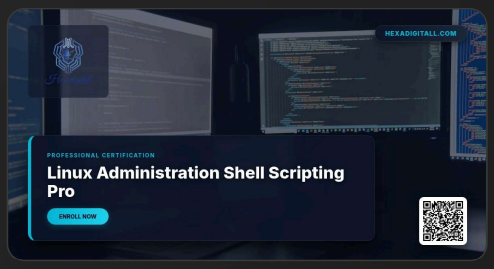
Hexadigitall Technologies
https://hexadigitall.com

Scan to open this course page, enroll, and access live updates.

Course Snapshot

Master the Linux command line, automation, and shell scripting with real-world labs and DevOps-ready skills.



Linux Administration & Shell Scripting Pro

Master the Command Line & Become a Linux Power User

Duration: 10 Weeks | **Level:** Intermediate | **Study Time:** 5 hours/week | **School:** Cloud & DevOps | **Modules:** 5 | **Lessons:** 25

Welcome to Linux Mastery!

Welcome to your comprehensive journey into Linux administration and shell scripting! This 10-week intensive program is designed to transform you from a Linux user into a proficient system administrator and automation expert.

Linux powers over 96% of the world's top 1 million web servers, and shell scripting is the backbone of automation in DevOps, cloud computing, and system administration. By mastering these skills, you'll open doors to countless career opportunities in the tech industry.

Throughout this course, you'll dive deep into Linux system administration, master the command line, write powerful shell scripts, and learn automation techniques used by industry professionals. You'll work on real-world scenarios, complete hands-on labs, and build a portfolio of practical scripts.

Get ready to unlock the power of Linux! This course emphasizes practical, hands-on learning with 5 modules covering 25 comprehensive lessons. By the end, you'll be confident managing Linux systems and automating complex workflows.

Prerequisites & What You Should Know

Before Starting This Course

- ✓ Basic computer literacy and familiarity with using computers
- ✓ Understanding of file systems and directory structures
- ✓ Basic text editing skills (using any text editor)
- ✓ Fundamental programming concepts (variables, loops, conditionals) - helpful but not required
- ✓ Willingness to work with command-line interfaces
- ✓ Access to a computer (Windows, Mac, or Linux) for lab practice

💡 **Getting Started:**

You'll need a Linux environment for hands-on practice. We recommend using VirtualBox with Ubuntu, Windows Subsystem for Linux (WSL2), or a cloud instance (AWS EC2, Azure VM). Setup instructions will be provided in Week 1.

Recommended Complementary Courses

DevOps Engineering & Kubernetes

Apply your Linux skills to container orchestration and DevOps workflows.

Python for Data Science

Complement shell scripting with Python for advanced automation tasks.

Cloud Infrastructure Deployment

Learn to deploy and manage Linux systems in cloud environments.

Network Security Administration

Secure your Linux systems with advanced networking and security practices.

Essential Learning Resources

Free Resources

Linux Journey

linuxjourney.com FREE

The Linux Documentation Project

tldp.org FREE

GNU Bash Manual

gnu.org/software/bash/manual FREE

Linux Command Library

linuxcommandlibrary.com FREE

ExplainShell

explainshell.com FREE

Ubuntu Official Docs

help.ubuntu.com FREE

💎 Recommended Paid Resources

Linux Academy / A Cloud Guru

acloudguru.com/linux PAID

The Linux Command Line (Book)

By William Shotts PAID

UNIX and Linux System Administration

By Evi Nemeth et al. PAID

Linux Foundation Training

training.linuxfoundation.org PAID

🛠️ Essential Tools to Install

Visual Studio Code

Text editor with excellent shell script support

VirtualBox or VMware

For running Linux virtual machines

PuTTY or Terminal

SSH client for remote server access

ShellCheck

Shell script static analysis tool

Your Learning Roadmap

Phase 1: Linux Fundamentals (Weeks 1-2)

Master the Linux file system, basic commands, and essential system navigation.

Phase 2: User & File Management (Weeks 3-4)

Learn user administration, permissions, and advanced file operations.

Phase 3: Shell Scripting Basics (Weeks 5-6)

Write your first shell scripts with variables, conditionals, and loops.

Phase 4: Advanced Administration (Weeks 7-8)

Manage processes, services, networking, and system monitoring.

Phase 5: Automation & Projects (Weeks 9-10)

Build advanced automation scripts and complete capstone projects.

Skills You'll Master

- Linux Command Line
- Bash Scripting
- File System Management
- User Administration
- Process Management
- Package Management
- System Monitoring
- Network Configuration
- Automation
- Cron Jobs & Scheduling
- Text Processing (sed, awk)
- Log Analysis

Detailed Weekly Curriculum

Week 1

5 hours

Linux Fundamentals & System Architecture

- $ Introduction to Linux and its distributions
- $ Linux architecture and kernel overview
- $ Setting up your Linux environment (VirtualBox/WSL)
- $ Navigating the file system (pwd, cd, ls)
- $ Understanding the Linux directory structure (/, /home, /etc, /var)
- $ Basic command structure and getting help (man, --help)

🔬 Hands-On Lab

- $ Install Ubuntu Linux (virtual machine or dual boot)
- $ Practice navigating the file system
- $ Create directory structure for course materials
- $ Explore man pages and command documentation

Week 2

5 hours

Essential Commands & File Operations

- $ File and directory operations (cp, mv, rm, mkdir, rmdir)
- $ Viewing file contents (cat, less, more, head, tail)
- $ Searching and finding files (find, locate, which, whereis)
- $ Text editors (nano, vim basics)
- $ File compression and archiving (tar, gzip, zip)
- $ Wildcards and globbing patterns

🔬 Hands-On Lab

- $ Organize project directories with proper structure
- $ Create, move, and delete files efficiently
- $ Practice with vim text editor
- $ Archive and compress project files

```
# Example commands from this week find /home -name "*.log" -type f tar -czf backup.tar.gz /home/user/documents grep -r "error"
/var/log/
```

Week 3

5 hours

User & Group Management, Permissions

- $ Understanding Linux users and groups
- $ Creating and managing users (useradd, usermod, userdel)
- $ Group management (groupadd, groupmod)
- $ File permissions and ownership (chmod, chown, chgrp)
- $ Understanding permission modes (rwx, numeric notation)
- $ Special permissions (SUID, SGID, sticky bit)
- $ Access Control Lists (ACLs)

🔬 Hands-On Lab

- $ Create users and groups for a development team
- $ Set appropriate file permissions for projects
- $ Practice with chmod numeric and symbolic notation
- $ Implement shared directories with proper permissions

```
# User and permission commands sudo useradd -m -s /bin/bash john sudo chmod 755 /var/www/html sudo chown -R www-data:www-data
/var/www
```

Week 4

5 hours

Package Management & Software Installation

- $ Understanding package managers (apt, yum, dnf)
- $ Installing, updating, and removing packages
- $ Repository management
- $ Compiling software from source
- $ Dependency resolution
- $ Snap and Flatpak package systems

🔬 Hands-On Lab

- $ Install development tools and libraries
- $ Manage repositories and update system
- $ Compile and install software from source
- $ Set up local package repository

```
# Package management examples sudo apt update && sudo apt upgrade sudo apt install nginx mysql-server sudo apt-cache search python | grep dev
```

Week 5

5 hours

Introduction to Shell Scripting

- $ What is a shell script and why use it?
- $ Creating and executing your first script
- $ Shebang (#!/bin/bash) and script interpreters
- $ Variables and environment variables
- $ Command substitution and arithmetic operations
- $ Reading user input
- $ Script debugging basics (set -x, set -e)

🔬 Hands-On Lab

- $ Write a system information script
- $ Create a backup script with user input
- $ Build a calculator script
- $ Practice variable manipulation

```
#!/bin/bash # Simple backup script echo "Enter directory to backup:" read SOURCE_DIR BACKUP_FILE="backup_$(date +%Y%m%d).tar.gz" tar -czf $BACKUP_FILE $SOURCE_DIR echo "Backup created: $BACKUP_FILE"
```

Week 6

5 hours

Control Structures & Functions

- $ Conditional statements (if, else, elif)
- $ Test conditions and comparison operators
- $ Case statements
- $ Loops (for, while, until)
- $ Loop control (break, continue)
- $ Creating and using functions
- $ Function parameters and return values

🔬 Hands-On Lab

- $ Build a menu-driven system admin script
- $ Create a file processor with loops
- $ Write reusable functions library
- $ Develop a log file analyzer

```
#!/bin/bash # Function to check if service is running check_service() { if systemctl is-active --quiet "$1"; then echo "$1 is running" else echo "$1 is NOT running" fi } check_service nginx
```

Week 7

5 hours

Process & Service Management

- $ Understanding processes and PIDs
- $ Process management (ps, top, htop, kill)
- $ Background and foreground processes
- $ Job control (jobs, bg, fg)
- $ System services and daemons
- $ Systemd service management
- $ Creating custom systemd services

🔬 Hands-On Lab

- $ Monitor system processes and resources
- $ Manage running services (start, stop, restart)
- $ Create a custom systemd service
- $ Write a process monitoring script

Week 8

5 hours

Advanced Text Processing & Networking

- $ Text processing with grep, sed, and awk
- $ Regular expressions for pattern matching
- $ Stream editing and text transformation
- $ Network configuration basics
- $ Network troubleshooting (ping, netstat, ss, ip)
- $ SSH and remote server management
- $ Firewall basics (ufw, iptables)

🔬 Hands-On Lab

- $ Parse and analyze log files with sed/awk
- $ Write a log monitoring script
- $ Configure network interfaces
- $ Set up SSH key authentication

```
# Text processing examples grep -E 'ERROR|FATAL' /var/log/syslog sed 's/old/new/g' file.txt awk '{print $1, $3}' data.txt
```

Week 9

5 hours

System Monitoring & Automation

- $ System performance monitoring
- $ Disk usage and management (df, du)
- $ Log management and rotation
- $ Cron jobs and task scheduling
- $ At and batch commands
- $ System backup strategies
- $ Automated maintenance scripts

🔬 Hands-On Lab

- $ Set up automated backups with cron
- $ Create disk space monitoring script
- $ Implement log rotation
- $ Build system health check script

```
# Crontab example for daily backup at 2 AM 0 2 * * * /home/user/scripts/backup.sh # Every hour log cleanup 0 * * * * find /var/log -name "*.log" -mtime +7 -delete
```

Week 10

5 hours

Advanced Scripting & Capstone Projects

- $ Error handling and exit codes
- $ Script best practices and style guidelines
- $ Script optimization techniques
- $ Working with APIs and web data (curl, wget)
- $ Database interaction from scripts
- $ Security considerations in scripts
- $ Capstone project presentations

🔬 Hands-On Lab

- $ Complete capstone project (see below)
- $ Code review and optimization
- $ Document your scripts professionally
- $ Build a personal script library

Capstone Projects

Project 1: Automated Server Setup & Configuration Script

Create a comprehensive bash script that automates the initial setup of a Linux server, including user creation, security hardening, software installation, and service configuration.

Requirements:

- Menu-driven interface for different setup options
- Automatic security updates configuration
- Firewall setup with ufw
- SSH hardening (key-only auth, custom port)
- Install and configure LAMP/LEMP stack
- Error handling and logging
- Idempotent execution (can run multiple times safely)

Project 2: System Monitoring & Alerting Dashboard

Build a comprehensive monitoring solution that tracks system resources, sends alerts, and generates reports. Include email notifications for critical issues.

Requirements:

- Monitor CPU, memory, disk usage, and network
- Alert on threshold violations
- Generate daily/weekly reports
- Log analysis and anomaly detection
- Service health checks
- HTML dashboard generation
- Email notifications using sendmail or SMTP

Project 3: Automated Backup & Disaster Recovery System

Develop a robust backup solution that handles incremental backups, retention policies, compression, encryption, and automated restoration testing.

Requirements:

- Full and incremental backup support
- Configurable retention policies
- Compression and encryption
- Remote backup (rsync/scp to remote server)
- Backup verification and integrity checks
- Restoration script with rollback capability
- Email reports on backup success/failure

Project 4: Log Analysis & Security Audit Tool

Create an advanced log processing tool that analyzes system logs for security threats, performance issues, and generates actionable insights.

Requirements:

- Parse multiple log formats (/var/log/*)
- Detect failed login attempts
- Identify unusual patterns or anomalies
- Generate security audit reports
- Export results to CSV/JSON
- Integration with fail2ban or similar tools
- Scheduled execution via cron

Study Tips for Success

- ✓ Practice daily - even 30 minutes of command-line practice helps!
- ✓ Set up your own Linux environment (VM or WSL) for experimentation
- ✓ Read other people's scripts on GitHub to learn different approaches
- ✓ Always test scripts in a safe environment before production use
- ✓ Use version control (Git) for your scripts
- ✓ Document your code with comments - your future self will thank you
- ✓ Break complex problems into smaller, manageable functions
- ✓ Learn to read error messages - they're your friends!
- ✓ Join Linux communities (r/linux, Stack Overflow, Linux forums)
- ✓ Keep a personal "cheat sheet" of frequently used commands
- ✓ Experiment and break things - it's the best way to learn!

Shell Scripting Best Practices

Professional Standards

- ✓ Always use proper shebang (#!/bin/bash)
- ✓ Set error handling (set -e, set -u, set -o pipefail)
- ✓ Quote your variables ("$variable") to prevent word splitting
- ✓ Use meaningful variable names (BACKUP_DIR vs d)
- ✓ Comment complex logic and non-obvious code
- ✓ Check command exit codes ($?)
- ✓ Validate user input before processing
- ✓ Use functions to organize code and avoid repetition
- ✓ Test scripts with ShellCheck for potential issues
- ✓ Log important actions and errors

Career Opportunities

Linux System Administrator

Manage and maintain Linux servers, automate tasks, and ensure system reliability.

DevOps Engineer

Use shell scripting for CI/CD pipelines, infrastructure automation, and deployment.

Site Reliability Engineer (SRE)

Ensure system reliability through automation, monitoring, and incident response.

Cloud Engineer

Manage cloud infrastructure with Linux-based systems and automation scripts.

Security Engineer

Use shell scripting for security audits, log analysis, and vulnerability scanning.

Backend Developer

Leverage Linux and scripting skills for server-side application development.

Let's Master Linux Together! 🚀

You're about to embark on an exciting journey into the world of Linux and automation.

Remember: Every expert was once a beginner. Stay curious, practice consistently, and embrace the command line!

```
$ sudo apt install success
$ chmod +x your_career.sh
$ ./your_career.sh --execute
```