LINUX

Section 1: Basic Linux Commands

Command	Description->	Example	Explanation
`ls`	List directory contents	`ls`	Lists files and directories in the current directory
`cd`	Change directory	`cd /path/to/directory`	Changes the current directory to the specified path
`pwd`	Print working directory	`pwd`	Displays the full path of the current directory
`mkdir`	Create a new directory	`mkdir newdir`	Creates a directory named "newdir"
`rmdir`	Remove a directory	`rmdir olddir`	Removes the directory named "olddir" if it is empty
`rm -r`	Remove directory recursively	`rm -r olddir`	Removes the directory named "olddir" and its contents
`touch`	Create a new file	`touch newfile`	Creates an empty file named "newfile"
`echo`	Print text to the terminal or file	`echo "Hello" > file.txt`	Writes "Hello" to file.txt, creating it if it doesn't exist
`cat`	Concatenate and display file contents	`cat file.txt`	Displays the contents of file.txt
`cp`	Copy files or directories	`cp file1 file2`	Copies file1 to file2
`mv`	Move or rename files or directories	`mv file1 file2`	Moves or renames file1 to file2
`clear`	Clears the screen		
`exit`	Exit the shell or current session		
`rm`	Remove files or directories	`rm file.txt`	Deletes file.txt

Section 2: Getting Help at Command Line

Command	Description	Example	Explanation
`man`	Display the manual for a command	`man ls`	Opens the manual page for the `ls` command, detailing its usage and options man -k=>Search for manual pages containing the keyword
`info`	Display information about a command	`info ls`	Opens the info page for the `ls` command, providing detailed documentation
`--help`	Display help for a command	`ls --help`	Displays a brief help message with available options for the `ls` command
`whatis`	Brief description of a command	`whatis ls`	Provides a one-line description of the `ls` command
`apropos`	Search for commands related to a keyword	`apropos list`	Searches for commands related to the keyword "list"

Section 3: Working with Directories

Command	Description	Example	Explanation
`cd`	Change directory	`cd /var/log`	Changes the current directory to /var/log
`mkdir`	Create a new directory	`mkdir projects`	Creates a directory named "projects"
`rmdir`	Remove a directory	`rmdir projects`	Removes the "projects" directory if it is empty
`pwd`	Print working directory	`pwd`	Displays the full path of the current directory
`ls`	List directory contents	`ls -l`	Lists files in the current directory with detailed information

. – This directory

. . -The parent directory

$Path – determines command search path

./command -Execute command in this dir

rm -rf directory – recursively removes the diectory

Section 4: Listing Files and Understanding Its Output

Command	Description	Example	Explanation
`ls -l`	List files in long format	`ls -l`	Displays detailed information about files, including permissions, ownership, size, and modification date
`ls -F`	To reveal file type		/=>directory , @=>Link ,*=>Executable
`ls -r`	Reverse order		
`ls -a`	List all files including hidden	`ls -a`	Shows all files in the directory, including hidden files (those starting with a dot)
`ls -lh`	List files in human-readable format	`ls -lh`	Displays file sizes in a human-readable format (e.g., KB, MB)
`ls -ltr`	List files sorted by modification time, newest last	`ls -ltr`	Lists files sorted by modification time, with the newest files at the bottom
`ls -R`	List files recursively	`ls -R`	Lists all files and directories recursively, showing the contents of all subdirectories
`ls -S`	Sort files by size	`ls -S`	Lists files sorted by size, with the largest files first
`ls -t`	Sort files by modification time	`ls -t`	Lists files sorted by modification time, with the newest files first
`ls -r`	Reverse the order of the sort	`ls -r`	Lists files in reverse order of the sort, for example, `ls -lr` lists files in long format with the oldest files first
`Tree -C`	Colorise output		
`tree -d`	List directories only		

Section 5: File and Directory Permission

Command	Description	Example	Explanation
`chmod`	Change file mode (permissions)	`chmod 755 file.txt`	Sets permissions for file.txt to 755 (owner: read/write/execute, group: read/execute, others: read/execute)
`chown`	Change file owner and group	`chown user:group file.txt`	Changes the owner and group of file.txt to "user" and "group"
`chgrp`	Change group ownership	`chgrp group file.txt`	Changes the group of file.txt to "group"
`umask`	Set default file permissions	`umask 022`	Sets default permissions for newly created files (755)
`ls -l`	List files with permissions	`ls -l`	Displays file permissions in the listing

Section 6: Viewing Files and the Nano Editor

Command	Description	Example	Explanation
`cat`	Display file contents	`cat file.txt`	Displays the entire contents of file.txt
`more`	View file contents one screen at a time	`more file.txt`	Displays file.txt one screen at a time, allowing forward movement
`less`	View file contents with backward movement	`less file.txt`	Displays file.txt with navigation capabilities (forward and backward)
`nano`	Simple text editor	`nano file.txt`	Opens file.txt in the nano editor for editing

Section 7: Editing Files in vi

Command	Description	Example	Explanation
`vi`	Text editor with two modes: command and insert	`vi file.txt`	Opens file.txt in the vi editor, starting in command mode
`vim`	Improved version of vi	`vim file.txt`	Opens file.txt in the vim editor, which has additional features over vi
Insert Mode	Mode to insert text	`i`	Enter insert mode to add text
Command Mode	Mode to execute commands	`Esc`	Exit to command mode to execute commands
`:w`	Save file in vi/vim	`:w`	Saves changes made to the file
`:q`	Quit vi/vim	`:q`	Quits the editor without saving changes
`:wq`	Save and quit vi/vim	`:wq`	Saves changes and quits the editor
`:q!`	Quit vi/vim without saving	`:q!`	Discards changes and quits the editor

Section 8: Deleting, Copying, Moving and Renaming

Command	Description	Example	Explanation
`rm`	Remove files or directories	`rm file.txt`	Deletes file.txt
`rm -r`	Remove directory recursively	`rm -r dir`	Deletes the directory "dir" and its contents

`rm -i`	Prompt before every removal	`rm -i file.txt`	Asks for confirmation before deleting file.txt
`rm -f`	Force remove files or directories	`rm -f file.txt`	Deletes file.txt without prompting, ignoring non-existent files and never prompting
`cp`	Copy files or directories	`cp file1 file2`	Copies file1 to file2
`cp -r`	Copy directories recursively	`cp -r dir1 dir2`	Copies directory dir1 and its contents to dir2
`mv`	Move or rename files or directories	`mv file1 file2`	Moves or renames file1 to file2
`mv -i`	Prompt before overwrite	`mv -i file1 file2`	Prompts before overwriting file2 with file1
`rename`	Rename multiple files	`rename 's/old/new/' *`	Renames all files by replacing 'old' with 'new'

Section 9: Wildcards

Wildcard	Description	Example	Explanation
`*`	Matches any number of characters	`ls *.txt`	Lists all .txt files in the directory
`?`	Matches a single character	`ls file?.txt`	Lists files like file1.txt, file2.txt
`[]`	Matches any one of the enclosed characters	`ls file[1-3].txt`	Lists files file1.txt, file2.txt, file3.txt
`{}`	Matches any of the patterns within braces	`ls file{1,2}.txt`	Lists files file1.txt, file2.txt

Section 10: Input and Output Redirection

Command	Description	Example	Explanation
`>`	Redirect output to a file, overwriting it	`echo "Hello" > file.txt`	Writes "Hello" to file.txt, creating or overwriting it
`>>`	Append output to a file	`echo "Hello" >> file.txt`	Appends "Hello" to the end of file.txt
`<`	Redirect input from a file	`wc -l < file.txt`	Counts the lines in file.txt
`` ` ``	`` ` ``	Pipe output to another command	`` `ls ``

Section 11: Searching Files and Directories

Command	Description	Example	Explanation
`find`	Search for files and directories	`find /home -name "*.txt"`	Finds all .txt files in /home and its subdirectories
`locate`	Quickly find files by name	`locate file.txt`	Searches the prebuilt database for file.txt
`grep`	Search text using patterns	`grep "hello" file.txt`	Searches for the string "hello" in file.txt
`grep -r`	Recursively search directories	`grep -r "hello" /home`	Recursively searches for "hello" in /home

Section 12: File Compression and Archiving

Command	Description	Example	Explanation
`tar`	Archive files	`tar -cvf archive.tar file1 file2`	Creates an archive named "archive.tar" containing file1 and file2
`tar -xvf`	Extract files from an archive	`tar -xvf archive.tar`	Extracts files from "archive.tar"
`tar -tvf`	List contents of an archive	`tar -tvf archive.tar`	Lists the contents of "archive.tar"
`gzip`	Compress files	`gzip file.txt`	Compresses file.txt and creates file.txt.gz
`gunzip`	Decompress files	`gunzip file.txt.gz`	Decompresses file.txt.gz and restores file.txt
`zip`	Compress files into a zip archive	`zip archive.zip file1 file2`	Creates a zip archive "archive.zip" containing file1 and file2
`unzip`	Extract files from a zip archive	`unzip archive.zip`	Extracts files from "archive.zip"

Section 13: Transferring and Copying Files Over Network

Command	Description	Example	Explanation
`scp`	Secure copy files between hosts	`scp file.txt user@remote:/path/to/destination/`	Copies file.txt to a remote server at the specified path
`rsync`	Sync files and directories between hosts	`rsync -avz file.txt user@remote:/path/to/destination/`	Synchronizes file.txt to the remote server, preserving permissions and compressing data
`sftp`	Secure file transfer protocol	`sftp user@remote`	Opens an SFTP session to the remote server for file transfers
`ftp`	File Transfer Protocol (less secure)	`ftp remote`	Connects to a remote FTP server to transfer files

Section 14: Customizing Shell Prompt

Command	Description	Example	Explanation
`PS1`	Customize the shell prompt	`export PS1="\u@\h:\w\$ "`	Sets the shell prompt to display username, hostname, and current working directory
`PROMPT_COMMAND`	Command to execute before the prompt	`export PROMPT_COMMAND='echo -n "$(date)"'`	Executes the `date` command before displaying the prompt
`.bashrc`	Shell configuration file for bash	`source ~/.bashrc`	Loads the settings from the .bashrc file, which may include prompt customizations

Section 15: Shell Aliases

Command	Description	Example	Explanation
`alias`	Create a shortcut for a command	`alias ll='ls -l'`	Defines `ll` as a shortcut for `ls -l`
`unalias`	Remove an alias	`unalias ll`	Removes the alias `ll`

Section 16: Environment Variables

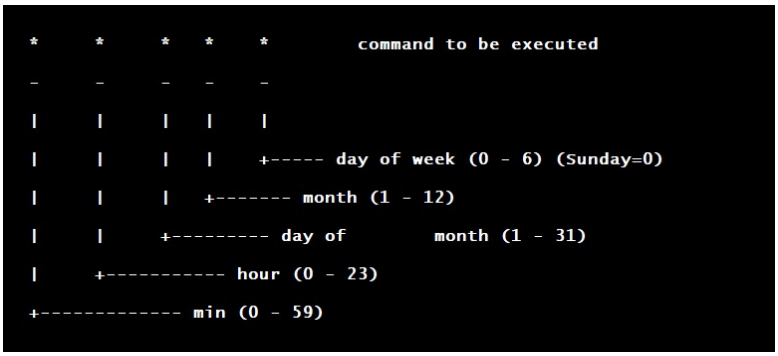
Command	Description	Example	Explanation
`export`	Set or export environment variables	`export PATH=$PATH:/new/path`	Adds `/new/path` to the system PATH variable
`env`	Display all environment variables	`env`	Lists all currently set environment variables
`printenv`	Print environment variable values	`printenv PATH`	Shows the value of the PATH environment variable

Section 17: Processes and Job Control

Command	Description	Example	Explanation
`ps`	Display current processes	`ps aux`	Lists all running processes with detailed information
`top`	Display a real-time view of processes	`top`	Provides a dynamic, real-time view of system processes
`jobs`	List background jobs	`jobs`	Lists all jobs that are running in the background
`bg`	Resume a suspended job in the background	`bg %1`	Resumes the job with ID 1 in the background
`fg`	Bring a job to the foreground	`fg %1`	Brings the job with ID 1 to the foreground
`kill`	Terminate a process by PID	`kill 1234`	Sends a SIGTERM signal to process with PID 1234
`killall`	Terminate processes by name	`killall processname`	Sends a SIGTERM signal to all processes named `processname`
`pkill`	Terminate processes by pattern	`pkill pattern`	Sends a SIGTERM signal to all processes matching the pattern

Section 18: Scheduling Repeated Jobs with Cron

Command	Description	Example	Explanation
`crontab`	Manage cron jobs	`crontab -e`	Edit the current user's cron jobs
`crontab -l`	List cron jobs	`crontab -l`	List all scheduled cron jobs for the current user
`crontab -r`	Remove all cron jobs	`crontab -r`	Remove all cron jobs for the current user
`* * * * * command`	Cron job timing syntax	`0 5 * * * /path/to/command`	Executes `/path/to/command` at 5 AM every day



Section 19: Switching Users and Running Commands

Command	Description	Example	Explanation
`su`	Switch user	`su - username`	Switches to the user `username` with a login shell
`sudo`	Execute commands as another user	`sudo ls /root`	Executes `ls /root` with root privileges
`sudo -i`	Start a root shell	`sudo -i`	Opens an interactive root shell

Section 20: Shell History and Tab Completion

Command	Description	Example	Explanation
`history`	Display command history	`history`	Lists the command history of the current shell session
`!number`	Execute command from history	`!10`	Repeats the command at history position 10
`TAB`	Autocomplete commands and filenames	`ls /u[TAB]`	Completes the path to `/usr/` if it exists

Section 21: Installing Software

Command	Description	Example	Explanation
`apt-get`	Debian-based package manager	`sudo apt-get install package`	Installs the specified package on Debian-based systems
`yum`	Red Hat-based package manager	`sudo yum install package`	Installs the specified package on Red Hat-based systems
`dnf`	Next-generation Red Hat package manager	`sudo dnf install package`	Installs the specified package on Fedora-based systems
`rpm`	RPM package manager	`sudo rpm -ivh package.rpm`	Installs the RPM package file

Section 22: Linux Boot Process

Command	Description	Example	Explanation
`dmesg`	Display boot messages	`dmesg	less`	
`systemctl`	Manage system services	`systemctl status`	Displays the status of system services
`journalctl`	View system logs	`journalctl -b`	Shows logs from the current boot
`telinit`	Changes system runlevel or state	`telinit runlevel`	telinit 0 – shutdown system telinit 6-reboot

Section 23: System Logging

Command	Description	Example	Explanation
`journalctl`	Query and view system logs	`journalctl`	Displays the system logs
`tail`	View the end of a file	`tail /var/log/syslog`	Shows the last lines of syslog
`logrotate`	Rotate, compress, and mail system logs	`logrotate /etc/logrotate.conf`	Rotates logs as specified in the configuration file
`logrotate`	Manages the rotation and compression of log files.	`logrotate /etc/logrotate.conf`	Rotates, compresses, and manages old log files based on the configuration in logrotate.conf.
`syslogd`	System logging daemon that handles logging requests from various services.	`syslogd -d`	Starts the syslog daemon with debugging output enabled. Typically configured through /etc/syslog.conf.
`rsyslogd`	Enhanced version of syslogd; manages system log messages with additional features.	`rsyslogd -n`	Starts the rsyslog daemon in the foreground for troubleshooting. Can be configured using /etc/rsyslog.conf.
`logger`	Adds entries to the system log.	`logger "This is a test log message"`	Writes a custom log message to the syslog, useful for logging from scripts or commands.
`systemctl status`	Displays the status of systemd services, including their logs.	`systemctl status sshd`	Shows the status and recent log entries for the SSH service.
`systemctl journal`	Displays logs for systemd services from the journal.	`systemctl --no-pager`	Views the logs of systemd services, allowing for detailed inspection.

Section 24: Disk Management

Command	Description	Example	Explanation
`df`	Display disk space usage	`df -h`	Shows disk space usage in a human-readable format
`du`	Display disk usage of files and directories	`du -sh /path`	Shows the disk usage of the directory at /path
`fdisk`	Partition table manipulator	`fdisk /dev/sda`	Opens fdisk to manage partitions on /dev/sda

Section 25: LVM - Logical Volume Manager

Command	Description	Example	Explanation
`lvcreate`	Create a logical volume	`lvcreate -n mylv -L 10G vgname`	Creates a 10GB logical volume named "mylv" in volume group "vgname"
`vgcreate`	Create a volume group	`vgcreate vgname /dev/sda1`	Creates a volume group named "vgname" with /dev/sda1
`pvcreate`	Create a physical volume	`pvcreate /dev/sda1`	Initializes /dev/sda1 for use by LVM

Section 26: Managing Users and Groups

Command	Description	Example	Explanation
`useradd`	Add a new user	`useradd username`	Creates a new user named "username"
`usermod`	Modify user account	`usermod -aG group username`	Adds user "username" to the "group" group
`userdel`	Delete a user	`userdel username`	Deletes the user "username"
`groupadd`	Add a new group	`groupadd groupname`	Creates a new group named "groupname"
`groupdel`	Delete a group	`groupdel groupname`	Deletes the group "groupname"

Section 27: TCP/IP Networking for Linux System

Command	Description	Example	Explanation
`ip addr`	Show IP addresses	`ip addr show`	Displays IP addresses and network interfaces
`ip route`	Show routing table	`ip route`	Displays the routing table
`ifconfig`	Display network interfaces	`ifconfig`	Displays network interfaces and their configurations
`netstat`	Network statistics	`netstat -tuln`	Shows active network connections and listening ports

Section 28: Networking

Command	Description	Example	Explanation
`ping`	Test network connectivity	`ping google.com`	Sends ICMP echo requests to google.com to test connectivity `ping -c 3 google.com =>3 packets send`
`traceroute`	Trace the route packets take	`traceroute google.com`	Shows the path packets take to reach google.com
`nmap`	Network exploration tool	`nmap 192.168.1.1`	Scans the network address 192.168.1.1 for open ports

Section 29: Network Troubleshooting

Command	Description	Example	Explanation
`ping`	Test connectivity to a host	`ping -c 4 example.com`	Sends 4 ICMP packets to example.com to test connectivity
`traceroute`	Trace the path of packets	`traceroute example.com`	Shows the route packets take to reach example.com
`netstat`	Network statistics	`netstat -an`	Displays all network connections and listening ports
`tcpdump`	Captures and displays all packets on the default network interface.	`tcpdump`	Basic command to start capturing packets on the default interface.
`ss`	Utility to investigate sockets	`ss -tuln`	Shows TCP and UDP sockets with detailed information

Section 30: Special Modes

Command	Description	Example	Explanation
`setuid`	Allows an executable file to run with the privileges of the file owner, not the user executing it.	chmod u+s /path/to/file	When set on a program like /usr/bin/passwd, any user executing the program does so with the file owner's privileges (usually root), enabling them to change passwords.
setgid	Causes an executable file to run with the group privileges of the file, or ensures directories inherit the group ownership for new files and directories.	chmod g+s /path/to/file_or_directory	For an executable, the program runs with the group permissions. For directories, all new files and directories created inside inherit the directory's group ownership.
Sticky Bit	Prevents users from deleting or renaming files in a directory unless they own the files.	chmod +t /path/to/directory	Commonly used in directories like /tmp, where many users can write files, but cannot delete or rename files owned by others, even though they have write access.
`single`	Single-user mode	`systemctl isolate rescue.target`	Switches to single-user mode for system maintenance
`rescue`	Rescue mode	`systemctl isolate rescue.target`	Boots into rescue mode for system recovery

Section 31: Shell Scripting

Command	Description	Example	Explanation
`bash`	Execute a shell script	`bash script.sh`	Executes the shell script "script.sh"
`chmod +x`	Make a script executable	`chmod +x script.sh`	Makes "script.sh" executable
`#!/bin/bash`	Shebang line for bash scripts	`#!/bin/bash`	Indicates that the script should be run with bash

Section 32: Finding Files and Directories

Command	Description	Example	Explanation
`find`	Search for files and directories	`find /home -name "*.txt"`	Finds all .txt files in the /home directory
`locate`	Quickly find files by name	`locate file.txt`	Finds file.txt using a prebuilt database
`which`	Locate a command	`which ls`	Shows the path to the executable for `ls`

Section 33: Linux Directories

Directory	Description	Typical Contents
/	Root directory; the top-level directory.	All other directories are subdirectories of /.
/bin	Essential user binaries and commands.	Common commands like ls, cp, mv, cat.
/boot	Static files for bootloader.	Kernel images, bootloader configuration files (e.g., vmlinuz, initrd.img).
/dev	Device files; represents hardware devices.	Device files like /dev/sda, /dev/tty0, /dev/null.
/etc	System-wide configuration files.	Configuration files for system services and applications (e.g., passwd, fstab, network).
/home	User home directories.	Individual user directories (e.g., /home/user1, /home/user2).
/lib	Shared libraries required by binaries in /bin and /sbin.	System libraries and kernel modules.
/lib64	Shared libraries for 64-bit systems.	64-bit system libraries.
/media	Mount point for removable media.	Mount points for USB drives, CDs, DVDs.
/mnt	Mount point for temporarily mounted filesystems.	Temporary mounts (e.g., external drives).
/opt	Optional application software packages.	Add-on application software (e.g., /opt/application).
/proc	Virtual filesystem providing process and system information.	Kernel and process information (e.g., /proc/cpuinfo, /proc/meminfo).
/root	Home directory for the root user.	Root user's personal files and settings.
/run	Runtime data; information about the system since the last boot.	PID files, lock files.
/srv	Data for services provided by the system.	Data for services (e.g., /srv/www for web data).
/sys	Virtual filesystem for kernel and device information.	Kernel and device information (e.g., /sys/class).
/tmp	Temporary files created by applications.	Temporary files and directories.
/usr	User-related programs and data.	System-wide applications and files (/usr/bin, /usr/lib).
/usr/bin	Binaries for user commands.	Executable files for user commands.
/usr/lib	Libraries for user programs.	Libraries used by binaries in /usr/bin.
/usr/sbin	System binaries for administrative tasks.	System management commands (e.g., useradd, shutdown).
/usr/local	Local software and scripts installed by the user.	Locally installed software and scripts.
/var	Variable data; files that change frequently.	Logs, spool files, mail, and temporary files (/var/log, /var/spool).
/var/log	System and application log files.	Log files for system and applications.
/var/spool	Spool directories for tasks and services.	Mail queues, print jobs.
/var/tmp	Temporary files that need to persist between reboots.	Persistent temporary files.
/var/lib	Variable data used by applications and services.	Database files, application state information.
/var/cache	Cached data from applications.	Cached files to speed up application performance.

/var/mail	User mailboxes.	Mail storage for individual users.
/var/run	Runtime information such as system and service status.	PID files, process runtime data.
/var/opt	Variable data for applications installed in /opt.	Application-specific data (similar to /opt).
/etc/passwd	User account information. Contains user details like username, UID, GID, and home directory.	User account entries (e.g., root:x:0:0:root:/root:/bin/bash).
/etc/shadow	Contains hashed passwords and account expiration information.	Password hashes and expiration info.
/etc/group	Defines user groups and their members.	Group definitions (e.g., root:x:0:).
/etc/fstab	Static filesystem information; lists filesystems and mount points.	Filesystem mount details (e.g., /dev/sda1 / ext4 defaults 0 1).
/etc/hosts	Static table of IP addresses to hostnames.	IP address and hostname mappings.
/etc/hostname	Contains the system's hostname.	The system's hostname.
/etc/resolv.conf	DNS resolver configuration file.	Nameserver IPs and domain search paths.
/etc/sysctl.conf	Configuration file for kernel parameters.	Kernel parameter settings (e.g., net.ipv4.ip_forward=1).
/etc/network/interfaces	Network interface configuration file (used in Debian-based systems).	Network interface settings (e.g., auto eth0 iface eth0 inet dhcp).
/var/log/httpd/	Directory for Apache HTTP server logs (may vary by distribution).	Apache access and error logs.
/var/log/nginx/	Directory for Nginx web server logs (may vary by distribution).	Nginx access and error logs.