

## Linux<sup>®</sup>: A Comprehensive Hands-On Introduction - 4 Days

*Course 143 Overview*

- You Will Learn How To**
- Manage, control and automate Linux GNU open source tools
  - Create, edit and search Linux files and directories
  - Limit access within the file system by controlling permissions and ownership
  - Combine GNU filter commands in pipelines to process and format text data
  - Exploit Bash shell features to enhance the command line interface
  - Perform multiple tasks in shell scripts

**Course Benefits** The Linux open source operating system offers a wide range of graphical and command line tools that can be used to implement a high-performance, stable and inexpensive server. Throughout this course, you gain the essential knowledge and hands-on skills to leverage Linux for your organizational advantage. You learn to create, edit and search Linux files, control permissions and ownership, process and format text data, and use shell scripts to perform multiple tasks.

**Who Should Attend** Those interested in gaining the fundamental knowledge necessary to work with Linux. Basic computer knowledge is assumed.

**Hands-On Training** Throughout this course, you gain hands-on experience with the Linux operating system and GNU tools using Red Hat Enterprise Linux. Exercises include:

- Building, linking and removing Linux files and directories
- Setting and testing file permissions
- Customizing start-up scripts for enhanced Bash interactivity
- Running shell scripts for automation
- Processing, formatting and searching for text in files

# Linux<sup>®</sup>: A Comprehensive Hands-On Introduction - 4 Days

## Course 143 Outline

### Introducing Linux

- The UNIX heritage
- Linux inception
- Linux kernel and GNU tools
- Open source licensing
- Distributions

### Accessing the System

#### The GNOME desktop

- Customizing panels, launchers and applets
- Examining graphical applications
- Personalizing the terminal window

### Starting at the command line

- Switching to console logins
- Performing a SSH login
- Structuring commands

### Managing Files and Directories

#### Naming files and directories

- Contrasting full and relative pathnames
- Unraveling the file system hierarchy
- Handling files with **cp** and **mv**

#### Organizing files under directories

- Making and navigating directories
- Listing attributes with **ls**
- Browsing with GNOME Nautilus

#### Working with Linux files

- Accelerating command line usage with Bash wildcards
- Scrolling through files with GNU **less**
- Comparing files with **diff**

### Controlling Access to Linux Resources

#### Defining access rights to files

- Identifying multiple users and groups
- Interpreting file and directory modes
- Adjusting access permissions: **chmod**

#### Collaborating via group membership

- Joining secondary groups
- Inheriting and changing group ownership

#### Adopting multiple roles

- Knowing who you are to the system
- Switching identity
- Changing passwords
- Raising privilege with **su**, **sudo** and **setuid**

#### Searching the system

- Locating files with **find**

- Finding pathnames with **slocate**

- Changing the date and time

### Manipulating streams

- Matching lines with GNU **grep**
- Selecting lines and fields using **head**, **tail**, **gawk** and **cut**
- Saving command output into files
- Connecting commands using pipes

### Editing files and streams

- Automating stream edits with **sed**
- Creating and modifying files: **vim**, **gedit**

### Leveraging Bash Shell Features

#### Customizing Bash behavior

- Setting options: **noclobber**, **ignoreeof**
- Assigning to built-in shell variables
- Aliasing commands

#### Initializing context

- Exporting variables to the environment
- Extending login and start-up scripts

#### Enhancing interactivity

- Retrieving and reusing previous commands
- Exploiting file name completion shortcuts

### Automating Tasks with Shell Scripts

#### Invoking shell scripts

- Taking **bash** input from a file
- Calling scripts as commands
- Running scripts using **source**
- Passing positional parameters

#### Testing and controlling execution

- Checking exit status with **if**
- Verifying file attributes with conditionals

### Executing Jobs and Processes

#### Monitoring processes with **ps** and **top**

- Launching multiple jobs
- Signaling with **kill**

#### Archiving and retrieving data

- Mounting storage devices
- Measuring free space
- Compressing with **bzip** and **gzip**
- Creating **tar** archives

#### System administration basics

- Installing Linux software
- Adding user accounts
- Adjusting network connections