static Libraries
- Shared Libraries
- Linking To Libraries
- Dynamic Linking Loader
- Labs

Module 6: Make

- Using make and Makefiles
- Building large projects
- More complicated rules
- Built-in rules
- Labs

Module 7: Source Control

- Source Control
- RCS and CVS
- Subversion
- git
- Labs

Module 8: Debugging and Core Dumps

- gdb
- What are Core Dump Files?
- Producing Core Dumps

Contact ISInc for more information at 916.920.1700 or by visiting our website at http://www.isinc.com

- Examining Core Dumps
- Labs

Module 9: Debugging Tools

- Getting the Time
- Profiling and Performance
- valgrind
- Address Sanitizer
- Labs

Module 10: System Calls

- System Calls vs. Library Functions
- How System Calls are Made
- Return Values and Error Numbers
- Labs

Module 11: Memory Management and Allocation

- Memory Management
- Dynamical Allocation
- Tuning malloc()
- Locking Pages
- Labs

Module 12: Files and Filesystems in Linux **

- Files, Directories and Devices
- The Virtual File System
- The ext2/ext3 Filesystem
- Journaling Filesystems
- The ext4/ Filesystem
- Labs

Module 13: File I/O

- UNIX File I/O
- Opening and Closing
- Reading, Writing and Seeking
- Positional and Vector I/O
- Standard I/O Library
- Large File Support (LFS)
- Labs

Module 14: Advanced File Operations

- Stat Functions

Contact ISInc for more information at 916.920.1700 or by visiting our website at http://www.isinc.com

- Directory Functions
- inotify
- Memory Mapping
- flock() and fcntl()
- Making Temporary Files
- Other System Calls
- Labs

Module 15: Processes - I

- What is a Process?
- Process Limits
- Process Groups
- The proc Filesystem
- Inter-Process Communication Methods
- Labs

Module 16: Processes - II

- Using system() to Create a Process
- Using fork() to Create a Process
- Using exec() to Create a Process
- Using clone()
- Exiting
- Constructors and Destructors
- Waiting
- Daemon Processes
- Labs

Module 17: Pipes and Fifos

- Pipes and Inter-Process Communication
- popen() and pclose()
- pipe()
- Named Pipes (FIFOs)
- splice(), vmsplice() and tee()
- Labs

Module 18: Asynchronous I/O**

- What is Asynchronous I/O?
- The POSIX Asynchronous I/O API
- Linux Implementation
- Labs

Module 19: Signals - I

- What are Signals?
- Signals Available
- Dispatching Signals
- Alarms, Pausing and Sleeping
- Setting up a Signal Handler
- Signal Sets
- sigaction()
- Labs

Module 20: Signals - II

- Reentrancy and Signal Handlers
- Jumping and Non-Local Returns
- siginfo and sigqueue()
- Real Time Signals
- Labs

Module 21: POSIX Threads - I

- Multi-threading under Linux
- Basic Program Structure
- Creating and Destroying Threads
- Signals and Threads
- Forking vs. Threading
- Labs

Module 22: POSIX Threads - II

- Deadlocks and Race Conditions
- Mutex Operations
- Semaphores
- Futexes
- Conditional Operations
- Labs

Module 23: Networking and Sockets

- Networking Layers
- What are Sockets?
- Stream Sockets
- Datagram Sockets
- Raw Sockets
- Byte Ordering
- Labs

Contact ISInc for more information at 916.920.1700 or by visiting our website at http://www.isinc.com

Module 24: Sockets - Addresses and Hosts

- Socket Address Structures
- Converting IP Addresses
- Host Information
- Labs

Module 25: Sockets - Ports and Protocols

- Service Port Information
- Protocol Information
- Labs

Module 26: Sockets - Clients

- Basic Client Sequence
- socket()
- connect()
- close() and shutdown()
- UNIX Client
- Internet Client
- Labs

Module 27: Sockets - Servers

- Basic Server Sequence
- bind()
- listen()
- accept()
- UNIX Server
- Internet Server
- Labs

Module 28: Sockets - Input/Output Operations

- write(), read()
- send(), recv()
- sendto(), recvfrom()
- sendmsg(), recvmsg()
- sendfile()
- socketpair()
- Labs

Module 29: Sockets - Options

- Getting and Setting Socket Options
- fcntl()

- ioctl()
- getsockopt() and setsockopt()
- Labs

Module 30: Netlink Sockets**

- What are netlink Sockets?
- Opening a netlink Socket
- netlink Messages
- Labs

Module 31: Sockets - Multiplexing and Concurrent Servers

- Multiplexed and Asynchronous Socket I/O
- select()
- poll()
- pselect() and ppoll()
- epoll
- Signal Driven and Asynchronous I/O
- Concurrent Servers
- Labs

Module 32: Inter Process Communication

- Methods of IPC
- POSIX IPC
- System V IPC**
- Labs

Module 33: Shared Memory

- What is Shared Memory?
- POSIX Shared Memory
- System V Shared Memory**
- Lab

Module 34: Semaphores

- What is a Semaphore?
- POSIX Semaphores
- System V Semaphores**
- Labs

Module 35: Message Queues

- What are Message Queues?
- POSIX Message Queues
- System V Message Queues**

Contact ISInc for more information at 916.920.1700 or by visiting our website at http://www.isinc.com

Contact ISInc for more information at 916.920.1700 or by visiting our website at http://www.isinc.com