Red Hat Enterprise Linux Development *(RHD251)*

Length: 5 Days

Summary: The Red Hat Enterprise Linux Development (RHD251) course rapidly trains programmers to develop applications and programs on Red Hat Enterprise Linux. Over the span of 5 days, students are provided hands-on training, concepts, and demonstrations with emphasis on realistic labs and programming exercises.

Prerequisites: Experience in C programming, shell scripting in a UNIX or Linux environment, experience with editors such as vi, emacs

Target Audience: Experienced C programmers who want to learn key skills for creating applications and programs on Red Hat Enterprise Linux. Microsoft Windows and UNIX programmers migrating their software to Linux.

COURSE CONTENT

1: GCC-GNU COMPILER COLLECTION

- GNU Compiler Collection (GCC)
- History of GCC
- Four Stages of GCC
- Interrupting the compiler
- Compiling a C program
- Preprocessor features
- Predefined preprocessor symbols
- Warnings and extensions
- Optimization
- Linking

2: BUILDING SOFTWARE WITH MAKE

- Introducing make
- How make works
- Makefile rule syntax
- Example: makefile first steps
- Makefile improved
- Implicit rules
- Example: simpler ls better makefile
- Variables
- Defining variables
- Example: makefile with variables
- Automatic variables
- Special targets
- Defining useful phony targets

3: THE GNU C LIBRARY AND SYSTEM CALLS

- Library goals
- Library standards
- GNU C library - glibc
- Library functions vs. system calls
- Using system calls
- Handling errors with errno
- Making sense of errno
- Using strace

4: PROGRAM ARGUMENTS AND ENVIRONMENT

- Program startup
- Using argc/argv
- Handling options with getopt()
- Handling options with getopt_long()
- Environment
- Manipulating the environment
- Program exit
- Registering exit handlers

5: BUILDING LIBRARIES

- Why use libraries?
- Static versus shared

- Static library benefits
- Shared library benefits
- Creating a static library
- Using static libraries

5: BUILDING LIBRARIES (CONT'D)

- Creating a shared library
- Using shared libraries
- Shared library management
- Library locations
- ldconfig

6: TIME FUNCTIONS

- When does time begin?
- Time data types
- Determining real time
- Converting time_t
- Converting tm structure
- Process time
- Time arithmetic
- Second resolution timers
- Fine-grained timers
- Real-time clock (RTC)

7: PROCESS MANAGEMENT

- What a process is
- Process relationships
- Create a child process
- Doing something else
- Related execve() functions
- Wait for a child
- More precise waiting
- Changing priority/nice
- Real-time priority

8: MEMORY OPERATIONS

- Allocating and freeing memory
- Memory alignment
- Locked memory
- Memory copy and initialization
- Memory comparison and search

9: DEBUGGING

- What is my program doing?
- Source-level debugging
- Invoking gdb
- Getting started with gdb
- Examining and changing memory
- Debuginfo libraries
- Using gdb with a running process
- Using gdb to autopsy a crash
- Debugging libraries - ElectricFence
- Debugging with valgrind
- Profiling for performance

10:BASIC FILE OPERATIONS

- Stream vs. system calls
- Opening and closing streams
- Stream input/output functions
- Stream status/errors
- Stream file positioning
- Stream buffering
- Temporary and scratch files
- Opening and closing file descriptors
- File descriptor I/O
- Repositioning file descriptors
- Stream/file descriptor conversions
- cat using ANSI I/O
- cat using POSIX I/O

11:COMMUNICATING WITH PIPES

- Introduction to pipes
- Standard I/O: popen()/pclose()
- Using popen()/pclose()
- System call: pipe()
- Using pipe()
- Named pipes
- Using named pipes
- For further reading

12:MANAGING SIGNALS

- What signals are
- Blocking and checking signals
- Working with signal sets
- Example of blocking signals
- Handling signals with sigaction()
- sigaction() example
- Handling signals with signal()
- Sending signals
- Real-time signals

13:PROGRAMMING WITH THREADS

- Introducing threaded programming

- Applications suited to threads
- Building threaded programs
- Creating threads
- Thread identity
- Synchronizing by joining
- Detaching threads
- Stopping threads
- Synchronizing with sutexes
- Using mutexes
- Read/ and write locks
- Conditional variables

13: PROGRAMMING WITH THREADS (CONT'D)

- Using conditional variables
- A conditional variable gotcha
- For further reading

14: ADVANCED FILE OPERATIONS

- Directory operations
- File system operations
- Multiplexed I/O with select()
- Miscellaneous I/O functions
- Memory mapped I/O
- Using memory mapped I/O
- File locking

15: INTERPROCESS COMMUNICATION (IPC)

- Interprocess communication (IPC)
- POSIX IPC overview
- POSIX shared memory
- POSIX semaphores
- POSIX message queues
- System V IPC overview
- System V IPC shared memory
- System V IPC semaphore arrays
- System V IPC message queues

16: BASIC NETWORK PROGRAMMING

- Linux networking overview
- Getting started with socket()
- Client functions
- Specifying IPv4 addresses
- Host versus network byte order
- Example TCP/IP client
- Address conversion functions
- Using getaddrinfo()
- Server functions
- Example TCP/IP server
- Datagram communication with UDP

17: WORKING WITH THE LINUX COMMUNITY

- Getting in touch with the community
- General considerations
- Building a community
- Licenses
- GPL
- LGPL
- BSD
- Creative commons