

Shell Scripting :- 20 days – 20hrs

UNIX Shells and Shell Scripts

- Role of shells in the UNIX environment
- Standard shell
- 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

- Reviewing User Startup Script
- Shell User Environment
- A Review of Variables
- Special Shell Variable
- Creating Variables in the Shell
- Exporting Variables to Sub shells
- Reserved Variable
- Review of Quoting Characters
- Shell Command Substitution
- Korn Shell Tilde Expansion
- Arithmetic Operations on Bourne Shell Variables
- Arithmetic Precedence
- The Korn Shell let Statement
- Korn Shell Aliases
- Built-in Aliases
- Shell Parse Order

Use Bourne and Korn shell variables

- Assign values to shell variables

- Display the value of shell variables
- Make variables available to sub processes using the export statement
- Display the value of environment variables
- Unset shell and environment variables
- Customize the user environment using the .profile file
- Perform arithmetic operations
- Create and use aliases
- Display aliases and the values assigned to them
- Define the built-in 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
- 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 non interactive editing tasks
- 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 file
- Write simple awk 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
- Place parameters on the command line
- Use conditional if, then, else if, else, and if 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
- When to use loops within a script
- Generate argument lists using command, variable, and file-name substitution
- Select and get opts

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
- Where functions are available for use

Traps

- How the trap statement works
- Include trap statements in a script

- Use the trap statement to catch signals and handle errors