



Shell Programming

Five days

Course Description: This course provides an understanding of the command and programming language capabilities of the standard Shell. It also explains commands particularly useful in conjunction with Shell programs, as well as the programming constructs available within Shell. The student is prepared to design, code, test debug, and execute structured Shell procedures ("shell scripts") making full use of the basic, as well as the more advanced features of the Shell.

Every section contains hands-on labs and solutions.

Audience: The course is for programmers who require a general knowledge of Shell Programming.

Prerequisites: An Introduction to UNIX or Linux course or equivalent experience is required

Outline

INTRODUCTION

- What you should already know
- What is the shell?
- The role of the shell
- Determining your shell type
- What is a shell program?
- Creating a shell script
- Comments
- The subshell type descriptor record
- Basic decision making: if blocks
- The opposite of if: else blocks
- Contracted nesting: elif
- The test command
- The test operators
- Short circuit operators
- The exit command
- Command substitution: `command`
- Simple math: expr and \$(())

I/O AND VARIABLES

- Using echo
- Using print and printf
- Variables: overview
- Variable assignment
- Variable substitution
- Variable "dump": set
- Variable assignment: read
- Removing a variable: unset
- Exporting variables: export
- "Sourcing" a script with the dot (.)
- The backslash as a quote: \
- Single quotes
- Double quotes

WHILE LOOPS

- Overview
- The while loop
- The until loop
- Loop control: continue and break
- Nested loops as branch points

FOR LOOPS AND POSITIONAL PARAMETERS

- Overview
- The for loop
- Positional parameters
- Accessing positional parameters: shift
- \$* versus \$@
- Loading up the positional parameters
- The for loop without a list
- Examples of for loops

CONDITIONAL BRANCHING

- Overview
- Return values

DEBUGGING

- Overview

Shell Programming

- Execution trace: set -x
- Verbose trace: set -v
- Combining -x and -v
- Error exit: set -e
- No execution: -n
- No unset: set -u
- Temporarily "commenting out" code
- Removing the # characters
- Temporary exits and breakpoints
- Bypasses

THE CASE AND SELECT STATEMENTS

- The case statement
- The patterns
- A case example
- The select statement
- A select example
- LINES and COLUMNS

THE HERE DOCUMENT

- Overview
- The 'here' document: <<
- Allowing indentation with tabs
- Preventing substitutions
- Self-extracting scripts

SPECIAL VARIABLE SUBSTITUTIONS

- Overview
- Default substitute value
- Assign default value
- Print error and exit
- Alternate value substitution
- Pattern substring deletion
- Variable value length

ARRAYS AND TYPESET

- Overview
- The typeset built in
- Typeset options
- Examples of typeset
- Removing attributes with typeset
- Indexed arrays
- Array example
- The IFS variable

SIGNALS AND TRAPS

- Keyboard "interrupts"
- Signals, PID's and kill
- Terminating a process: kill
- Protecting your scripts: trap
- Setting an ignore trap
- Setting a catch trap
- Setting a catch trap for the exit signal
- A trap within a trap

FUNCTIONS

- Overview
- Creating a function
- Function definition files
- Removing functions
- Autoloading functions
- Variable scope in functions
- Positional parameters and functions

INTRODUCTION TO SED

- Automating changes: sed
- How sed works
- sed commands
- sed command examples
- sed example
- sed example walk through
- The -n option

MORE SED

- Overview
- Reading in a file: r
- Writing to a file: w
- Quitting: q
- Grouped commands: { }
- Multiline read next: N
- Hold and get: h H g G
- Do not execute: !
- Comments in a sed script: #

INTRODUCTION TO AWK

- A programming language: awk
- Command line syntax
- How awk works
- Patterns and actions
- Old, new or GNU?

Shell Programming

- Records, fields and special variables
- Simple actions
- Simple patterns
- Regular expressions as patterns
- More about printing
- Literal values
- Special patterns: BEGIN and END
- Better output format: printf
- User defined variables
- Operators
- The END pattern

MORE AWK

- Overview
- Conditional logic
- The while loop
- The for loop
- awk functions
- String manipulation
- Other awk things not covered here

ADVANCED I/O

- Overview
- exec and file descriptors 3 through 9
- Co-processes
- Named pipes

COMMAND LINE PROCESSING: GETOPTS

- Overview: getopt
- The Option String
- Examples
- Bad options
- Sample Script

APPENDIX

- What you should know
- vi editor summary
- Regular expression metacharacter summary
- Finding patterns with grep
- Extracting fields with cut