

Bourne Shell Programming Course Summary

Description

This course is designed to help technical staff gain the essential skills needed to maintain, write, and debug UNIX Bourne shell scripts, along with key UNIX utilities such as grep, sed, and awk. Most existing UNIX shell programs are actually Bourne Shell scripts, since they are highly portable across diverse UNIX systems and can be run without modification by the Bourne, Korn and POSIX shells. Topics 2 (Review of UNIX Essentials) and 3 (Review of the vi editor) are covered on-demand if the attendees UNIX skills are rusty. This course contains hands-on programming labs to deepen the students understanding of presented concepts and facilities and to gain experience in program design, testing, and debugging.

Topics

- Introduction to Shell Programming
- Review of UNIX Essentials
- Review of the vi editor
- Using the Bourne Shell

Audience

This course is intended for technical staff that develops and maintains UNIX Bourne shell programs either for personal productivity, as an aid to application development, or for advanced UNIX system monitoring and maintenance.

Prerequisites

The student should have completed the ProTech UNIX Systems Basics I course or possess equivalent knowledge including an understanding of the UNIX file system, permissions, basic file management commands, use of the vi editor and familiarity with using a UNIX shell interactively.

Duration

Two days

Bourne Shell Programming

Course Outline

I. Intro to the UNIX Shell

- A. The UNIX Shell
- B. Key Shell Features
- C. Different Shell Flavors
- D. The Shell Prompt
- E. Intro to Shell Variables
- F. Creating & Using Shell Variables
- G. The Environment & Shell Variables
- H. Exporting Shell Variables
- I. Useful Shell Variables
- J. Understanding Shell Quote Usage
- K. Command-line Editing
- L. Shell Initialization: Startup Scripts
- M. Shell Initialization Variables: Lab
- N. Shell Scripts
- O. Shell Scripts - Example
- P. Languages: Compiled vs. Scripting
- Q. Viewing Exit Status in the Shell
- R. Using Exit Status in a Shell Script
- S. Shell: Redirection to & from Files
- T. Shell: Pipes
- U. Shell: Wildcards
- V. Shell: Command Alias
- W. UNIX Process Management
- X. UNIX Process Mgmt: the ps cmd
- Y. UNIX Process Mgmt: Background
- Z. Shell Job Control Commands
- AA. UNIX Process Mgmt: Kill
- BB. UNIX Process Management Lab

II. Review of UNIX Essentials

- A. Review of UNIX Essentials
- B. UNIX Command Overview
- C. Logging In
- D. Changing your Password
- E. Logging Out
- F. Commands, Switches, and Arguments
- G. Reading the Manual: the man cmd
- H. Userid, UID, Group, GID
- I. Understanding UNIX Permissions
- J. The UNIX File System
- K. Home and Working Directory
- L. Commands to List Contents of A File
- M. head and tail Commands
- N. Copying and Moving Files

- O. Deleting Files
- P. Summary Slide (cont.)
- Q. UNIX Filenames
- R. File Management Lab
- S. Creating Directories
- T. Removing Directories
- U. Copying Files Between Directories
- V. UNIX File Security: Permissions
- W. Working with Permissions
- X. chmod Examples
- Y. UNIX Directory Permissions
- Z. Lab: chmod
- AA. File/Dir Permissions - umask
- BB. Changing File Ownership
- CC. Under the covers of the File System
- DD. File Systems, inodes, vnodes
- EE. Linking files
- FF. Hard vs. Symbolic Links
- GG. Linking to a file - ln
- HH. UNIX Power Tools
- II. Finding Files
- JJ. UNIX Power Tools: sort
- KK. Archiving Files
- LL. Compressing files
- MM. TCP/IP Networking
- NN. TCP/IP Diagnostic Commands
- OO. TCP/IP Applications: rsh, rexec, ftp, mail
- PP. Example FTP Session
- QQ. TCP/IP Applications: write
- RR. TCP/IP Applications: wall
- SS. TCP/IP Applications: talk
- TT. Scheduling Work w/ cron & at

III. Review of the vi Editor

- A. UNIX Text Editors
- B. Editing Files with vi - Modes
- C. Editing Files with vi - Scrolling
- D. Editing Files with vi - Editing
- E. Editing Files with vi - File cmds
- F. Editing Files with vi - .exrc
- G. vi: Lab
- H. vi: Regular Expressions
- I. vi: Regular Expression Examples
- J. vi: Searching for Text
- K. vi: Search Lab
- L. vi: Text Substitution

M. vi: Text Substitution Lab

Bourne Shell Programming

Course Outline

IV. Using the Bourne Shell

- A. Why learn & Use the Bourne Shell?
- B. Bourne Shell Syntax Example
- C. Using Bourne Shell Variables
- D. Using Null Variable Error Messages
- E. Reading User Input
- F. Bourne Shell Arithmetic: expr, bc, and awk
- G. Command Line Arguments
- H. Condition Testing: if/then/elif/else
- I. Condition Testing: if/then/fi example
- J. Condition Testing: test command
- K. Compound Condition Testing
- L. Condition Testing: Optional Lab
- M. Conditional Execution
- N. Exiting from a script
- O. File Validation & Exit: Optional Lab
- P. Condition Testing: case/esac
- Q. Here Documents
- R. Looping Overview: for and while
- S. for Loop: Examples
- T. for Loop: Batch e-mail Example
- U. for Loop: Filename Expansion
- V. while Loop
- W. while Loop: Handling Parameters
- X. while Loop: Using getopts
- Y. Nested while Loops
- Z. Infinite Loops with while
- AA. Infinite Loops: Optional Lab
- BB. while Loops: Optional Lab
- CC. File I/O Overview
- DD. Shell Statement Redirection Syntax
- EE. Redirecting the Shell with exec
- FF. File I/O: Optional Lab
- GG. Multitasking in a Shell Program
- HH. Shell Functions
- II. Shell Functions: Usage Rules
- JJ. Trapping Signals in Shell Programs
- KK. Overview of Useful Signals
- LL. trap Command: Syntax
- MM. Signal Trapping to a Shell Function
- NN. Determining Signal Type & Status
- OO. Type Technique: read w/ Timeout