1. `ls` -Linux List Directory Contents

Command: `ls [options] [directory]`
Description:
Lists files and directories in the specified directory. If no directory is provided, it lists the contents of the current directory.
Options:

- `-l` : List in long format (detailed information)
- `-a` : Show hidden files
- `-h` : Display file sizes in a human-readable format (e.g., KB, MB)

---

2. `grep` - Search Text Using Patterns

Command: `grep [options] pattern [file]`
Description:
Searches for a specified pattern in the given file or standard input.
Options:

- `-i` : Case-insensitive search
- `-r` : Recursively search directories
- `-v` : Invert match (return lines that don’t match the pattern)

---

3. `rm` - Remove Files or Directories

Command: `rm [options] [file]`
Description:
Deletes files and directories. Be careful, as files deleted with `rm` are not easily recoverable.
Options:

- `-r` : Recursively delete directories and their contents
- `-f` : Force deletion without prompting for confirmation

---

4. `cp` - Copy Files and Directories

Command: `cp [options] source destination`
Description:
Copies files or directories from the source to the destination.
Options:

- `-r` : Copy directories recursively
- `-i` : Prompt before overwriting files

---

5. `mkdir` - Make Directories

Command: `mkdir [options] directory_name`
Description:
Creates a new directory.
Options:

- `-p` : Create parent directories if they do not exist

---

6. `touch` - Create or Update Files

Command: `touch [file]`
Description:
Creates an empty file if it doesn't already exist, or updates the timestamp of the file if it does exist.

---

7. `vi` - Text Editor

Command: `vi [file]`
Description:
Opens the `vi` text editor to create or edit text files. `vi` is a powerful, modal editor that works in two modes: insert mode and command mode.

---

8. nano - Simple Text Editor

Command: `nano [file]`
Description:

Opens the `nano` text editor, a simpler and more user-friendly alternative to `vi`. It's commonly used for basic text editing.

---

9. `systemctl` - Control the System Services

Command: `systemctl [options] [command]`
Description:
Used to control system services and manage systemd.
Options:

- `start` : Start a service (e.g., `systemctl start apache2`)
- `stop` : Stop a service
- `status` : Show the status of a service
- `restart` : Restart a service

---

10. `vim` - Improved vi Text Editor

Command: `vim [file]`
Description:
Vim is an improved version of `vi`. It includes syntax highlighting, multi-level undo, and other features to enhance the text editing experience.

---

11. `mv` - Move or Rename Files

Command: `mv [source] [destination]`
Description:
Moves a file or directory to a new location or renames a file. If the destination is a directory, the file is moved there.
Options:

- `-i` : Prompt before overwriting an existing file

---

12. `sudo` - Execute Command as Superuser

Command: `sudo [command]`
Description:
Allows a permitted user to execute a command as the superuser (or another user), as specified in the sudoers file. Used for administrative tasks.

---

13. `man` - Display Manual for Command

Command: `man [command]`
Description:
Displays the manual page for the specified command, providing detailed information and options available for that command.

---

14. `df` - Disk Space Usage

Command: `df [options]`
Description:
Shows disk space usage for mounted filesystems.
Options:

- `-h` : Human-readable format (e.g., MB, GB)
- `-T` : Display the filesystem type

---

15. `pwd` - Print Working Directory

Command: `pwd`
Description:
Prints the absolute path of the current working directory.

---

16. `cat` - Concatenate and Display Files

Command: `cat [file]`
Description:
Concatenates and displays the contents of the specified file.

---

17. `history` - Command History

Command: `history`
Description:
Displays the history of commands you’ve executed in the current session.

---

18. `ssh` - Secure Shell

Command: `ssh [user]@[hostname]`
Description:
Connects to a remote machine securely over a network using the SSH protocol. Allows for secure communication and execution of commands on the remote machine.

Linux Monitoring Commands

1. **`top` (Task Manager):**
    - Shows real-time system processes and resource usage.

Example:
code

```
top
```

    ○

2. **`htop` (Interactive Process Viewer):**
    - Enhanced version of `top` with a more user-friendly interface.

Example:
code

```
htop
```

    ○

3. **`vmstat` (Virtual Memory Statistics):**
    - Displays information about processes, memory, swap, I/O, and CPU performance.

Example:
code

```
vmstat 2 5  # Update every 2 seconds for 5 iterations
```

    ○

4. **`iostat` (Input/Output Statistics):**
    - Reports CPU and I/O statistics for devices and partitions.

Example:
code

```
iostat -x 2  # Detailed statistics every 2 seconds
```

○

5. **`sar` (System Activity Report):**
   ○ Collects and displays system activity information, such as CPU, memory, and I/O usage.

Example:
code

```
sar -u 5  # Show CPU usage every 5 seconds
```

○

6. **`df` (Disk Free Space):**
   ○ Displays file system disk space usage.

Example:
code

```
df -h
```

○

7. **`free` (Memory Usage):**
   ○ Displays memory usage information (RAM and swap).

Example:
code

```
free -h  # Human-readable format
```

○

8. **`dstat` (System Resource Statistics):**
   ○ Combines `vmstat`, `iostat`, and `netstat` in one view.

Example:
code

```
dstat
```

○

9. **`uptime` (System Uptime):**
   ○ Shows how long the system has been running, along with load averages.

Example:
code

```
uptime
```

- ○

10. **`ps` (Process Status):**
    - Shows information about running processes.

Example:
code

```
ps aux   # Display all processes with detailed info
```

- ○

11. **`netstat` (Network Statistics):**
    - Shows network connections, routing tables, and interface statistics.

Example:
code

```
netstat -tuln  # List listening ports
```

- ○

12. **`nmap` (Network Mapper):**
    - Network scanning tool to discover devices, ports, and services.

Example:
code

```
nmap -sP 192.168.1.0/24  # Ping scan on a local subnet
```

- ○

13. **`ss` (Socket Statistics):**
    - Displays socket statistics, similar to `netstat`, but faster.

Example:
code

```
ss -tuln  # List open TCP and UDP ports
```

`14. lsof` (List Open Files):

- Lists all open files and the processes that opened them.
- Example:
  code

  ```
  lsof -i :80  # Show all processes using port 80
  ```