Linux Basic Commands Guide

Welcome to the **Linux Basic Commands Guide**! This repository serves as a comprehensive resource for beginners and those looking to refresh their knowledge of fundamental Linux commands. Whether you're new to Linux or seeking to enhance your command-line proficiency, this guide provides clear explanations, practical examples, and organized content to help you navigate and manage your Linux system effectively.

Table of Contents

---

1. Introduction

The **Linux Basic Commands Guide** is designed to introduce users to the essential commands required for daily tasks in a Linux environment. Mastering these commands will empower you to perform file management, system monitoring, process control, and more with confidence and efficiency.

Who Should Use This Guide?

- **Beginners:** Individuals new to Linux who want to learn the basics.
- **Intermediate Users:** Those looking to reinforce their understanding of fundamental commands.
- **System Administrators:** Professionals who need a quick reference for essential commands.

---

2. Getting Started

Before diving into the commands, ensure you have access to a Linux system. You can use a physical machine, set up a virtual machine, or use cloud-based Linux environments.

Prerequisites:

- **Basic Understanding of Command-Line Interfaces (CLI)**
- **Access to a Terminal Emulator**

How to Use This Guide:

Each section covers specific commands with explanations and examples. It's recommended to practice the commands in a terminal as you progress through the guide to reinforce learning.

---

3. Navigating the File System

Understanding how to navigate the Linux file system is fundamental. This section covers commands to move around directories, list contents, and understand the directory structure.

Key Commands:

- `pwd`: Print Working Directory
- `ls`: List Directory Contents
- `cd`: Change Directory
- `tree`: Display Directory Tree
- `pushd` and `popd`: Navigate Directory Stack

Examples:

```
# Display the current directory
pwd

# List all files and directories in the current directory with detailed information
ls -la

# Change to the /home directory
cd /home

# Navigate to a subdirectory named 'Documents'
cd Documents

# Display the directory structure as a tree
```

```
tree

# Save current directory and switch to /var
pushd /var

# Return to the previous directory
popd
```

---

4. File and Directory Operations

Managing files and directories is a core aspect of Linux administration. Learn how to create, copy, move, and delete files and directories.

Key Commands:

- `cp`: Copy Files and Directories
- `mv`: Move/Rename Files and Directories
- `rm`: Remove Files and Directories
- `mkdir`: Create Directories
- `rmdir`: Remove Directories
- `ln`: Create Links
- `touch`: Create or Update File Timestamps

Examples:

```
# Create a new directory named 'projects'
mkdir projects

# Copy a file from one location to another
cp ~/Downloads/report.txt ~/Documents/projects/

# Copy a directory recursively
cp -r ~/Downloads/photos ~/Documents/projects/

# Move and rename a file
mv ~/Documents/projects/report.txt ~/Documents/projects/summary.txt

# Move a directory to another location
mv ~/Documents/projects ~/backup/

# Remove a file
rm ~/Documents/projects/summary.txt

# Remove a directory and its contents recursively
rm -r ~/backup/projects
```

```
# Create a symbolic link
ln -s /usr/local/bin/myapp ~/bin/myapp

# Create an empty file or update its timestamp
touch ~/Documents/projects/newfile.txt
```

---

5. Viewing and Editing Files

Learn how to view the contents of files and edit them using various text editors available in Linux.

Key Commands:

- `cat`: Concatenate and Display Files
- `less` and `more`: View File Contents with Navigation
- `head` and `tail`: View Beginning and End of Files
- `nano`, `vim`, `gedit`: Text Editors
- `grep`: Search Within Files

Examples:

```
# Display the contents of a file
cat /etc/passwd

# View a large file with pagination using less
less /var/log/syslog

# View the last 10 lines of a file
tail /var/log/syslog

# View the first 20 lines of a file
head -n 20 /var/log/syslog

# Edit a file using nano
nano ~/Documents/projects/summary.txt

# Edit a file using vim
vim ~/Documents/projects/summary.txt

# Open a file with the graphical editor gedit
gedit ~/Documents/projects/summary.txt &

# Search for the term 'error' in a log file
grep "error" /var/log/syslog
```

6. System Information

Gather essential information about your system's hardware and software status.

Key Commands:

- `uname`: System Information
- `top` / `htop`: Process Monitoring
- `df`: Disk Space Usage
- `du`: Directory Space Usage
- `free`: Memory Usage
- `lscpu`: CPU Information
- `lsblk`: Block Device Information
- `lsusb` and `lspci`: USB and PCI Device Information

Examples:

```
# Display kernel and system information
uname -a

# Monitor running processes in real-time using top
top

# Alternatively, using htop for an enhanced interface
htop

# Check disk usage of all mounted filesystems
df -h

# Estimate file and directory space usage
du -sh ~/Documents

# Display memory usage
free -h

# Show detailed CPU information
lscpu

# List all block devices
lsblk

# List all USB devices
lsusb
```

```
# List all PCI devices
lspci
```

---

7. Managing Permissions

Control access to files and directories by managing permissions and ownership.

Key Commands:

- `chmod`: Change File Permissions
- `chown`: Change File Owner
- `chgrp`: Change File Group
- `umask`: Set Default File Creation Permissions

Examples:

```
# Grant execute permission to the user
chmod u+x script.sh

# Set read and write permissions for the group
chmod g+rw data.txt

# Remove write permission from others
chmod o-w report.pdf

# Change the owner of a file to 'alice'
chown alice report.pdf

# Change the group of a directory to 'developers'
chgrp developers /var/www/html

# Change both owner and group
chown alice:developers /var/www/html

# Set default file creation permissions to 755
umask 022
```

---

8. Networking Commands

Manage and troubleshoot network connections and configurations.

Key Commands:

- `ping`: Check Connectivity

- `ifconfig` / `ip`: Network Interfaces
- `ssh`: Secure Shell
- `netstat` / `ss`: Network Statistics
- `curl` and `wget`: Download Files and Interact with Web Services
- `scp`: Secure Copy
- `ftp`: File Transfer Protocol
- `traceroute`: Trace Route to a Host
- `nslookup` and `dig`: DNS Queries

Examples:

```
# Ping a server to check connectivity
ping example.com

# Display all network interfaces and their configurations using ifconfig
ifconfig

# Alternatively, using the 'ip' command
ip addr show

# Connect to a remote server via SSH
ssh user@remote-server.com

# Securely copy a file to a remote server
scp ~/Documents/report.txt user@remote-server.com:/home/user/

# Download a file using curl
curl -O https://example.com/file.zip

# Download a file using wget
wget https://example.com/file.zip

# Display listening ports and active connections using netstat
netstat -tuln

# Alternatively, using the 'ss' command
ss -tuln

# Trace the route to a host
traceroute example.com

# Perform a DNS lookup using nslookup
nslookup example.com

# Perform a DNS lookup using dig
dig example.com
```

---

9. Searching and Sorting

Efficiently search through files and data, and organize output as needed.

Key Commands:

- `grep`: Search Text
- `find`: Find Files
- `sort`: Sort Lines of Text
- `uniq`: Report or Filter Unique Lines
- `awk`: Pattern Scanning and Processing Language
- `sed`: Stream Editor for Filtering and Transforming Text

Examples:

```
# Search for the term 'error' in a log file
grep "error" /var/log/syslog

# Search recursively for 'TODO' in all .c files within a directory
grep -r "TODO" ~/projects/*.c

# Find all .txt files in the home directory
find ~/ -type f -name "*.txt"

# Sort the contents of a file alphabetically
sort unsorted.txt > sorted.txt

# Remove duplicate lines from a sorted file
uniq sorted.txt unique.txt

# Print the second and third columns from a file
awk '{print $2, $3}' data.txt

# Substitute 'foo' with 'bar' in a file and display the result
sed 's/foo/bar/g' input.txt
```

---

10. Package Management

Install, update, and manage software packages on your Linux system.

Key Commands:

- `apt` (Debian/Ubuntu)

- `yum` / `dnf` (CentOS/Fedora)
- `pacman` (Arch Linux)
- `dpkg`: Debian Package Management
- `rpm`: Red Hat Package Management
- `snap`: Package Management for Snaps
- `flatpak`: Package Management for Flatpaks

Examples (Using APT):

```
# Update the package list
sudo apt update

# Upgrade all installed packages to their latest versions
sudo apt upgrade

# Install a new package, e.g., git
sudo apt install git

# Remove an installed package
sudo apt remove git

# Search for a package
apt search nginx

# Display information about a package
apt show nginx

# Clean up unused packages and dependencies
sudo apt autoremove

# List all installed packages
apt list --installed
```

Examples (Using YUM/DNF):

```
# Install a package using yum
sudo yum install httpd

# Install a package using dnf
sudo dnf install httpd

# Remove a package
sudo dnf remove httpd

# Update all packages
sudo dnf update
```

```
# Search for a package
dnf search nginx
```

---

11. Process Management

Monitor and control running processes on your system.

Key Commands:

- `ps`: Report a Snapshot of Current Processes
- `top` / `htop`: Interactive Process Viewer
- `kill`: Terminate Processes
- `pkill` and `killall`: Kill Processes by Name
- `bg` and `fg`: Manage Background and Foreground Processes
- `nice` and `renice`: Set Process Priority

Examples:

```
# Display all current processes
ps aux

# Monitor running processes in real-time using top
top

# Alternatively, using htop for an enhanced interface
htop

# Terminate a process using its PID
kill 1234

# Force terminate a process
kill -9 1234

# Kill all processes with the name 'firefox'
killall firefox

# Move a running job to the background
bg %1

# Bring a background job to the foreground
fg %1

# Start a process with a lower priority
nice -n 10 ./long_running_script.sh
```

```
# Change the priority of an existing process
sudo renice -n -5 -p 1234
```

---

12. Disk Management

Manage disk partitions, filesystems, and storage devices.

Key Commands:

- `fdisk`: Partition Table Manipulation for MBR
- `gdisk`: Partition Table Manipulation for GPT
- `lsblk`: List Block Devices
- `blkid`: Locate/Print Block Device Attributes
- `mount` and `umount`: Mount and Unmount Filesystems
- `df`: Disk Space Usage
- `du`: Directory Space Usage
- `parted`: Advanced Partitioning Tool

Examples:

```
# List all block devices and their partitions
lsblk

# Display detailed information about block devices
blkid

# Create a new partition table (MBR) on /dev/sdb
sudo fdisk /dev/sdb

# Create a new partition table (GPT) on /dev/sdb
sudo gdisk /dev/sdb

# Mount a partition to /mnt
sudo mount /dev/sdb1 /mnt

# Unmount the filesystem
sudo umount /mnt

# Check disk space usage
df -h

# Estimate directory space usage
du -sh ~/Downloads
```

```
# Use parted to create a new partition
sudo parted /dev/sdb
(parted) mklabel gpt
(parted) mkpart primary ext4 0% 100%
(parted) quit
```

---

13. User and Group Management

Create, modify, and manage user accounts and groups on your system.

Key Commands:

- `adduser` and `useradd`: Add New Users
- `deluser` and `userdel`: Delete Users
- `usermod`: Modify User Accounts
- `passwd`: Change User Passwords
- `groupadd`: Add New Groups
- `groupdel`: Delete Groups
- `groups`: Display User Groups
- `id`: Display User and Group IDs

Examples:

```
# Add a new user named 'john'
sudo adduser john

# Delete a user named 'john' and their home directory
sudo deluser --remove-home john

# Modify a user's shell to /bin/zsh
sudo usermod -s /bin/zsh john

# Change a user's password
sudo passwd john

# Add a new group named 'developers'
sudo groupadd developers

# Add a user to a group
sudo usermod -aG developers john

# Remove a user from a group
sudo gpasswd -d john developers

# Display the groups a user belongs to
```

```
groups john

# Display user and group IDs
id john
```

---

14. Miscellaneous Commands

A collection of useful commands that don't fit neatly into other categories but are essential for various tasks.

Key Commands:

- `echo`: Display a Line of Text
- `date`: Display or Set System Date and Time
- `cal`: Display a Calendar
- `history`: Show Command History
- `alias` and `unalias`: Create and Remove Aliases
- `wget` and `curl`: Download Files and Interact with Web Services
- `tar`: Archive Files
- `gzip` and `gunzip`: Compress and Decompress Files
- `chmod` and `chown`: Change File Permissions and Ownership
- `man`: Display Manual Pages

Examples:

```
# Display a line of text
echo "Hello, Linux!"

# Display the current date and time
date

# Display a calendar for the current month
cal

# Show command history
history

# Create an alias for listing files in long format
alias ll='ls -la'

# Remove an alias
unalias ll

# Download a file using wget
wget https://example.com/file.zip
```

```
# Download a file using curl and save it with a specific name
curl -o myfile.zip https://example.com/file.zip

# Create a tarball of the 'projects' directory
tar -cvf projects.tar ~/Documents/projects

# Extract a tarball
tar -xvf projects.tar

# Compress a file using gzip
gzip report.txt

# Decompress a gzip file
gunzip report.txt.gz

# Display the manual page for the 'ls' command
man ls
```

---

15. Tips and Best Practices

Enhance your command-line efficiency and maintain system security with these tips and best practices.

- **Use Tab Completion:** Press `Tab` while typing commands or filenames to auto-complete.
- **Leverage Command History:** Use the `Up` and `Down` arrow keys to navigate through previous commands.
- **Understand Permissions:** Properly manage file permissions to maintain system security.
- **Regularly Update Your System:** Keep your system and packages up-to-date to ensure security and stability.
- **Use `man` Pages:** Access detailed documentation for commands using `man command_name` (e.g., `man ls`).
- **Redirect Output:** Use `>` to redirect output to a file and `>>` to append.

  ```
  ls -la > directory_contents.txt
  ```

- **Pipe Commands:** Use `|` to pass the output of one command as input to another.

  ```
  ps aux | grep firefox
  ```

- **Use Sudo Wisely:** Only use `sudo` when necessary to prevent unintended system changes.
- **Create Aliases:** Simplify long or frequently used commands with aliases.

  ```
  alias update='sudo apt update && sudo apt upgrade'
  ```

- **Practice Regularly:** Consistent use of commands reinforces memory and improves proficiency.