

# # Essential Linux Commands

## 1. File Operations:

- `ls`: Lists all files and directories in the present working directory
- `ls -R`: Lists files in sub-directories as well
- `ls -a`: Shows hidden files
- `ls -al`: Lists files and directories with detailed information like permissions, size, owner, etc.
- `cd directoryname`: Changes the directory
- `cd ..`: Moves one level up
- `pwd`: Displays the present working directory
- `cat > filename`: Creates a new file
- `cat filename`: Displays the file content
- `cat file1 file2 > file3`: Joins two files (file1 and file2) and stores the output in a new file (file3)
- `touch filename`: Creates or modifies a file
- `rm filename`: Deletes a file
- `cp source destination`: Copies files from the source path to the destination path
- `mv source destination`: Moves files from the source path to the destination path
- `find / -name filename`: Finds a file or a directory by its name starting from root
- `file filename`: Determines the file type
- `less filename`: Views the file content page by page
- `head filename`: Views the first ten lines of a file
- `tail filename`: Views the last ten lines of a file
- `lsof`: Shows which files are opened by which process.
- `du -h --max-depth=1`: Shows the size of each directory. Use `--max-depth=1` to limit the output to the current directory and its immediate children.
- `fdisk`: Disk partition manipulation command.

## 2. Directory Operations:

- `mkdir directoryname`: Creates a new directory in the present working directory
  - `rmdir directoryname`: Deletes a directory
  - `cp -r source destination`: Copies directories recursively
  - `mv olddir newdir`: Renames directories
  - `find / -type d -name directoryname`: Finds a directory starting from root
- 

## 3. Process Operations:

- `ps`: Displays your currently active processes
  - `top`: Displays all running processes
  - `kill pid`: Kills the process with given pid
  - `pkill name`: Kills the process with the given name
  - `bg`: Resumes suspended jobs without bringing them to foreground
  - `fg`: Brings the most recent job to foreground
  - `fg n`: Brings job n to the foreground
  - `renice +n [pid]`: Change the priority of a running process.
  - `&>filename`: Redirects both the stdout and the stderr to the file filename.
  - `1>filename`: Redirect the stdout to file filename.
  - `2>filename`: Redirect stderr to file filename.
- 

## 4. File Permissions:

- `chmod octal filename`: Change the permissions of file to octal, which can be between 0 (no permissions) to 7 (full permissions)
- `chown ownername filename`: Change file owner
- `chgrp groupname filename`: Change group owner

## 5. Networking:

- `ping host`: Ping a host and outputs results
  - `whois domain`: Get whois information for domain
  - `dig domain`: Get DNS information for domain
  - `netstat -pnltu`: Display various network related information such as network connections, routing tables, interface statistics etc.
  - `ifconfig`: Displays IP addresses of all network interfaces
  - `ssh user@host`: Remote login into the host as user
  - `scp`: Transfers files between hosts over ssh
  - `wget url`: Download files from the web
  - `curl url`: Sends a request to a URL and returns the response
  - `traceroute domain`: Prints the route that a packet takes to reach the domain.
  - `mtr domain`: mtr combines the functionality of the traceroute and ping programs in a single network diagnostic tool.
  - `ss`: Another utility to investigate sockets. It's a more modern alternative to `netstat`.
  - `nmap`: Network exploration tool and security scanner.
- 

## 6. Archives and Compression:

- `tar cf file.tar files`: Create a tar named file.tar containing files
  - `tar xf file.tar`: Extract the files from file.tar
  - `gzip file`: Compresses file and renames it to file.gz
  - `gzip -d file.gz`: Decompresses file.gz back to file
  - `zip -r file.zip files`: Create a zip archive named file.zip
  - `unzip file.zip`: Extract the contents of a zip file
- 

## 7. Text Processing:

- `grep pattern files`: Search for pattern in files
- `grep -r pattern dir`: Search recursively for pattern in dir

- `command | grep pattern`: Pipe the output of command to grep for searching
  - `echo 'text'`: Prints text
  - `sed 's/string1/string2/g' filename`: Replaces string1 with string2 in filename
  - `diff file1 file2`: Compares two files and shows the differences
  - `wc filename`: Count lines, words, and characters in a file
  - `awk`: A versatile programming language for working on files.
  - `sed -i 's/string1/string2/g' filename`: Replace string1 with string2 in filename. The `-i` option edits the file in-place.
  - `cut -d':' -f1 /etc/passwd`: Cut out the first field of each line in `/etc/passwd`, using colon as a field delimiter.
- 

## 8. Disk Usage:

- `df`: Shows disk usage
  - `du`: Shows directory space usage
  - `free`: Show memory and swap usage
  - `whereis app`: Show possible locations of app
- 

## 9. System Info:

- `date`: Show the current date and time
- `cal`: Show this month's calendar
- `uptime`: Show current uptime
- `w`: Display who is online
- `whoami`: Who you are logged in as
- `uname -a`: Show kernel information
- `df -h`: Disk usage in human readable format
- `du -sh`: Disk usage of current directory in human readable format
- `free -m`: Show free and used memory in MB

## 10. Package Installations:

- `sudo apt-get update`: Updates package lists for upgrades
  - `sudo apt-get upgrade`: Upgrades all upgradable packages
  - `sudo apt-get install pkgname`: Install pkgname
  - `sudo apt-get remove pkgname`: Removes pkgname
- 

## 11. Others (mostly used in scripts):

- `command1 ; command2`: Run command1 and then command2
  - `command1 && command2`: Run command2 if command1 is successful
  - `command1 || command2`: Run command2 if command1 is not successful
  - `command &`: Run command in background
- 

## 12. Version Control (Git commands):

- `git init`: Initialize a local git repository
- `git clone url`: Create a local copy of a remote repository
- `git add filename`: Add a file to the staging area
- `git commit -m "Commit message"`: Commit changes with a message
- `git status`: Check the status of the working directory
- `git pull`: Pull latest changes from the remote repository
- `git push`: Push changes to the remote repository
- `git branch`: List all local branches
- `git branch branchname`: Create a new branch
- `git checkout branchname`: Switch to a branch
- `git merge branchname`: Merge a branch into the active branch
- `git stash`: Stash changes in a dirty working directory
- `git stash apply`: Apply changes from a stash
- `git log`: View commit history
- `git reset`: Reset your HEAD pointer to a previous commit
- `git rm filename`: Remove a file from version control
- `git rebase`: Reapply commits on top of another base tip.

- `git revert`: Create a new commit that undoes all of the changes made in a particular commit, then apply it to the current branch.
  - `git cherry-pick commitID`: Apply the changes introduced by some existing commits.
- 

### 13. Environment Variables:

- `env`: Display all environment variables
  - `echo $VARIABLE`: Display the value of an environment variable
  - `export VARIABLE=value`: Set the value of an environment variable
  - `alias new_command='old_command options'`: Create a new command that executes the old command with the specified options.
  - `echo $PATH`: Print the PATH environment variable.
  - `export PATH=$PATH:/new/path`: Add /new/path to the PATH.
- 

### 14. Job Scheduling (Cron Jobs):

- `crontab -l`: List all your cron jobs
  - `crontab -e`: Edit your cron jobs
  - `crontab -r`: Remove all your cron jobs
  - `crontab -v`: Display the last time you edited your cron jobs
  - `crontab file`: Install a cron job from a file
  - `@reboot command`: Schedule a job to run at startup
- 

### 15. Package Installations (using pip, a Python package installer):

- `pip install packagename`: Install a Python package.
- `pip uninstall packagename`: Uninstall a Python package.
- `pip freeze > requirements.txt`: Freeze the installed packages into a requirements file.
- `pip install -r requirements.txt`: Install packages from a requirements file.

## 16. Shell Scripting:

- `#!/bin/bash`: Shebang line to specify the script interpreter.
  - `$0, $1, ..., $9, ${10}, ${11}`: Script arguments.
  - `if [condition]; then ... fi`: if statement in bash scripts.
  - `for i in {1..10}; do ... done`: for loop in bash scripts.
  - `while [condition]; do ... done`: while loop in bash scripts.
  - `function name() {...}`: Define a function.
- 

## 17. System Monitoring and Performance:

- `iostat`: Reports Central Processing Unit (CPU) statistics and input/output statistics for devices, partitions, and network filesystems.
  - `vmstat`: Reports information about processes, memory, paging, block IO, traps, disks, and CPU activity.
  - `htop`: An interactive process viewer for Unix systems. It's a more user-friendly alternative to `top`.
- 

## 10. Search and Find:

- `locate filename`: Find a file by its name. The database updated by `updatedb` command.
  - `whereis programname`: Locate the binary, source, and manual page files for a command.
  - `which commandname`: Shows the full path of (shell) commands.
- 

## 11. Compression / Archives:

- `tar -cvf archive.tar dirname/`: Create a tar archive.
- `tar -xvf archive.tar`: Extract a tar archive.
- `tar -jcvf archive.tar.bz2 dirname/`: Create a compressed bz2 archive.
- `tar -jxvf archive.tar.bz2`: Extract a bz2 archive.

## 12. Disk Usage:

- `dd if=/dev/zero of=/tmp/output.img bs=8k count=256k`: Create a file of a certain size for testing disk speed.
  - `hdparm -Tt /dev/sda`: Measure the read speed of your hard drive.
- 

## 13. Others:

- `yes > /dev/null &`: Use this command to push a system to its limit.
  - `:(){ :|:& };;:` A fork bomb – handle with care. Do not run this command on a production system.
- 

Remember, you can always use the `man` command (e.g. `man ls`) to get more information about each command.