

Linux, Networking, and Shell Scripting: From Fundamentals to Administration – Progressive Structured Guide

1. Introduction to Linux & Command Line

1.1 Linux Core System Components

- Kernel - Core of Linux; manages hardware, memory, processes, and communication between software and hardware.
- Shell - Command-line interface (CLI) such as Bash or Zsh used to type commands.
- Terminal - Program providing access to the shell where commands are typed and run.
- File System Hierarchy - Tree-like structure starting from Root Directory (/) where all files reside.
- Root User - "Super user" account with unlimited system access.
- Permissions - Controls determining who can read, write, or execute files/directories.
- Process (PID) - Running program instance identified by unique Process ID.
- Init System (systemd) - First process started by kernel; launches/manages services.
- Daemons - Background processes (sshd, cron) running without direct user interaction.

1.2 Getting Started with Commands

- Terminal Prompt Structure: username@hostmachine:~\$
- date - Prints current time/date
- df - Shows free disk space
- free - Shows free memory
- exit - Exits terminal session
- Command History - Use up/down arrow keys to navigate

2. File System Navigation & Management

2.1 Basic Navigation Commands

- pwd - Print Working Directory (shows current location)
- ls - List directory contents
- cd - Change Directory
- cd alone - Returns to home directory
- cd .. - Moves up one directory

Shortcuts:

Tab Autocomplete - Automatically completes file/directory names

clear - Clears terminal screen

2.2 Path Types

- Absolute Path - Starts from root (/home/user/file.txt)
- Relative Path - Relative to current location (./documents/file.txt)

2.3 File Properties & Listing Options

- LS Options:
 - ls -l - Long format with details
 - ls -a - Includes hidden files
 - ls -la - Combined options

- Long Format (`ls -l`) Output:
 - Access Rights - Permissions (read/write/execute)
 - Hard Links Count - Number of links to file
 - Owner & Group
 - File Size
 - Modification Date/Time
 - File Name
 - Leading d - Denotes directory
- File Type Commands:
 - `touch <filename>` - Creates empty file
 - `file <filename>` - Shows file type/details
 - `less <filename>` - Views text files (press q to exit)

3. File Operations & Links

3.1 Directory & File Manipulation

- `mkdir <dir_name>` - Creates directory
- `cp <source> <destination>` - Copies files/directories
- `cp -r` - Copies directories recursively
- `mv <source> <destination>` - Moves or renames files
- `rm <file_name>` - Removes file
- `rm -r <dir_name>` - Removes directory recursively

3.2 Wildcards

* - Matches zero or more characters (e.g., *.html)

3.3 Links in Linux

- Hard Links:
 - Points directly to file data
 - Deleting original doesn't remove data
 - Same inode number as original
- Symbolic (Soft) Links:
 - Shortcut to file/directory
 - Breaks if original deleted
 - Denoted by l in `ls -l`
- Creating Links:
 - `ln <target> <link_name>` - Creates hard link
 - `ln -s <target> <link_name>` - Creates symbolic link

4. I/O Redirection & Pipelines

4.1 Standard Streams

- STDIN (0) - Input (keyboard)
- STDOUT (1) - Output (screen)
- STDERR (2) - Error output

4.2 Redirection Operators

- > - Redirects STDOUT to file (overwrites)
- >> - Redirects STDOUT to file (appends)

- 2> - Redirects STDERR to file
- < - Redirects file content to STDIN

4.3 Pipelines

- | - Pipe operator sends STDOUT of one command to STDIN of next
- Pipeline Commands:
 - cat <file> - Prints file contents
 - sort - Sorts input alphabetically
 - uniq - Removes duplicate lines
 - grep <pattern> <file> - Searches for text patterns
 - head <file> - Prints first 10 lines
 - tail <file> - Prints last 10 lines
 - tee - Sends output to both file and STDOUT

5. Shell Expansion & Variables

5.1 Types of Expansion

- Pathname Expansion - Wildcards (*)
- Tilde Expansion - ~ expands to home directory
- Arithmetic Expansion - \$() for math calculations
- Parameter Expansion - Uses variables (\$USER)
- Command Substitution - \$() uses command output

5.2 Variables

- System Variables - \$USER, \$PATH
- printenv - View system variables

5.3 Quoting

- Double Quotes (" ") - Preserve spaces; allow expansions
- Escaping (\) - Prevents special interpretation (\\$)

6. Commands, Aliases & Help

6.1 Command Types

- Executable programs
- Shell built-ins
- Shell functions
- Aliases

6.2 Command Information

- type <command> - Shows command type
- which <command> - Shows full path to executable
- man <command> - Opens manual page
- whatis <command> - One-line description

6.3 Command Execution

- Sequential Commands - Separate with ;
- Aliases - alias name='command1; command2'

- Remove Alias - `unalias <name>`

7. Networking Fundamentals

7.1 IP Addresses

- Definition - Unique identifier for network devices
- IPv4 Format - Four bytes separated by dots (172.16.3.4)
- Range - Each byte: 0-255 (8 bits = 2^8 -1)

7.2 Subnetting

- Purpose: Split large networks into smaller, isolated segments
- Private Subnet - No direct internet access
- Public Subnet - Internet access via Internet Gateway
- Benefits:
- Enhanced security
- Privacy
- Isolation between groups

7.3 CIDR (Classless Inter-Domain Routing)

- Format - Slash notation (/24)
- Calculation - Available IPs = $2^{(32-N)}$ where N is CIDR number
- Example: /24 = 256 IP addresses
- Private IP Ranges: 10.x.x.x, 172.16.x.x, 192.168.x.x

7.4 Ports

- Function - Identify specific applications on devices
- Format - IP_Address:Port
- Purpose - Distinguish between applications on same server

8. OSI Model & Network Communication

8.1 Pre-OSI Steps

- DNS Resolution - Maps domain name to IP address
- TCP Handshake - Three-way connection establishment

8.2 OSI Model Layers

Layer	Name	Function	Key Protocols/Tools
7	Application	Interface for network applications	HTTP, HTTPS, FTP
6	Presentation	Data formatting, encryption	SSL/TLS (HTTPS)
5	Session	Connection management	Maintains login sessions
4	Transport	Data segmentation	TCP (reliable), UDP (fast)
3	Network	IP addressing, routing	IP, routers
2	Data Link	MAC addressing	Ethernet, switches
1	Physical	Physical transmission	Cables, wireless signals

- Data Flow:
 - Sending: Layer 7 → Layer 1
 - Receiving: Layer 1 → Layer 7
 - TCP/IP Model: Combines OSI Layers 5-7 into single Application Layer

9. Shell Scripting Fundamentals

9.1 Purpose & Need

- Automation - Replace repetitive manual tasks
- DevOps Applications:
 - Infrastructure automation
 - Configuration management
 - Code/Git management
 - Node health monitoring

9.2 Basic Script Structure

```
#!/bin/bash # Shebang - specifies shell
# Author: Name
# Date: YYYY-MM-DD
# Purpose: Description
# Version: V1.0

set -x # Debug mode
set -e # Exit on error
set -o pipefail # Catch pipeline failures

echo "Starting script..."
# Commands here
```

9.3 Essential Commands for Scripting

- System Health Monitoring:
 - df -h - Disk space
 - free -g - Memory usage
 - nproc - CPU core count
 - top/htop - Running processes
- Process Management:
 - ps -ef - List all processes
 - ps -ef | grep <pattern> - Filter processes
 - ps -ef | awk '{print \$2}' - Extract specific columns
- File Operations:
 - touch - Create empty files
 - cat - View file content
 - vim/vi - Edit files
 - Insert mode: Press i
 - Save/quit: Esc then :wq
- Network Operations:
 - curl - Retrieve data from URLs

- wget - Download files

9.4 Control Structures

If-Else Conditions:

```
if [ condition ]; then
    # commands
elif [ condition ]; then
    # commands
else
    # commands
fi
```

For Loops:

```
for I in list; do
    # commands
done
```

trap Command: Intercept signals (Ctrl+C = SIGINT)

trap 'cleanup_function' SIGINT

9.5 AWS Automation Script Example

- Script Purpose: List AWS resources for cost management
- Usage: ./AWS_resource_list.sh [region] [service]
- Validation Checks Required:
 - Argument count check
 - AWS CLI installation check
 - AWS CLI configuration check
- Security Best Practice:
 - chmod 771 script.sh # Restrict permissions

10. Essential Linux Commands for DevOps

10.1 System Monitoring

- top/htop - Real-time system performance
- ps aux - Process listing with details
- pgrep <app> - Find process ID
- pstree - Process hierarchy

10.2 Network Tools

- netstat -tnl - Network connections/ports
- tcpdump -i eth0 - Network packet analysis
- ping <target> - Basic connectivity
- traceroute <target> - Network path analysis

10.3 Disk & Memory

- df -h - Disk space usage
- du -sh [folder] - Directory size
- free -h - Memory utilization

10.4 System Logs & Files

- journalctl -u <service> - Service logs
- lsof -i :<port> - Process using port
- tail -n 10 [file] - Last lines of file
- head -n 10 [file] - First lines of file

10.5 Useful Shortcuts

- Ctrl + R - Reverse command history search
- export PS1 - Customize terminal prompt

11. Linux Administration Fundamentals

11.1 Linux Structure & Distributions

- History:
 - Unix (1960s) - First popular OS
 - Windows (1980s) - GUI revolution
 - Linux (1990s) - Open-source kernel
 - Modern Status: 90% of production workloads use Linux
 - Distributions: Ubuntu (beginners), Red Hat, Debian
- Setup Options:
 - WSL - Windows Subsystem for Linux
 - Docker Container - Lightweight Linux environment

11.2 Package Management

- Package Managers: apt (Ubuntu), dnf (Fedora), pacman (Arch)
- apt update - Refresh package lists
- apt install <package> - Install software

11.3 Linux Directory Structure

Directory	Purpose	Key Contents
/	Root Directory	All files/directories
/bin	User Binaries	Essential user commands (ls, date)
/sbin	System Binaries	System admin tools (useradd, fdisk)
/usr	User Services	Secondary binaries, libraries
/lib	System Libraries	Kernel/library files
/etc	Configuration	System config files
/home	User Home Directories	Personal files per user
/root	Root User Home	Administrator home
/opt	Optional Software	Third-party applications
/mnt	Mount Point	Temporary filesystem mounts

/var	Variable Data	Logs, mail, caches
/boot	Boot Files	OS boot/restart files
/tmp	Temporary Files	Cleared periodically
/proc / /dev / /sys	Virtual Filesystems	Process/device/kernel info

PATH Variable: Directory list searched for executables

11.4 User & Group Management

- Why Individual Users?
 - Accountability tracking
 - Security prevention
 - Multi-user server requirements
- User Commands:
 - useradd <username> - Basic user creation
 - adduser <username> - Interactive with home directory
 - passwd <username> - Set password
 - userdel <username> - Delete user
 - su - <username> - Switch user
- Password Security:
 - Stored in /etc/shadow (encrypted)
 - One-way encryption (cannot decrypt)
- Group Management:
 - groupadd <groupname> - Create group
 - usermod -aG <group> <user> - Add user to group
 - Efficient permission management for multiple users
- Remote Access:
 - ssh username@IP_address

11.5 File Management Commands

- Navigation:
 - ls - List contents
 - cd - Change directory
 - pwd - Print working directory
- Creation:
 - mkdir - Create directory
 - touch - Create file
 - echo "text" > file - Write to file (overwrite)
 - echo "text" >> file - Append to file
- Deletion:
 - rmdir - Remove empty directory
 - rm - Remove file
 - rm -rf - Force recursive delete (DANGEROUS)
- Manipulation:
 - cp - Copy
 - mv - Move/rename

11.6 Vi/Vim Editor

- Three Modes:
 - Normal Mode - Navigation, commands (default)
 - Insert Mode - Text editing (press i)
 - Command Mode - File operations (Esc then :)
- Essential Commands:
 - :wq! - Save and quit
 - :q! - Quit without saving
 - cat <file> - View small files
 - less <file> - View large files (press q)
 - head -n <num> <file> - First N lines
 - tail -n <num> <file> - Last N lines

12. File Permissions Management

12.1 Permission Structure

- 10-character string from ls -l:
- First char: File type (- = file, d = directory)
- Next 9 chars: Three sets for:
 - User (u) - File owner
 - Group (g) - Owner's group
 - Others (o) - Everyone else
- Permission Types:
 - Read (r) - View file/list directory
 - Write (w) - Modify file/create files in directory
 - Execute (x) - Run script/enter directory

12.2 Permission Management

- chmod - Change Mode:
 - Alphabetical: chmod u=rwx,g=rw,o=r filename
 - Numerical (Octal): r = 4, w = 2, x = 1
- Example: chmod 764 filename (User=7, Group=6, Others=4)
 - chown - Change Owner:
 - chown newuser:newgroup filename

Important Note: Directory permissions override file permissions

13. Process Management

13.1 Process Basics

Definition: Running instance of a program managed by OS

Challenge: Resource-intensive processes can starve others

13.2 Process Commands

- Viewing Processes:
 - ps aux - Detailed process list with CPU/memory
 - ps -ef - Alternative listing
 - ps aux | wc -l - Count running processes
- Process Control:

- kill <PID> - Graceful termination
- kill -9 <PID> - Forceful termination
- kill -3 <PID> - Thread dump (Java debugging)
- kill -stop <PID> - Pause process
- kill -cont <PID> - Resume process
- Priority Management:
 - renice -n -5 <PID> - Increase priority (lower nice value)
 - renice -n 10 <PID> - Decrease priority

13.3 Service Management

- Services: Background processes auto-starting on boot
- systemctl list-units --type=service - View services
- systemctl stop/start <service> - Manage services

14. System Monitoring

14.1 Resource Monitoring

- CPU/Memory:
 - top - Real-time process monitoring
 - htop - Enhanced visual interface
 - vmstat - System performance report
 - free -h - Memory usage (script-friendly)
 - nproc - CPU core count
- Disk Usage:
 - df -h - Filesystem space
 - du -sh - Directory size analysis

14.2 Modern Monitoring

- Production Tools: Prometheus + Grafana for:
 - Real-time metrics
 - Complex queries
 - Automated alerting

15. Disk Management

15.1 Volume Management

- Viewing Storage:
 - lsblk - Block storage devices
 - sudo fdisk -l - Detailed partition info

15.2 Adding New Storage

- Step-by-Step Process:
 - Attach Volume - Create and attach block storage
 - Format Volume - Create filesystem: mkfs -t ext4 /dev/xvdf
 - Mount Volume - Make accessible:
 - mount /dev/xvdf /mnt/demo-volume
 - Unmount: umount /mnt/demo-volume

16. Linux Software & Automation Tools

16.1 Package & Software Management

- Package Managers: apt (Debian/Ubuntu), dnf (Fedora), pacman (Arch)
 - Fetch software from Package Repositories
 - Kernel Modules - Loadable kernel code for features/drivers
 - Shell Scripts - Text files automating command sequences
 - Environment Variables - Configuration variables like \$PATH

16.2 Advanced Tools

- Compiling Tools:
 - make - Build automation
 - GCC - Compiler suite
 - Containers & Virtualization:
- Containers: Docker, Podman
- Virtualization: KVM, QEMU
- File Systems: ext4, XFS, Btrfs (different performance features)
 - File Sharing:
 - Protocols: NFS, Samba
 - Transfer Tools: SCP, rsync

16.3 Modern Packaging

- Distribution Packages: .deb (Debian), .rpm (Red Hat)
- Universal Packages: Flatpak, Snap, AppImage (cross-distribution)

16.4 Graphical Environment

- Display Servers: X11, Wayland
- Desktop Environments: GNOME, KDE, XFCE
- Shell Built-ins: cd, echo (faster than external programs)

17. Data Flow & File Systems

17.1 Virtual File Systems

- /proc - Real-time process/system info
- /dev - Device files for hardware interaction
- dmesg log - Kernel/system messages for debugging

17.2 Storage Management

- LVM (Logical Volume Manager) - Flexible disk management
- Swap Space - Disk area acting as backup memory
- Partition Tools: fdisk, parted

18. Security & Access Controls

18.1 Security Tools

- Firewalls: iptables, ufw (network traffic control)
- Mandatory Access Controls: SELinux, AppArmor
- PAM (Pluggable Authentication Modules) - Login/credential management

18.2 User Privileges

- sudo - Execute commands with admin privileges
- Run Levels/Targets - System operating modes (single/multi-user, graphical)

19. Terminal Efficiency Tools

- Terminal Multiplexer (tmux) - Multiple terminals in one window
- Cron Jobs - Scheduled automated tasks
- System Logs (/var/log) - Troubleshooting data
- SSH (Secure Shell) - Encrypted remote connections

Key Linux Concepts Summary Table

Category	Key Concepts	Essential Commands
File System	Hierarchy, Links, Permissions	ls, cd, pwd, chmod, chown
Processes	PID, Services, Daemons	ps, top, kill, systemctl
Networking	IP, Subnets, Ports, OSI Model	ping, netstat, ssh, curl
Storage	Partitions, Mounting, LVM	df, du, mount, lsblk
Scripting	Automation, Variables, Control Flow	bash, echo, if, for, trap
Monitoring	Resource usage, Logs	htop, journalctl, free, nproc
Security	Users, Permissions, Firewalls	useradd, passwd, sudo, iptables
Package Mgmt	Installation, Dependencies	apt, dnf, snap, flatpak

10 Essential Linux commands for software engineers.

- top / htop - Monitor system performance, including CPU, memory, and active processes, in real-time.
- ps / pgrep / pstree - View running processes, find process IDs, and visualize parent-child relationships.
- netstat / ss - Inspect network connections, listening ports, and interface statistics.
- tcpdump - Capture and analyze network packets to diagnose connectivity issues.
- ping / traceroute / mtr - Test network connectivity and trace the path to a remote host.
- df / du - Check disk space usage and find large directories or files.
- free / vmstat - Monitor memory usage and view virtual memory statistics.
- journalctl - Access system logs managed by systemd to troubleshoot service issues.
- lsof - Identify open files and the processes that opened them.
- tail / less / grep - Inspect and filter log files to find errors or specific patterns.