

Multi-vendor UNIX Fundamentals

Learn via: **Classroom**

Duration: **4 Days**

Overview

Multi-Vendor UNIX Fundamentals course from QA will provide the delegates with transferable skills, and - equally important - will give them appreciation of working in a multi-vendor UNIX environment. The course concentrates on the common approach to the variants of UNIX whilst looking at some specific areas of particular systems and shells. The platform used for practical exercises is Linux, with SSH access provided to at least one other version of UNIX, for example HP-UX.

The course explains how to use the basic facilities of a UNIX system and demystifies most of the major concepts and principles of UNIX operation. We discuss the role the shell plays in interfacing with the operating system, analyse the file system and explore file, directory and data manipulation utilities.

We also take a look at more advanced use of the shell, including a brief introduction to shell scripts (command or batch files) and how to use them to automate repetitive tasks. There is a very useful overview of simple UNIX system administration and network communication utilities.

Hands-on work (labs, demos and interactive exercises) accounts for nearly half of the course. Each lab session is carefully structured to lead you through the concepts of the preceding chapter, thereby building knowledge and confidence in using UNIX. Each delegate is assigned his/her own machine, and all machines are configured as a network.

Prerequisites

- No specific UNIX knowledge is required, but delegates must have a basic knowledge of the major computer components as well as computer operating system functionality and concepts.
- Ability of working with command line (such as in DOS or VMS) and familiarity with hierarchical file structure, organisation and management is assumed.
- This course is aimed at IT professionals wanting to become proficient in using a UNIX platform in a multi-vendor environment. It is also recommended for users of any UNIX system who need to 'escape' an application and use the system at the command line level.

Outline

Introduction

- What is UNIX?
- Origins and variants
- Architecture and purpose
- Good and bad points

UNIX Session

- UNIX users
- Logging in process
- GUI vs. CLI
- Shell programs
- Basic commands
- Logging out

The File System

- File system concepts
- Navigating the file system
- Manipulating files and directories
- Examining file contents

The vi Editor

- Concepts of the visual editor
- Useful vi commands
- Extended vi commands

The Korn Shell

- What is a shell?
- Shell as a command line interpreter
- Commands and arguments
- Using wildcards
- Quotes
- Command history and line editing

The Shell Environment

- Shell variables
- Customising user's environment
- The search path
- Korn shell aliases
- Start-up files: /etc/profile, .profile and .kshrc

Command Input and Output

- UNIX I/O streams
- Standard input, output and error
- I/O redirection
- Using noclobber option
- Redirecting to /dev/null

Processes

- Programs, processes and daemons
- Communicating with processes using signals
- Background jobs and job control
- Multiple commands and sub-shells
- Command substitution
- Batch jobs with at

Pipes and Filters

- Command pipelines
- Simple filters
- Building complex filter pipelines
- Using pipelines in problem solving

Organising Files

- File ownership
- File protection
- Hard and soft links
- Modifying file access attributes

Power Tools

- Using diff and find
- Regular expressions
- Using regular expressions in a variety of UNIX tools
- Overview of a stream editor sed

Writing Shell Scripts

- Simple scripts
- Positional parameters
- Selection commands
- Looping commands
- Interactive input

Basic Communications

- UNIX in a networked environment
- TCP/IP utilities: ifconfig, ping, traceroute, hostname
- Remote sessions with SSH suite of tools

Looking After a UNIX System

- Starting and stopping the system
- Adding users
- Backup tools
- Interfacing to tapes and disks
- Working with DOS format files

- Deferring tasks with at and batch