

HP-UX Posix Shell Programming

Course Outline

Course Introduction

- Administration and Course Materials
- Course Structure and Agenda
- Delegate and Trainer Introductions

Session 1: UNIX COMMAND REVIEW

- Basic Unix commands
- General commands
- File and directory handling commands
- Filename generation characters
- I/O Redirection features
- Other commands

Session 2: GETTING STARTED

- What is a shell script?
- Development guidelines
- Creating and editing shell scripts
- Naming and storing shell scripts
- Executing shell scripts
- Exercise: Write a simple shell script

Session 3: USING VARIABLES

- Environment variables
- Local variables
- Assigning values to variables
- Assessing variable values
- Using quotes
- Delimiting variable names
- Echo control sequences
- Exercise: Add variables to a script

Session 4: INTEGER ARITHMETIC

- Using the `expr` command
- Using the `(())` notation
- Exercises: Add integer arithmetic to a shell script

Session 5: HANDLING RUN TIME DATA

- The `read` command
- Command line arguments
- Exercise: Writing a generic shell script
- Exercise: Writing an interactive shell script

Session 6: CONDITIONAL EXECUTION

- The `if` statement
- The `test` command
- Exercise: Adding validation to previous scripts

Session 7: ADDITIONAL KORN, BASH & POSIX SYNTAX

- Other test notations
- Default and substitute variables
- Exit status codes
- Exercise

Session 8: LOOP CONSTRUCTS

- The `while` loop
- The `until` loop
- The `for` loop
- The `while true` and `until false` loops
- Loop control commands
- Exercise: Enhancing the previously written scripts
- Exercise: Writing a script to copy files using a `'for'` loop
- Exercise: Writing a script to generate numbers with the `'while'` loop

Session 9: MULTI-BRANCH DECISIONS

- The `case` statement
- Menu driven applications
- Exercise: Developing and writing a menu system

Session 10: FUNCTIONS

- What is a function?
- Syntax
- Examples
- Creating a Function Library
- Exercise: Add a function to a script

Session 11: INTERRUPT HANDLING

- Interrupt signals
- Trapping interrupts
- Exercise: Adding traps to the menu script

Session 12: ADDITIONAL FEATURES AND FACILITIES

- The exec commands
- The includes notation
- More about loops
- Arrays
- Here Documents
- Exercise: Create a here script

Session 13: BACKUP AND RESTORE UTILITIES

- Backing-up and restoring files
- Basic and advanced use of tar
- Compression utilities gzip, bzip2, zip and compress
- Exercise: Backing up and restoring files using tar
- Exercises: Compressing files

Session 14: BACKGROUND JOB SCHEDULING

- Scheduling jobs with the cron command
- Scheduling jobs with the at command
- Exercises: Running background jobs

Session 15: COMMANDS FOR COMPARING FILES

- Compare two files with the cmp command
- Compare two files with the comm command
- Compare two files with the diff and sdiff commands

- Compare large files with the bdiff command
- Exercises: Identifying file differences

Session 16: SPLITTING FILES

- The split and csplit commands
- Exercises: Splitting files

Session 17: IDENTIFYING AND TRANSLATING CHARACTERS

- od - octal dump
- Use cat to display non-printing characters
- The expand and unexpand commands to convert between tab and space characters
- The tr command for character translation
- Exercises: Translating characters with tr

Session 18: REGULAR EXPRESSION NOTATION REVIEW

- Standard regular expressions
- Extended regular expressions

Session 19: THE STREAM EDITOR sed

- sed command line syntax
- sed script files
- sed command processing
- sed addresses and simple instructions
- sed pattern space and hold space
- Grouping sed commands
- Hold and get functions
- Advanced flow control
- Exercises: Text processing with sed

Session 20: FUNDAMENTALS OF AWK

- Basic AWK usage
- AWK program-files
- AWK scripts
- AWK variables
- Pattern matching with AWK
- AWK extended patterns

- AWK operators
- AWK arithmetic operations
- AWK output
- Formatting output with printf
- Exercises: Create awk scripts to extract selected data from a file and generate reports

Session 21: AWK PROGRAM CONTROL STRUCTURES

- The BEGIN and END functions
- The AWK if construct
- The AWK else if construct
- The AWK while construct
- Other program control statements
- The AWK break, continue and exit statements
- User defined functions
- Exercises: Create AWK scripts and program-files utilising program control structures

Session 22: AWK FUNCTIONS

- AWK string functions
- AWK length, tolower, toupper, index, sub, gsub, match, substr, split, sprintf, system and getline functions
- Exercises: Generate AWK scripts and program-files to extract and format data using AWK functions

Session 23: AWK ARRAYS

- AWK associative arrays
- Multi-dimensional arrays
- Exercises: Create AWK associative arrays to process text files and generate reports

Session 24: MISCELLANEOUS TOOLS

- bc (calculator)
- fuser (testing for files in use)
- getops (checking options passed to shell scripts)
- printf (formatting screen output)
- logger (script logging)
- xargs (generating arguments for a command)
- eval (re-evaluating variables)

- Exercises: Using tools within a shell script