Linux Fundamentals

Linux is used by many large websites - Google, Facebook, Twitter, Yahoo, Amazon, YouTube, wordpress.com, flickr. Have you wondered why? Learn the fundamentals of how to harness Linux's power in this engaging course, which is intended for those who want to get the most out of using Linux.

I place particular emphasis on the simple yet general concepts underlying Linux. These concepts enable you to do very powerful things. You will learn:

1. how Linux's very simple structure gives you great flexibility
2. how to hook commands together as 'building blocks' to make more powerful commands
3. the common features the 'shell' gives you that can be used with any command
4. how to process files and text

Approach

The course is part lecture, part hands-on.

Prerequisites

Delegates must have familiarity in using computer systems such as Windows or Mac OS X.

Duration

5 days

Equipment

The course may be run either:

1. Using ‘blank’ PCs on which Red Hat Enterprise Linux or CentOS is installed; or
2. On Windows, using Virtual Machine software.

Objectives

On completion of this course, you will:

- Understand the structure of a Linux system
- Be able to navigate around a Linux system
- Be able to manipulate and process files and data

- Understand the role of the Shell and the services it provides
- Know how to set up the Shell Environment
- Understand and change file permissions
- Understand the text editors available and use ‘vi’
- Know how to manipulate processes
- Use the networking facilities within Linux
- Understand and use the backup and archive utilities

Course Outline

Introduction to Linux

- Beginnings – UNIX
- What is ‘Open Source’?
- Linux & GNU
- Linux Distributions

System Architecture

- Kernel
- Shell
- The Filesystem
- Graphical User Interfaces: Gnome, KDE

Logging On and Getting Help

- Command Line vs GUI
- `exit` / `^D`
- Getting Help – `whatis`, `man`, `info` (for Gnu utilities)
- Online Documentation

- Exercise

Getting Started

- Command Structure
- Displaying Output - `echo`
- Getting Current Date and Time - `date`
- Listing Who's On - `who`
- Displaying Files - `cat`
- Listing Directories - `ls`; `ls –l`; `ls –a`; `ls –al`
- Changing Directory - `cd`
- Printing the Working Directory - `pwd`

The Filesystem

- Tour of the Filesystem
- Filenames
- Pathnames – absolute and relative
- Changing Directories – `cd`
- Listing Directories – `ls`
- Command Syntax
- Referring to Files & Directories
- Navigating the Filesystem
- Disk Space
- Exercise

Working with Files and Directories

- Copying files - `cp`
- Moving/Renaming Files - `mv`
- Removing Files - `rm`

- Linking Files - `ln`
- Making Directories - `mkdir`
- Removing Directories - `rmdir`
- Determining File Contents with the '`file`' command
- Identifying problem characters with '`od`' and '`cat`'
- Exercise

Using Shell Metacharacters

- Shell Metacharacters (Wildcards)
- Exercise

Understanding and Using Shell Variables

- Variables
- Environment Variables
- Exercise

Setting up the Shell Environment

- Login Shell
- Non-login Shell
- `profile`
- `rc` file
- Exercise

Knowing How the Command Line is Processed

- Quoting
- Command Separation
- Exercise

I/O Redirection, Pipes and Filters

- Redirecting I/O to and from Files

- Using Device Files
- Pipes
- Common Filters
- Command Substitution
- Exercise

Processing Text Data

- Concatenating Files - `cat`
- Displaying Files a Page At A Time - `more` & `less`; `pr`
- Viewing the Head or Tail of a File - `head` & `tail`
- Extracting Vertical 'Slices' of a File – `cut`
- Merging Lines of Files - `paste`
- Translating Characters - `tr`
- Finding Data - `grep`
- Counting Words – `wc`
- Exercise

Regular Expressions

- Why Regular Expressions?
- Searching for Simple Text
- Wildcards
- Extended Regular Expressions
- Using Regular Expressions at the command line
- The `grep` command (including `fgrep` & `egrep`)
- Exercise

Comparing Files

- Comparing Files - `cmp`

- Getting Differences between Files - `diff`
- Getting Commonalities across Files – `comm`
- Exercise

Sorting Files

- The `sort` command
- Specifying sort keys
- Sorting by column positions
- The `uniq` command
- Exercise

File Permissions and Security

- How Permissions Work
- Permission Evaluation
- Permission Types
- Permissions on Files
- Permissions on Directories
- Changing Permissions – Symbolic
- Changing Permissions – Numeric
- Groups
- Changing File Ownership
- Becoming Root
- SUID & SGID
- Exercise

Script-based Editing in Linux

- The '`ed`' Line Editor
- The '`sed`' Stream Editor

- `sed` - Pattern Space
- `sed` - Addresses
- `sed` - The hold and get Functions
- `sed` - Grouping
- `sed` - Advanced Flow Control
- Exercise

The vi Editor

- `vi` – A Visual Editor
- Using `vi`
- Exercise

Processes

- Listing & Viewing Processes
- Sending Signals to Processes
- Running Background Jobs
- Grouping Commands
- Killing Processes
- Shell Job Control
- Running Processes at Specified Times - `at` and `cron`
- Exercise

Finding Files

- Searching for Files in a Directory Tree - `find`
- Finding Files by Name - `locate`
- Exercise

The Pattern Scanning Utility – awk

- Command Line Syntax
- Program Structure

- Patterns
- Logical Operators
- `BEGIN` and `END` Patterns
- Variables
- Controlling Output
- Program Control Structures
- Functions
- Exercise

Networking

- Finding & Setting IP Address
- Network Tools
- Exercise

Working Remotely

- File Transfer – `ftp` & `sftp`
- Downloading Files Non-Interactively - `wget`
- Secure Shell (Remote Access) - `ssh`
- Secure Copy (Remote Copy) - `scp`
- Synchronising Local and Remote Directory Trees - `rsync`
- Sharing Files with Windows - Samba
- Exercise

Compression & Archiving Utilities

- Compressing Files - `compress`; `gzip`; `bzip2`; `zip`
- Archiving Files - `tar`
- Archiving Files - `cpio`
- Exercise

Shell Programming

- Using variables

- Program Control Flow
- Conditional Execution
- The `test` Command
- The `case` Statement
- The `while` Statement
- The `until` Statement

Wrap-up