Comprehensive Guide to Linux: From Basics to Advanced

This guide provides an in-depth, chapter-wise exploration of Linux, covering everything from beginner concepts to advanced system administration.

Chapter 1: Introduction to Linux

What is Linux?

Linux is a free, open-source operating system kernel developed by Linus Torvalds in 1991. It is widely used in servers, desktops, embedded systems, and supercomputers.

Why Use Linux?

- **Free and Open Source**: Cost-effective and transparent.
- **Security**: Advanced permissions and configurations.
- **Customizability**: Modify as needed.
- **Community Support**: Large community for assistance.

Chapter 2: Installing Linux

Choosing a Linux Distribution (Distro)

Popular options:
- **Ubuntu**: User-friendly and beginner-focused.
- **CentOS/Red Hat**: Enterprise-grade.
- **Debian**: Stable and flexible.
- **Kali Linux**: Designed for penetration testing.

Installation Steps (Ubuntu Example)

1. Download the ISO file.
2. Create a bootable USB using Rufus.
3. Boot from USB and follow the installation wizard.

Chapter 3: Linux Filesystem Hierarchy

Understanding the Filesystem

The Linux filesystem is hierarchical, starting with the root (`/`) directory.

- **`/` (Root)**: The base of the filesystem.
- **`/bin`**: Essential binaries like `ls`, `cp`, `mkdir`.
- **`/etc`**: System configuration files.
- **`/home`**: User home directories (e.g., `/home/username`).
- **`/var`**: Variable files such as logs (`/var/log`) and databases.
- **`/dev`**: Device files for hardware like `/dev/sda`.
- **`/tmp`**: Temporary files; cleared on reboot.
- **`/usr`**: User binaries and libraries.

Practical Example

```
cd /etc
ls
cat /etc/passwd
```

- Navigate to `/etc` to view configuration files.

Chapter 4: Basic Linux Commands

Navigation and Filesystem

- `pwd`: Show current directory.
- `ls`: List files.
- `cd [directory]`: Change directory.
- `mkdir [directory_name]`: Create a new directory.

File Operations

- `cp [source] [destination]`: Copy files.
- `mv [source] [destination]`: Move/rename files.

- `rm [file]`: Remove files or directories.

Chapter 5: File Permissions and Ownership

Understanding Permissions

Permissions format: `rwxr-xr--`.
- `r`: Read
- `w`: Write
- `x`: Execute

Commands

- `chmod [permissions] [file]`: Change permissions.
  Example: `chmod 755 script.sh`

- `chown [user:group] [file]`: Change ownership.

Chapter 6: Processes and System Monitoring

Managing Processes

- `ps`: List running processes.

- `top/htop`: Real-time process monitoring.

- `kill [PID]`: Terminate a process.

Monitoring Resources

- `free -h`: Show memory usage.

- `df -h`: Show disk usage.

- `uptime`: System uptime and load average.

Chapter 7: Package Management

Debian-Based Systems (APT)

- `sudo apt update`: Update package lists.

- `sudo apt install [package]`: Install a package.

Red Hat-Based Systems (YUM/DNF)

- `sudo yum install [package]`: Install a package.

- `sudo yum remove [package]`: Remove a package.

Chapter 8: Networking Basics

Commands

- `ifconfig/ip a`: Display network interfaces.
- `ping [host]`: Test network connectivity.
- `curl [URL]`: Fetch data from URLs.

Chapter 9: Shell Scripting

Why Use Shell Scripts?

Automate repetitive tasks and simplify system management.

Basic Script Example

```
#!/bin/bash
echo "Hello, Linux!"
```

Chapter 10: User and Group Management

User Management

- `adduser [username]`: Add a user.
- `passwd [username]`: Set a password.
- `deluser [username]`: Delete a user.

Group Management

- `groupadd [groupname]`: Add a group.
- `usermod -aG [groupname] [username]`: Add a user to a group.

Chapter 11: System Administration

Scheduling Tasks

- `crontab -e`: Edit cron jobs.
  Example:

  ```
  0 5 * * * /path/to/script.sh
  ```

Chapter 12: Disk Management

Partition Management

- `fdisk [device]`: Manage partitions.
- `mkfs.ext4 [partition]`: Format a partition.

Chapter 13: Linux Security

Firewall

- `ufw`: Simple firewall utility.

```
sudo ufw enable
sudo ufw allow [port]
```

Chapter 14: Advanced Linux Concepts

Systemd

- `systemctl start [service]`: Start a service.
- `systemctl status [service]`: Check the status of a service.

Chapter 15: Linux Commands Cheat Sheet

- **Filesystem**:
  - `find`, `du`, `df`.
- **Processes**:
  - `jobs`, `fg`, `bg`.
- **Networking**:
  - `arp`, `nslookup`.

Chapter 16: Linux for Developers

Text Editors

- Vim: `vim file.txt`.
- Nano: `nano file.txt`.

Chapter 17: Linux for DevOps

- **Containers**: Docker and Kubernetes.
- **Infrastructure as Code**: Terraform and Ansible.

Chapter 18: Troubleshooting Linux Issues

Debugging Tools

- `dmesg`: Kernel logs.
- `strace`: Trace system calls.

Chapter 19: Special Chapter – Top 35 Linux Commands with Practical Examples

1. `ls` – List Directory Contents

- **Purpose**: Displays files and directories in the current directory.
- **Examples**:

```
ls            # Lists files in the current directory.
ls -l         # Lists files with detailed information (permissions,
size, owner).
```

---

2. `cd` – Change Directory

- **Purpose**: Navigate between directories.
- **Examples**:

```
cd /home      # Changes directory to /home.
cd ..         # Moves one level up.
```

---

3. `pwd` – Print Working Directory

- **Purpose**: Shows the current directory.
- **Examples**:

```
pwd           # Displays the absolute path of the current directory.
cd /var && pwd  # Changes to /var and prints the current directory.
```

---

4. `mkdir` – Make Directories

- **Purpose**: Creates new directories.

- **Examples**:

```
mkdir my_dir          # Creates a directory named 'my_dir'.
mkdir -p a/b/c        # Creates nested directories 'a/b/c'.
```

---

5. `rmdir` – Remove Directories

- **Purpose**: Deletes empty directories.

- **Examples**:

```
rmdir my_dir          # Removes an empty directory.
rmdir -p a/b/c        # Removes nested empty directories.
```

---

6. `cp` – Copy Files

- **Purpose**: Copies files and directories.

- **Examples**:

```
cp file1 file2        # Copies 'file1' to 'file2'.
cp -r dir1 dir2       # Copies 'dir1' and its contents to 'dir2'.
```

---

7. `mv` – Move or Rename Files

- **Purpose**: Moves or renames files and directories.

- **Examples**:

```
mv oldname newname    # Renames a file or directory.
mv file1 /tmp/        # Moves 'file1' to the /tmp directory.
```

---

8. `rm` – Remove Files

- **Purpose**: Deletes files and directories.

- **Examples**:

```
rm file1              # Removes 'file1'.
rm -rf my_dir         # Removes 'my_dir' and its contents.
```

---

9. find – Search Files

- **Purpose**: Locates files based on conditions.

- **Examples**:

```
find . -name "*.txt"      # Finds all .txt files in the current
directory.
find /var -size +10M      # Finds files larger than 10MB in /var.
```

---

10. grep – Search Text

- **Purpose**: Searches for patterns in text files.

- **Examples**:

```
grep "error" file.log     # Finds lines containing 'error' in file.log.
grep -r "keyword" /dir    # Searches for 'keyword' recursively in a
directory.
```

---

11. chmod – Change File Permissions

- **Purpose**: Modifies file permissions.

- **Examples**:

```
chmod 755 script.sh       # Sets read, write, and execute for the
owner.
chmod u+x file.sh         # Adds execute permission to the owner.
```

---

12. chown – Change Ownership

- **Purpose**: Changes file owner or group.

- **Examples**:

```
chown user file.txt       # Changes the owner to 'user'.
chown user:group file.txt # Changes owner and group.
```

---

13. `ps` – Process Status

- **Purpose**: Displays running processes.

- **Examples**:

```
ps                      # Lists processes for the current session.
ps aux | grep python    # Finds Python processes.
```

---

14. `top` – Monitor Processes

- **Purpose**: Displays real-time system stats and processes.

- **Examples**:

```
top                     # Opens an interactive process viewer.
top -u username         # Shows processes of a specific user.
```

---

15. `kill` – Terminate Processes

- **Purpose**: Stops running processes.

- **Examples**:

```
kill 1234               # Kills the process with PID 1234.
kill -9 1234            # Force kills the process with PID 1234.
```

---

16. `df` – Disk Usage

- **Purpose**: Shows available disk space.

- **Examples**:

```
df -h                   # Displays disk usage in human-readable format.
df /home                # Checks disk usage for /home.
```

---

17. `du` – Directory Disk Usage

- **Purpose**: Shows space used by directories.

- **Examples**:

```
du -sh .                # Shows total size of the current directory.
du -h /var/log          # Shows size of /var/log files.
```

---

18. `free` – Memory Usage

- **Purpose**: Displays memory statistics.

- **Examples**:

```
free -h                 # Shows memory usage in human-readable format.
free -m                 # Displays memory in MB.
```

---

19. `wget` – Download Files

- **Purpose**: Downloads files from the web.

- **Examples**:

```
wget http://example.com/file.zip  # Downloads file.zip.
wget -c http://example.com/file   # Resumes an interrupted download.
```

---

20. `curl` – Transfer Data

- **Purpose**: Fetches data from URLs.

- **Examples**:

```
curl http://example.com      # Displays webpage content.
curl -O file.zip             # Downloads file.zip.
```

---

21. `tar` – Archive Files

- **Purpose**: Archives and extracts files.

- **Examples**:

```
tar -cvf archive.tar file.txt    # Creates an archive.
tar -xvf archive.tar             # Extracts the archive.
```

---

22. `zip` and `unzip` – Compress Files

- **Purpose**: Zips or unzips files.

- **Examples**:

```
zip archive.zip file.txt        # Compresses file.txt.
unzip archive.zip               # Extracts archive.zip.
```

---

23. `nano` – Text Editor

- **Purpose**: Edits files.

- **Examples**:

```
nano file.txt                   # Opens file.txt for editing.
nano /etc/hosts                 # Edits the hosts file.
```

---

24. `man` – Manual Pages

- **Purpose**: Displays command documentation.

- **Examples**:

```
man ls                          # Shows manual for the 'ls' command.
man chmod                       # Displays details for 'chmod'.
```

---

25. `alias` – Shortcut Commands

- **Purpose**: Creates command shortcuts.

- **Examples**:

```
alias ll='ls -l'                # Creates an alias 'll' for 'ls -l'.
alias rmf='rm -rf'              # Creates an alias for recursive
deletion.
```

---

26. `history` – Command History

- **Purpose**: Displays previously executed commands.

- **Examples**:

```
history                         # Shows the list of previously executed
commands.
history | grep "install"        # Searches for 'install' in command
history.
```

---

27. chmod – Change File Permissions

- **Purpose**: Modifies permissions of a file or directory.

- **Examples**:

```
chmod 755 script.sh             # Sets read, write, and execute
permissions for the owner, and read and execute for others.
chmod u+x file.sh               # Adds execute permission to the user
(owner) of file.sh.
```

---

28. tail – Display the End of a File

- **Purpose**: Displays the last part of files.

- **Examples**:

```
tail file.log                   # Shows the last 10 lines of file.log.
tail -f file.log                # Continuously watches the log file for
new lines.
```

---

29. head – Display the Beginning of a File

- **Purpose**: Shows the first part of files.

- **Examples**:

```
head file.txt                   # Displays the first 10 lines of
file.txt.
head -n 20 file.txt             # Shows the first 20 lines of file.txt.
```

---

30. cut – Remove Sections from Each Line of a File

- **Purpose**: Cuts and extracts sections from each line of input.

- **Examples**:

```
cut -d',' -f1 file.csv          # Extracts the first column from a CSV
file.
cut -c1-5 file.txt              # Extracts the first 5 characters of
each line from file.txt.
```

---

31. `ln` – Create Hard and Symbolic Links

- **Purpose**: Creates links to files.

- **Examples**:

```
ln file1 file2                  # Creates a hard link named file2 to
file1.
ln -s /path/to/file symlink     # Creates a symbolic link to the file.
```

---

32. `ps aux` – Display All Running Processes

- **Purpose**: Displays all running processes on the system.

- **Examples**:

```
ps aux                          # Shows all processes running in the
system.
ps aux | grep "nginx"           # Filters processes related to 'nginx'.
```

---

33. `useradd` – Add a User to the System

- **Purpose**: Creates a new user account.

- **Examples**:

```
useradd username                # Adds a new user called 'username'.
useradd -m -s /bin/bash user    # Creates a user with a home directory
and bash shell.
```

---

34. `usermod` – Modify a User Account

- **Purpose**: Modifies an existing user account.

- **Examples**:

```
usermod -aG sudo username      # Adds user to the sudo group.
usermod -l newname oldname     # Renames a user from 'oldname' to
'newname'.
```

---

35. shutdown – Shutdown or Reboot the System

- **Purpose**: Shuts down or reboots the system.

- **Examples**:

```
shutdown now                   # Immediately shuts down the system.
shutdown -r +5                 # Reboots the system in 5 minutes.
```

Authored by Kunal Namdas