Top Linux Commands You Must Know

Introduction

Using Linux commands regularly? Here’s a concise guide to the top Linux commands every user should know. These commands are essential for navigating and managing a Linux system effectively.

Prerequisites

These commands are demonstrated on an Ubuntu server, but they are applicable to any modern Linux distribution.

Essential Linux Commands

- **ls**: List directory contents.
- **pwd**: Print the current working directory.
- **cd**: Change the current directory.
- **mkdir**: Create a new directory.
- **mv**: Move or rename files.
- **cp**: Copy files or directories.
- **rm**: Remove files or directories.
- **touch**: Create an empty file or update the timestamp of an existing file.
- **ln**: Create hard and symbolic links.
- **clear**: Clear the terminal screen.
- **cat**: Concatenate and display file content.
- **echo**: Display a line of text.
- **less**: View file content one screen at a time.
- **man**: Display the manual page for a command.
- **uname**: Print system information.
- **whoami**: Display the current username.
- **tar**: Archive files.
- **grep**: Search text using patterns.

- **head**: Display the first lines of a file.
- **tail**: Display the last lines of a file.
- **diff**: Compare files line by line.
- **cmp**: Compare two files byte by byte.
- **comm**: Compare two sorted files line by line.
- **sort**: Sort lines of text files.
- **export**: Set environment variables.
- **zip/unzip**: Compress and decompress files.
- **ssh**: Securely connect to remote servers.
- **service**: Manage system services.
- **ps**: Display current processes.
- **kill/killall**: Terminate processes.
- **df**: Report file system disk space usage.
- **mount**: Mount a file system.
- **chmod**: Change file permissions.
- **chown**: Change file owner and group.
- **ifconfig**: Configure network interfaces.
- **traceroute**: Trace the route packets take to a network host.
- **wget**: Download files from the web.
- **ufw/iptables**: Manage firewall rules.
- **apt/pacman/yum/rpm**: Package managers for different distributions.
- **sudo**: Execute a command as another user.
- **cal**: Display a calendar.
- **alias**: Create command shortcuts.
- **dd**: Convert and copy files.
- **whereis**: Locate the binary, source, and manual page files for a command.
- **whatis**: Display a one-line description of a command.
- **top**: Display tasks and system status.
- **useradd/usermod**: Add or modify user accounts.
- **passwd**: Update user’s authentication tokens.

Detailed Command Descriptions

ls

The `ls` command lists files and directories in the current working directory. It supports various options to display additional information, such as `-l` for a long listing format.

pwd

The `pwd` command prints the full path of the current working directory.

cd

The `cd` command changes the current directory. For example, `cd /etc` navigates to the `/etc` directory.

mkdir

The `mkdir` command creates a new directory. For example, `mkdir new_folder` creates a directory named `new_folder`.

mv

The `mv` command moves or renames files and directories. For example, `mv old_name new_name` renames a file or directory.

cp

The `cp` command copies files or directories. For example, `cp source_file destination_file` copies `source_file` to `destination_file`.

rm

The `rm` command removes files or directories. Use the `-r` option to remove directories recursively. For example, `rm -r folder_name` removes a directory and its contents.

touch

The `touch` command creates an empty file or updates the timestamp of an existing file. For example, `touch new_file` creates a file named `new_file`.

ln

The `ln` command creates hard and symbolic links. Use the `-s` option for symbolic links. For example, `ln -s target link_name` creates a symbolic link.

clear

The `clear` command clears the terminal screen.

cat

The `cat` command concatenates and displays file content. For example, `cat file_name` displays the content of `file_name`.

echo

The `echo` command displays a line of text. For example, `echo "Hello, World!"` prints "Hello, World!" to the terminal.

less

The `less` command allows you to view file content one screen at a time. For example, `less file_name` opens `file_name` in a scrollable view.

man

The `man` command displays the manual page for a command. For example, `man ls` shows the manual for the `ls` command.

uname

The `uname` command prints system information. Use the `-a` option to display all available information. For example, `uname -a`.

whoami

The `whoami` command displays the current username.

tar

The `tar` command archives files. Use the `-c` option to create an archive and the `-x` option to extract it. For example, `tar -cvf archive.tar file_name` creates an archive, and `tar -xvf archive.tar` extracts it.

grep

The `grep` command searches text using patterns. For example, `grep "search_term" file_name` searches for "search_term" in `file_name`.

head

The `head` command displays the first lines of a file. For example, `head -n 10 file_name` shows the first 10 lines of `file_name`.

tail

The `tail` command displays the last lines of a file. For example, `tail -n 10 file_name` shows the last 10 lines of `file_name`.

diff

The `diff` command compares files line by line. For example, `diff file1 file2` shows the differences between `file1` and `file2`.

cmp

The `cmp` command compares two files byte by byte. For example, `cmp file1 file2` shows the first byte that differs between `file1` and `file2`.

comm

The `comm` command compares two sorted files line by line. For example, `comm file1 file2` shows lines unique to each file and lines common to both.

sort

The `sort` command sorts lines of text files. For example, `sort file_name` sorts the lines in `file_name`.

export

The `export` command sets environment variables. For example, `export VAR=value` sets the environment variable `VAR` to `value`.

zip/unzip

The `zip` command compresses files, and the `unzip` command decompresses them. For example, `zip archive.zip file_name` compresses `file_name` into `archive.zip`, and `unzip archive.zip` extracts it.

ssh

The `ssh` command securely connects to remote servers. For example, `ssh user@host` connects to `host` as `user`.

service

The `service` command manages system services. For example, `service ssh status` checks the status of the SSH service.

ps

The `ps` command displays current processes. For example, `ps -ef` shows all running processes.

kill/killall

The `kill` command terminates processes by PID. For example, `kill 1234` terminates the process with PID 1234. The `killall` command terminates processes by name. For example, `killall process_name`.

df

The `df` command reports file system disk space usage. For example, `df -h` shows disk space usage in human-readable format.

mount

The `mount` command mounts a file system. For example, `mount /dev/sdX /mnt` mounts the device `/dev/sdX` to the directory `/mnt`.

chmod

The `chmod` command changes file permissions. For example, `chmod +x file_name` makes `file_name` executable.

chown

The `chown` command changes file owner and group. For example, `chown user:group file_name` changes the owner and group of `file_name`.

ifconfig

The `ifconfig` command configures network interfaces. For example, `ifconfig eth0` shows the configuration of the `eth0` interface.

traceroute

The `traceroute` command traces the route packets take to a network host. For example, `traceroute example.com` shows the route to `example.com`.

wget

The `wget` command downloads files from the web. For example, `wget http://example.com/file` downloads `file` from `example.com`.

ufw/iptables

The `ufw` and `iptables` commands manage firewall rules. For example, `ufw allow 80` allows traffic on port 80, and `iptables -A INPUT -p tcp --dport 80 -j ACCEPT` does the same using `iptables`.

apt/pacman/yum/rpm

These are package managers for different distributions. For example, `apt install package_name` installs a package on Debian-based systems, `pacman -S package_name` on Arch-based systems, `yum install package_name` on Red Hat-based systems, and `rpm -i package_name` on RPM-based systems.

sudo

The `sudo` command executes a command as another user. For example, `sudo command` runs `command` with superuser privileges.

cal

The `cal` command displays a calendar. For example, `cal` shows the current month's calendar.

alias

The `alias` command creates command shortcuts. For example, `alias ll='ls -l'` creates an alias `ll` for `ls -l`.

dd

The `dd` command converts and copies files. For example, `dd if=/dev/sdX of=/dev/sdY` copies data from `sdX` to `sdY`.

whereis

The `whereis` command locates the binary, source, and manual page files for a command. For example, `whereis ls` shows the locations of the `ls` command.

whatis

The `whatis` command displays a one-line description of a command. For example, `whatis ls` describes the `ls` command.

top

The `top` command displays tasks and system status. For example, `top` shows a real-time view of system processes.

useradd/usermod

The `useradd` command adds a new user, and the `usermod` command modifies an existing user. For example, `useradd new_user` creates a new user, and `usermod -aG group user` adds `user` to `group`.

passwd

The `passwd` command updates user's authentication tokens. For example, `passwd user` changes the password for `user`. This guide provides a quick reference to essential Linux commands. Mastering these commands will enhance your efficiency and capability in managing Linux systems.

Follow us if you haven't