



Shell Programming for System Administrators

Duration: 5 days

Course Description

The Shell Programming for System Administrators course provides students with the skills to read, write, and debug UNIX shell scripts. The course begins by describing simple scripts to automate frequently executed commands and continues by describing conditional logic, user interaction, loops, menus, traps, and functions. This course is intended for system administrators who have mastered the basics of a UNIX Operating Environment (OE) such as the Solaris OE or Linux and who would like to read and understand the various boot scripts and write their own scripts to automate their day-to-day tasks. This course explores, in detail, the Bourne and Korn shell scripting languages.

Course topics:

UNIX Shells and Shell Scripts

- Describe the role of shells in the UNIX environment
- Describe the standard shells
- Define the components of a shell script
- Write a simple shell script

Writing and Debugging Scripts

- Start a script with #!
- Put comments in a script
- Change permissions on a script
- Execute a script
- Debug a script

The Shell Environment

- Use Bourne and Korn shell variables
- Assign values to shell variables



- Unset shell and environment variables
- Customize the user environment using the .profile file
- Perform arithmetic operations
- Create and use aliases
- Customize the Bourne and Korn shell environments
- Use the tilde expansion and command substitution features of the Korn shell

Regular Expressions and the grep Command

- Use and describe regular expressions
- Describe the grep command
- Use the grep command to find patterns in a file
- Use the regular expression characters with the grep command

The sed Editor

- Use the sed editor to perform noninteractive editing tasks
- Use regular expression characters with the sed command

The nawk Programming Language

- Use nawk commands from the command line
- Write simple nawk programs to generate data reports from text files
- Write simple nawk programs to generate numeric and text reports from text files

Conditionals

- Use the exit status of a command as conditional control
- Use the "if" statement to test a condition
- Pass values using command-line arguments (positional parameters) into a script
- Create USAGE messages
- Use conditional if, then, elif, else, and fi constructs
- Use exit, let, and test statements ([[]], " ")
- Apply the &&, ||, and ! Boolean logic operators
- Use the case statement



Interactive Scripts

- Use the print and echo commands to display text
- Use the read command to interactively assign data to a shell variable
- Read user input into one or more variables, using one read statement
- Use special characters, with print and echo, to make the displayed text more user friendly
- Create a "here" document
- Use file descriptors to read from and write to multiple files

Loops

- Write scripts that use for, while, and until loops
- Write a script using the select statement
- Describe when to use loops within a script
- Generate argument lists using command, variable, and file-name substitution

The getopt Statement

- Process script options using the getopt statement

Advanced Variables, Parameters, and Argument Lists

- Declare strings, integers, and array variables
- Manipulate string variables
- Change the values of the positional parameters using the set statement within a script
- Use Korn shell arrays
- Set default values for parameters
- Use the Korn shell built-in let, print, set, and typeset statements

Functions

- Create user-defined functions in a shell script
- Create, invoke, and display functions from the command line
- Pass arguments into a function
- Call functions from special (function) files that are saved in one or more function directories
- Describe where functions are available for use