

# UNIX Fundamentals and Command-Line Operations Training Course

27 September – 1 October 2027
London

AGILE LEADERS
Training Center

AGILE LEADERS
Training Center

# UNIX Fundamentals and Command-Line Operations Training Course

**Ref:** 1751_231582 **Date:** 27 September – 1 October 2027 **Location:** London **Fees:** 5700 Euro

## Overview

UNIX Fundamentals and Command-Line Operations Training Course is a one-week course for new UNIX users, support staff, junior administrators, developers, and operations personnel who leave with a UNIX Command-Line Operations Workbook and Troubleshooting Checklist. Participants navigate filesystems, manage permissions, process text, control processes, build shell scripts, test network utilities, and diagnose command-line problems. The course distinguishes portable POSIX concepts from implementation-specific behavior. Agile Leaders Training Center develops UNIX fundamentals and command-line operations capability.

## Who Should Attend

- Technical support functions responsible for diagnosing command-line and access issues
- Operations functions responsible for routine UNIX system tasks and monitoring
- Junior administration functions responsible for files, permissions, processes, and user support
- Development functions responsible for running tools, scripts, and build tasks in UNIX environments
- Application support functions responsible for logs, services, and basic network checks

The course assumes participants can use a keyboard and manage files on a computer and leaves out kernel development, vendor certification preparation, and advanced system administration.

## Departments and Industries

The course supports technical teams that use UNIX command-line environments across service and operating settings.

- Information technology, infrastructure, and service operations
- Software development, testing, and platform engineering
- Telecommunications, financial services, and healthcare technology
- Energy, manufacturing, and logistics operations
- Education, research computing, and public services

## Learning Objectives

By the end of this course, participants will be able to:

AGILE LEADERS
Training Center

- Use UNIX shell syntax, help resources, paths, and command options
- Apply file, directory, ownership, and permission operations
- Build text-processing pipelines with redirection and standard utilities
- Analyze processes, jobs, signals, environment variables, and exit status
- Build portable shell scripts for routine operational tasks
- Diagnose command, filesystem, process, and network utility problems

## Course Agenda

## Day 1: UNIX Shell and Navigation

- UNIX Architecture and Shell Interaction Map
- Command Syntax, Options, and Arguments Pattern
- Manual Page and Built-In Help Search Method
- Absolute and Relative Path Navigation Practice
- Environment Variable and Search Path Inspection

## Day 2: Files and Permissions

- Filesystem Object and Directory Tree Model
- File Creation, Copy, Move, and Removal Commands
- Wildcard and Filename Expansion Rules
- Owner, Group, and Other Permission Matrix
- chmod, chown, and umask Access Practice

## Day 3: Text and Data Streams

- Standard Input, Output, and Error Flow Diagram
- Redirection and Pipeline Construction Method
- grep Pattern Search and Result Filtering
- cut, sort, uniq, and wc Text Processing Chain
- sed and awk Introductory Transformation Practice

## Day 4: Processes and Shell Scripts

- Process Hierarchy and Status Inspection Method
- Foreground, Background, and Job Control Commands
- Signal, Exit Status, and Error Handling Checklist
- Shell Variable, Quoting, and Expansion Rules
- Portable Shell Script Structure and Test Plan

## Day 5: UNIX Operations Practice

- Exercise: Repair a Filesystem Navigation and Permission Issue
- Exercise: Build a Log-Filtering Pipeline
- Exercise: Diagnose a Process and Job-Control Problem
- Exercise: Test Network and Name-Resolution Utilities
- Capstone Exercise: UNIX Command-Line Operations Workbook and Troubleshooting Checklist

## Practical Exercises

The course uses suggested activities to connect UNIX commands with operating evidence and repeatable checks.

- Suggested activity: organize application files and apply access modes for a development support case.
- Suggested activity: filter service logs and summarize recurring errors for a healthcare platform team.
- Suggested activity: inspect processes, manage background jobs, and interpret exit status during an operations incident.
- Suggested activity: write and test a portable shell script that checks files, commands, and network responses.

## FAQs

### Who suits UNIX fundamentals and command-line operations, and what does it assume?

UNIX fundamentals suits new users, support staff, junior administrators, developers, and operations personnel. It assumes basic computer file handling and does not require prior UNIX command-line experience.

### How do UNIX fundamentals differ from advanced system administration training?

UNIX fundamentals develops command-line, filesystem, permission, text, process, scripting, network utility, and troubleshooting skills. Advanced system administration training usually covers installation, storage, services, security configuration, performance tuning, and recovery.

### What makes a UNIX shell command portable?

A UNIX shell command is more portable when it uses POSIX-defined shell behavior and utilities without relying on vendor-specific options. Implementations still need testing because available commands, paths, and extensions can differ.

AGILE LEADERS
Training Center

## How do UNIX permissions control file and directory access?

UNIX permissions assign read, write, and execute modes to the owner, group, and others. Their effect differs between files and directories, so access checks should consider the whole path.

## What should a UNIX troubleshooting checklist include?

A UNIX troubleshooting checklist should capture the command, arguments, working directory, user identity, permissions, environment, input and output, exit status, process state, network response, and repeatable verification steps.

## Conclusion

Participants take a UNIX Command-Line Operations Workbook and Troubleshooting Checklist back to work. It combines command patterns, path checks, permission tables, text pipelines, process controls, script tests, network checks, and diagnostic evidence. The workbook changes trial-and-error command use into repeatable operating steps that support staff, developers, and junior administrators can review and reuse.