Linux Bash Shell Programming

Category: Infrastructure & IT Operations | **Vendor:** Linux Professional Institute

Duration: 40.00 hours (5 days) **32.5 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Online

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.

About This Course

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.

Who Should Attend

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

Prerequisites & Entry Requirements

General Prerequisites:

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

Learning Outcomes

Upon successful completion of this course, participants will be able to:

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.

The delegate will learn and acquire skills as follows:

- Writing simple scripts to enhance basic command output
- Using the various shell quoting mechanisms appropriately
- Manipulating shell variables and user-defined variables in scripts
- Implementing conditional execution facilities
- Using the shell's built-in loop constructs where appropriate
- Writing scripts to trap user interrupts
- User defined Functions
- Developing menu-driven shell scripts
- Backing up and restoring files with tar
- Advanced tar commands
- File compression
- Scheduling background jobs with crontab and at
- Comparing file contents
- Splitting files based on content and context
- Identifying and translating characters
- Use og grep and regular expressions
- Basic and advanced file editing with sed
- Creating AWK program-files and scripts
- Pattern matching with AWK
- AWK variables, operators and arithmetic functions
- AWK program control structures
- AWK functions
- AWK associative and multi-dimensional arrays
- Various miscellaneous commands

Detailed Course Outline

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

- Other test notations

- Default and substitute variables

- Exit status codes

- Exercise

Day 2

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

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
- 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

Day 4

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

Day 5

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

Skills You Will Learn:

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.

The delegate will learn and acquire skills as follows:

- Writing simple scripts to enhance basic command output
- Using the various shell quoting mechanisms appropriately
- Manipulating shell variables and user-defined variables in scripts
- Implementing conditional execution facilities
- Using the shell's built-in loop constructs where appropriate
- Writing scripts to trap user interrupts
- User defined Functions
- Developing menu-driven shell scripts
- Backing up and restoring files with tar
- Advanced tar commands
- File compression
- Scheduling background jobs with crontab and at
- Comparing file contents
- Splitting files based on content and context
- Identifying and translating characters
- Use og grep and regular expressions
- Basic and advanced file editing with sed
- Creating AWK program-files and scripts
- Pattern matching with AWK
- AWK variables, operators and arithmetic functions
- AWK program control structures
- AWK functions
- AWK associative and multi-dimensional arrays
- Various miscellaneous commands

Frequently Asked Questions

Q: What delivery options are available for Linux Bash Shell Programming?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: How many CPD hours does this course provide?

The 5-day Linux Bash Shell Programming course provides up to 32.5 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the Linux Bash Shell Programming training?

The training takes place over 5 day(s), with each day lasting approximately 40.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for Linux Bash Shell Programming?

Yes, we provide corporate training, dedicated training, and closed classes for Linux Bash Shell Programming. Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: Linux Bash Shell Programming, Linux, LBSP

Q: Why choose Nexus Human for Linux Bash Shell Programming?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your Linux Bash Shell Programming training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 889 8700 (Ireland) | +44 20 8142 8961 (UK)

© 2026 Nexus Human. All rights reserved. This brochure was generated on 14/11/2026.