

Linux commands

1. Navigation and File Management

1. ****`pwd`****
 - Shows the full path of the current directory.
 - Useful with scripts to confirm the current directory.
2. ****`ls`****
 - ``ls -l``: Long listing format (shows permissions, owner, size, etc.).
 - ``ls -a``: Includes hidden files (files starting with ``.``).
 - ``ls -lh``: Long format with human-readable sizes.
 - ``ls -R``: Recursively lists all subdirectories.
3. ****`cd`****
 - ``cd ..``: Moves up one directory level.
 - ``cd ~``: Goes to the home directory.
 - ``cd -``: Switches to the previous directory.
4. ****`mkdir`****
 - ``mkdir -p``: Creates parent directories if they don't exist.

```
```bash
mkdir -p parent/child/grandchild
```
```
5. ****`rm`****
 - ``rm -i``: Prompts for confirmation before deletion.
 - ``rm -r``: Deletes directories and their contents recursively.
 - ``rm -f``: Forces deletion without confirmation.
6. ****`cp`****
 - ``cp -r``: Copies directories recursively.
 - ``cp -i``: Prompts before overwriting files.
 - ``cp -u``: Copies only if the source file is newer.
7. ****`mv`****
 - ``mv -i``: Prompts before overwriting.
 - Renaming files:

```
```bash
mv old_name new_name
```
```

2. Viewing and Editing Files

8. ****`cat`****
 - ``cat file.txt``: Displays the file content.
 - ``cat file1 file2 > combined.txt``: Concatenates files.
 - ``cat -n file.txt``: Adds line numbers.
9. ****`nano`****
 - Simple text editor:
 - ``Ctrl + O``: Save file.
 - ``Ctrl + X``: Exit.
10. ****`less`****
 - Navigate with:
 - ``Space``: Next page.
 - ``b``: Previous page.
 - ``/search_term``: Search within the file.
11. ****`touch`****
 - ``touch file.txt``: Creates an empty file.
 - ``touch -a file.txt``: Updates access time only.
 - ``touch -m file.txt``: Updates modification time.

3. Permissions and Ownership

12. **chmod**
- `chmod 755 file.sh`: Gives owner full permission, others read and execute.
 - `chmod +x file.sh`: Adds execute permission.
 - `chmod -w file.txt`: Removes write permission.
13. **chown**
- `chown user file.txt`: Changes owner.
 - `chown user:group file.txt`: Changes owner and group.
 - `chown -R user:group dir/`: Applies recursively.

4. System Monitoring

14. **top**
- `top`: Displays processes sorted by CPU usage.
 - Interactive commands:
 - `h`: Help.
 - `k`: Kill a process by ID.
 - `q`: Quit.
15. **df**
- `df -h`: Shows disk usage in human-readable format.
 - `df -T`: Displays file system type.
16. **free**
- `free -h`: Human-readable memory usage.
 - `free -t`: Displays total memory (RAM + swap).

5. Networking

17. **ping**
- `ping -c 5 google.com`: Sends 5 packets to test connectivity.
 - `ping -i 2 google.com`: Sets interval between packets (2 seconds).
18. **ifconfig** (deprecated in favor of `ip`)
- `ifconfig`: Shows network interface details.
 - `ifconfig eth0 down`: Disables the `eth0` interface.
- ip** Command (Modern Alternative)
- `ip addr`: Shows IP addresses.
 - `ip link set eth0 up`: Enables `eth0`.
 - `ip route`: Displays routing table.

Other Handy Commands

19. **find**: Locates files in directories.
- ```
bash
find /path -name "filename"
find . -type f -size +1M
```
20. **grep**: Searches for patterns in files.
- ```
bash
grep "pattern" file.txt
grep -r "search_term" /path/to/directory
```
21. **tar**: Archives and extracts files.
- ```
bash
tar -cvf archive.tar file_or_dir
tar -xvf archive.tar
```
22. **wget**: Downloads files from the web.
- ```
bash
wget http://example.com/file.zip
```
23. **sudo**: Runs commands as root.
- ```
bash
```

```
sudo apt update
\\
```

+++++

## Poweroff

24. **poweroff**: This immediately shuts down the system and powers off the computer. It's equivalent to pressing the power button on the computer.

**poweroff -f** : Forces an immediate shutdown, bypassing any ongoing processes or warnings.

## Reboot

25. **reboot**: This command restarts the system immediately.