

Programming in the Linux Environment

Course Summary

Description

This course is available for the RedHat and SUSE distributions, but a variety of customized topics are available for other distributions as well (primarily Debian-based). Students are invited to bring their current ideas and questions to the classroom for discussion. Case studies, lecture, group problem solving, and online laboratories will be used. Students will be encouraged to enhance their skills utilizing the techniques presented through classroom problem solving and controlled online workshops.

Objectives

At the end of this course, students will be able to:

- Write application code in the C language that conforms to published Linux programming standards

Topics

- Overview of Programming on Linux (tools and techniques)
- Linux Programming Standards
- Command Line Tools
- Operating System APIs
- Packaging and Deployment

Audience

This course is designed for experienced C programmers who have worked on Unix/Linux systems in the past and are interested in developing or porting applications to the Linux environment.

Prerequisites

- Experience in C programming (prefer six months or more)
- Familiarity with Linux commands and directory structure
- Experience with the vi or emacs text editor. (Some classroom installations may also have pico or joe available.)
- Understanding of basic shell scripting for Unix/Linux

Duration

Five days

Programming in the Linux Environment

Course Outline

I. Overview of Programming on Linux (tools and techniques)

II. Linux Programming Standards

III. Command Line Tools

- A. Shell Scripting with a Focus on Programming Tasks
- B. Configuration Management and Version Control (RCS, CVS, *Subversion*)
- C. Overview of Compiling, Linking, and Debugging (gcc and ddd)
- D. Automating Builds Using make (includes an overview of imake)
- E. Overview of lex and yacc

IV. Operating System APIs

- A. Process Lifecycle and Management (including *capability* management)
- B. Virtual Filesystem API (both POSIX standard and Linux-specific)
- C. Filesystem Services (directory lookup, *FAM*, statistics)
- D. Multi-threaded Applications (including *futexes*)
- E. Signal Handling
- F. System V Interprocess Communication (shared memory, semaphores, message queues)
- G. Named and Unnamed Pipes
- H. Socket Programming (including multicasting)
- I. Asynchronous I/O
- J. Overview of the PTY Implementation

V. Packaging and Deployment

- A. Building Custom RPM or .deb Packages
- B. Converting Between Package Types Using alien
- C. Kernel Compilation (why and how)

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically.