Linux Bash and Korn Shell Programming

Course Summary

Length: 5 days **Prerequisite:** Linux/UNIX fundamentals course

Recommendation Statement:

You must be familiar with basic UNIX commands, and the VI editor. A Linux or UNIX fundamentals course is recommended. It is recommended that the student is proficient in fundamental Linux commands including, but not limited to: rm, cp, man, more, mkdir, ps and chmod.

The student should be able to create and edit text files in a text editor and have a basic understanding of processes and file permissions, and understanding of the relationship between parent and child processes, manipulating files and directories, basic variables, piping and redirection, and the find and grep commands.

Course Description:

A Linux Shell Scripting course that teaches the Linux user and administrator to create, interpret and debug Bash and Korn Shell scripts. You will learn how to assign environment variables, use shell special variables, arrays, argument lists, test conditions, utilize positional parameters and command substitution. You'll learn how to use shell builtins and conditionals to control your script creating loops, interactive programs, and functions. In addition to shell scripting, you will learn how to enhance your scripts by learning the intricacies of grep, sed, and awk utilizing metacharacters and regular expressions – beyond anything you've learned in a Linux fundamentals course. In this course you'll learn by doing, creating real world scripts that are relevant to current administrative tasks.

Upon completion of this course the student will be able to:

- Understand the various shells that are available in Linux and you'll be equipped to make an informed decision on which shell to use.
- Read, write, and debug Bash and Korn shell scripts
- Use local and environmental variables and shell metacharacters in scripts
- Use regular expression characters with the grep, sed, and awk utilities
- Write sed scripts to perform non-interactive editing tasks
- Write awk scripts to manipulate individual fields within a record and to write reports based upon an input file
- Use the exit status of a command to determine if the command succeeded or failed
- Access and process command-line arguments passed into a script
- Develop messages to display when a script is used incorrectly
- Use flow control constructs, such as branching and looping
- Perform string manipulation and integer arithmetic on shell variables

Linux Bash and Korn Shell Programming

Course Outline

UNIX Shells and Shell Scripts

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

Introduction to the POSIX shell

- Popular Shells: Bourne, Korn, Bash, csh, tcsh
- The POSIX shell
- Describe differences between Korn and Bash
- Bash and Korn shell initialization files

Writing and Debugging Scripts

- Creating a shell script!
- Put comments in a script
- Change permissions on a script
- Execute a script
- Debug a script

The Shell Environment

- Use shell variables
- Assign values to shell variables
- Display the value of shell variables
- Make variables available to subprocesses using the export statement
- Display the value of environment variables
- Unset shell and environment variables
- Customize the user shell environment using initialization files (Bash and Korn)
- Perform arithmetic operations
- Create and use aliases
- Display aliases and the values assigned to them
- Define the built-in aliases
- Customize the shell environment
- Use the tilde expansion and command substitution features of the 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
- Metacharacters
- The grep command

Stream Editing with 'sed'

- Use sed to perform noninteractive editing tasks
- Substituting text
- Deleting and printing lines
- Reading and writing files
- Multiple 'sed' editor functions
- Use regular expression characters with the sed command

The awk Programming Language

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

- String Manipulation
- String Comparison
- String Relations
- Concatenation
- Substring Manipulations
- Wildcards and Pattern Matching

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
- Place parameters on the command line
- Use conditional if/then constructs
- Use exit, let, and test statements ([[ ]], " ")
- Apply the &&, ||, and ! Boolean logic operators
- Using case and select constructs

Interactive Scripts

- Use the printf 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

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
- Pass values using command-line arguments (positional parameters) into a script
- Declare shell string and array variables
- Setting and retrieving arrays
- Set default values for parameters
- Use the shell built-in let, printf, set, and typeset statements
- Reading user input
- Positional parameters
- Creating positional parameters
- Conditional substitution
- Working with arithmetic operators and data

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

Controlling Script Behavior

- Manage the use of temporary files
- Write scripts using the setuid bit
- Use the crontab command

Scheduling Execution

- sleep
- date
- cron and crontab

Traps

- Describe how the trap statement works
- Include trap statements in a script
- Use the trap statement to catch signals and handle errors