

## Linux Bash Shell Programming

Duration: 5 Days    Course Code: LBSP

---

### Overview:

#### Linux Bash Shell Programming Course Overview

The Linux Bash Shell Programming course is designed to give delegates practical experience in developing and writing Bash shell scripts. Most of the built-in shell commands are introduced together with the main program control structures. The course also gives practical experience using a range of Linux tools to manipulate text and incorporate into Linux Bash shell scripts. Exercises and examples are used throughout the course to give practical hands-on experience with the techniques covered.

---

### Target Audience:

#### Who will the Course Benefit?

Programmers, developers and system administrators who need to construct shell scripts and process text files using advanced text handling facilities.

The Linux Bash Shell Programming course assumes knowledge of the Linux Operating System to the level covered in the Linux Introduction Course. Some programming experience may also prove advantageous.

---

### Objectives:

- Course Objectives
  - To provide the skills needed to develop and customise shell programs and to make effective use of a wide range of standard Linux programming and development tools.
- 

### Prerequisites:

- The Linux Bash Shell Programming course assumes knowledge of the Linux Operating System to the level covered in the Linux Introduction course. Some programming experience may also prove advantageous.
- 

### Follow-on-Courses:

#### Further Learning

- Linux System Administration (LPI)
  - Oracle SQL
-

## Content:

### Linux Bash Shell Programming Training Course Course Contents - DAY 1

#### 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

#### 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 Linux Bash Shell Programming Training Course Course Contents - DAY 3

#### 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

#### Session 17: IDENTIFYING AND TRANSLATING CHARACTERS

- od - octal dump
- Use cat to display non-printing characters
- View and format files with nl
- 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
- Searching with grep
- Metacharacters, positional characters and quantifiers
- Extended regular expressions
- POSIX character classes
- PERL 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
- Write output to temporary files
- 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

#### Session 21: AWK OPERATORS

- 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 Linux Bash Shell Programming Training Course Course Contents - DAY 5

#### Session 22: 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

- Other test notations
- Default and substitute variables
- Exit status codes
- Exercise Linux Bash Shell Programming Training Course Course Contents - DAY 2

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

#### Session 15: COMMANDS FOR COMPARING FILES

- Compare files with the cmp command
- Compare and format files with pr
- Compare files with the comm command
- Compare files with the diff and sdiff commands
- Compare large files with the bdiff command
- Exercises: Identifying file differences

#### Session 16: SPLITTING FILES

- The fold command
- Split files using context and content rules
- Exercises: Splitting files Linux Bash Shell Programming Training Course Course Contents - DAY 4

- The AWK break,continue and exit statements
- User defined functions
- Exercises: Create AWK scripts and program-files utilising program control structures

#### Session 23: 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 24: AWK ARRAYS

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

#### Session 25: 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

### Further Information:

For More information, or to book your course, please call us on 0800/84.009

[info@globalknowledge.be](mailto:info@globalknowledge.be)

[www.globalknowledge.com/en-be/](http://www.globalknowledge.com/en-be/)