

## “Mastering the Command Line”

### Top 50 Linux Commands Every User Should Know “ Explained with Real-World Examples”

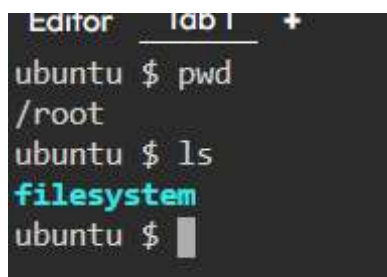
1. **ls**—The most frequently used command in Linux to list directories
2. **pwd**—Print working directory command in Linux
3. **cd**—Linux command to navigate through directories
4. **mkdir**—Command used to create directories in Linux
5. **mv**—Move or rename files in Linux
6. **cp**—Similar usage as mv but for copying files in Linux
7. **rm**—Delete files or directories
8. **touch**—Create blank/empty files
9. **ln**—Create symbolic links (shortcuts) to other files
10. **cat**—Display file contents on the terminal
11. **clear**—Clear the terminal display
12. **echo**—Print any text that follows the command
13. **less**—Linux command to display paged outputs in the terminal
14. **man**—Access manual pages for all Linux commands
15. **uname**—Linux command to get basic information about the OS
16. **whoami**—Get the active username
17. **tar**—Command to extract and compress files in Linux
18. **grep**—Search for a string within an output
19. **head**—Return the specified number of lines from the top
20. **tail**—Return the specified number of lines from the bottom

21. **diff**—Find the difference between two files
22. **cmp**—Allows you to check if two files are identical
23. **comm**—Combines the functionality of diff and cmp
24. **sort**—Linux command to sort the content of a file while outputting
25. **export**—Export environment variables in Linux
26. **zip**—Zip files in Linux
27. **unzip**—Unzip files in Linux
28. **ssh**—Secure Shell command in Linux
29. **service**—Linux command to start and stop services
30. **ps**—Display active processes
31. **kill** and **killall**—Kill active processes by process ID or name
32. **df**—Display disk filesystem information
33. **mount**—Mount file systems in Linux
34. **chmod**—Command to change file permissions
35. **chown**—Command for granting ownership of files or folders
36. **ifconfig**—Display network interfaces and IP addresses
37. **traceroute**—Trace all the network hops to reach the destination
38. **wget**—Direct download files from the internet
39. **ufw**—Firewall command
40. **iptables**—Base firewall for all other firewall utilities to interface with
41. **apt**, **pacman**, **yum**, **rpm**—Package managers depending on the distro

- 42. **sudo**—Command to escalate privileges in Linux
- 43. **cal**—View a command-line calendar
- 44. **alias**—Create custom shortcuts for your regularly used commands
- 45. **dd**—Majorly used for creating bootable USB sticks
- 46. **whereis**—Locate the binary, source, and manual pages for a command
- 47. **whatis**—Find what a command is used for
- 48. **top**—View active processes live with their system usage
- 49. **useradd** and **usermod**—Add new user or change existing users data
- 50. **passwd**—Create or update passwords for existing users

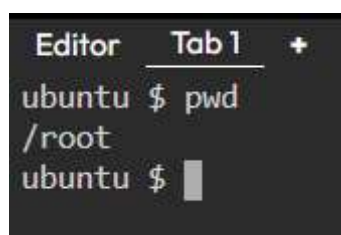
## Learn more about each command's real-time use case :

- 1. **ls**—The most frequently used command in Linux to list directories.



```
Editor  Tab 1  +
ubuntu $ pwd
/root
ubuntu $ ls
filesystem
ubuntu $
```

- 2. **pwd**—Print working directory command in Linux



```
Editor  Tab 1  +
ubuntu $ pwd
/root
ubuntu $
```

3. **cd**—Linux command to navigate through directories

```
Editor  Tab 1  +
ubuntu $ dir
filesystem
ubuntu $ cd filesystem
ubuntu $ pwd
/root/filesystem
ubuntu $
```

4. **mkdir**—Command used to create directories in Linux

```
Editor  Tab 1  +
ubuntu $ mkdir mani
ubuntu $ ls
filesystem mani
ubuntu $
```

5. **mv**—Move or rename files in Linux

```
ubuntu $ mkdir mani
ubuntu $ ls
filesystem mani
ubuntu $ mv mani mani_2023
ubuntu $ ls
filesystem mani_2023
ubuntu $
```

6. **cp**—Similar usage as mv but for copying files in Linux

```
Editor  Tab 1  +
ubuntu $ ls
filesystem mani
ubuntu $ cp -r mani mani-copy
ubuntu $ ls
filesystem mani mani-copy
ubuntu $
```

7. **rm**—Delete files or directories

```
ubuntu $ ls
filesystem mani_2023
ubuntu $ rmdir mani_2023
ubuntu $ ls
filesystem
ubuntu $
```

8. **touch**—Create blank/empty files

```
ubuntu $ ls
filesystem
ubuntu $ touch mani.txt
ubuntu $ ls
filesystem mani.txt
ubuntu $
```

9. **ln**—Create symbolic links (shortcuts) to other files

```
ubuntu $ ls
filesystem mani.txt
ubuntu $ ln -s mani.txt mani.txt-link
ubuntu $ ls
filesystem mani.txt mani.txt-link
ubuntu $ ls -l
total 0
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 0 Mar 23 08:27 mani.txt
lrwxrwxrwx 1 root root 8 Mar 23 08:31 mani.txt-link -> mani.txt
ubuntu $
```

10. **cat**—Display file contents on the terminal

```
ubuntu $ ls
filesystem mani.txt mani.txt-link
ubuntu $ cat mani.txt
hello , this txt edited with nano.
ubuntu $
```

11. **clear**—Clear the terminal display

```
ubuntu $ ls -l
total 4
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 35 Mar 23 08:36 mani.txt
lrwxrwxrwx 1 root root 8 Mar 23 08:31 mani.txt-link -> mani.txt
ubuntu $ clear
```

A screenshot of a terminal window. At the top, there is a tab labeled 'Editor' and 'Tab 1' with a '+' icon to its right. Below the tab, the terminal prompt 'ubuntu \$' is visible, followed by a cursor.

12. **echo**—Print any text that follows the command.

```
ubuntu $ ls -l
total 4
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 35 Mar 23 08:36 mani.txt
lrwxrwxrwx 1 root root 8 Mar 23 08:31 mani.txt-link -> mani.txt
ubuntu $ echo mani.txt
mani.txt
ubuntu $ echo filesysteme
filesysteme
```

13. **less**—Linux command to display paged outputs in the terminal.

```
ubuntu $ ls -l
total 4
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem ->
-rw-r--r-- 1 root root 474 Mar 23 12:50 mani.txt
ubuntu $ less mani.txt
```

```
<!DOCTYPE html>
<html>
<head>
  <title>My Simple Website</title>
</head>
<body>
  <header>
    <h1>Welcome to My Website</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
  <main>
    <h2>About Us</h2>
    <p>We are a small team of developers who create amazing websites.</p>
  </main>
  <footer>
    <p>&copy; 2023 My Simple Website. All rights reserved.</p>
  </footer>
</body>
</html>

mani.txt (END)
```

14. **man**—Access manual pages for all Linux commands ;

The `man` command in Linux is used to display the manual pages of Linux commands and other system functionalities.

```
Editor  Tab 1  +
ubuntu $ man mkdir
```

```
Editor  Tab 1  +
MKDIR(1)                                         User Commands

NAME
  mkdir - make directories

SYNOPSIS
  mkdir [OPTION]... DIRECTORY...

DESCRIPTION
  Create the DIRECTORY(ies), if they do not already exist.

  Mandatory arguments to long options are mandatory for short options too.

  -m, --mode=MODE
      set file mode (as in chmod), not a=rwx - umask

  -p, --parents
      no error if existing, make parent directories as needed

  -v, --verbose
      print a message for each created directory

  -Z
      set SELinux security context of each created directory to the default type

  --context[=CTX]
      like -Z, or if CTX is specified then set the SELinux or SMACK security context to CTX

  --help
      display this help and exit

  --version
      output version information and exit

AUTHOR
  Written by David MacKenzie.

REPORTING BUGS
  GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
  Manual page mkdir(1) line 1 (press h for help or q to quit)
```

15. **uname**—Linux command to get basic information about the OS

```
ubuntu $ uname
Linux
ubuntu $
```

16. **whoami**—Get the active username

```
ubuntu $ whoami
root
ubuntu $
```

17. **tar**—Command to extract and compress files in Linux

```
ubuntu $ ls -l
total 4
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 474 Mar 23 12:50 mani.txt
ubuntu $ tar -cvf archive.tar mani.txt
mani.txt
ubuntu $ tar -xvf archive.tar
mani.txt
ubuntu $ tar -tvf archive.tar
-rw-r--r-- root/root      474 2023-03-23 12:50 mani.txt
ubuntu $
```

18. **grep**—Search for a string within an output

```
ubuntu $ ls -l
total 16
-rw-r--r-- 1 root root 10240 Mar 23 13:05 archive.tar
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 474 Mar 23 12:50 mani.txt
ubuntu $ cat mani.txt | grep "2023"
<p>&copy; 2023 My Simple Website. All rights reserved.</p>
ubuntu $
```

19. **head**—Return the specified number of lines from the top



```
ubuntu $ ls -l
total 16
-rw-r--r-- 1 root root 10240 Mar 23 13:05 archive.tar
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 474 Mar 23 12:50 mani.txt
ubuntu $ head mani.txt
<!DOCTYPE html>
<html>
<head>
    <title>My Simple Website</title>
</head>
<body>
    <header>
        <h1>Welcome to My Website</h1>
    </header>
    <nav>
```

20. **tail**—Return the specified number of lines from the bottom

```
ubuntu $ ls -l
total 16
-rw-r--r-- 1 root root 10240 Mar 23 13:05 archive.tar
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 474 Mar 23 12:50 mani.txt
ubuntu $ tail mani.txt
    <main>
        <h2>About Us</h2>
        <p>We are a small team of developers who create amazing websites.</p>
    </main>
    <footer>
        <p>&copy; 2023 My Simple Website. All rights reserved.</p>
    </footer>
</body>
</html>
ubuntu $
```

21. **diff**—Find the difference between two files

```
ubuntu $ ls -l
total 24
-rw-r--r-- 1 root root 513 Mar 23 13:27 a
-rw-r--r-- 1 root root 10240 Mar 23 13:05 archive.tar
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 474 Mar 23 12:50 mani1.txt
-rw-r--r-- 1 root root 507 Mar 23 13:29 mani2.txt
ubuntu $ diff mani1.txt mani2.txt
26a27,28
> This code edited with nano tool
>
ubuntu $
```

22. **cmp**—Allows you to check if two files are identical.

```
ubuntu $ ls -l
total 24
-rw-r--r-- 1 root root 513 Mar 23 13:27 a
-rw-r--r-- 1 root root 10240 Mar 23 13:05 archive.tar
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 474 Mar 23 12:50 mani1.txt
-rw-r--r-- 1 root root 507 Mar 23 13:29 mani2.txt
ubuntu $ cmp mani1.txt mani2.txt
cmp: EOF on mani1.txt after byte 474, line 26
ubuntu $
```

23. **comm**—Combines the functionality of diff and cmp

```
ubuntu $ comm mani1.txt mani2.txt
<!DOCTYPE html>
<html>
<head>
  <title>My Simple Website</title>
</head>
<body>
  <header>
    <h1>Welcome to My Website</h1>
  </header>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
  <main>
    <h2>About Us</h2>
    <p>We are a small team of developers who create amazing websites.</p>
  </main>
  <footer>
    <p>&copy; 2023 My Simple Website. All rights reserved.</p>
  </footer>
</body>
</html>

This code edited with nano tool
comm: file 2 is not in sorted order
```

24. **sort**—Linux command to sort the content of a file while outputting

```
ubuntu $ ls -l
total 4
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 39 Mar 23 13:51 mani.txt
ubuntu $ sort mani.txt

hello
this txt file edited with nano
ubuntu $
```

25. **export**—Export environment variables in Linux

```
ubuntu $ export PASSWORD=mysupersecretpassword
ubuntu $ #!/bin/bash
ubuntu $ echo "The password is: $PASSWORD"
The password is: mysupersecretpassword
ubuntu $
```

## 26. zip—Zip files in Linux

```
ubuntu $ ls -l
total 4
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 39 Mar 23 13:51 mani.txt
ubuntu $ zip mani.zip mani.txt
  adding: mani.txt (stored 0%)
ubuntu $ ls -l
total 8
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 39 Mar 23 13:51 mani.txt
-rw-r--r-- 1 root root 205 Mar 23 14:17 mani.zip
```

## 27. unzip—Unzip files in Linux

```
ubuntu $ ls -l
total 8
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 39 Mar 23 13:51 mani.txt
-rw-r--r-- 1 root root 205 Mar 23 14:17 mani.zip
ubuntu $ unzip mani.zip
Archive:  mani.zip
replace mani.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: n
ubuntu $ ls -l
total 8
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 39 Mar 23 13:51 mani.txt
-rw-r--r-- 1 root root 205 Mar 23 14:17 mani.zip
ubuntu $
```

## 28. ssh—Secure Shell command in Linux

ssh username@hostname

## 29. service—Linux command to start and stop services

root@ubuntu:~ ->> service ssh status

root@ubuntu:~ ->> service ssh stop

root@ubuntu:~ ->> service ssh start

```
ubuntu $ service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-02-23 12:36:58 UTC; 4 weeks 0 days ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 597 (sshd)
    Tasks: 75 (limit: 2339)
   Memory: 1.3G
   CGroup: /system.slice/ssh.service
           └─ 597 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
             └─ 21578 /opt/theia/node /opt/theia/browser-app/src-gen/backend/main.js /root --hostname=0.0.0.0 --port 40205
               └─ 21589 bash -c while true; do /bin/kc-terminal -p 40200 -t disableLeaveAlert=true bash; done
                 └─ 21591 /bin/kc-terminal -p 40200 -t disableLeaveAlert=true bash
                   └─ 21612 sshd: kc-internal@notty
                     └─ 21625 dhclient -v
                       └─ 21677 dhclient -v
                         └─ 21710 dhclient -v
```

### 30. ps—Display active processes

```
ubuntu $ ps
  PID TTY          TIME CMD
 22514 pts/0        00:00:00 bash
 23118 pts/0        00:00:00 systemctl
 23122 pts/0        00:00:00 pager
 23296 pts/0        00:00:00 sudo
 23297 pts/0        00:00:00 su
 23320 pts/0        00:00:00 bash
 23848 pts/0        00:00:00 ps
ubuntu $
```

### 31. kill and killall—Kill active processes by process ID or name

```
Editor  Tab 1  +
ubuntu $ ps
  PID TTY          TIME CMD
 22514 pts/0        00:00:00 bash
 23118 pts/0        00:00:00 systemctl
 23122 pts/0        00:00:00 pager
 23296 pts/0        00:00:00 sudo
 23297 pts/0        00:00:00 su
 23320 pts/0        00:00:00 bash
 24066 pts/0        00:00:00 bash
 24658 pts/0        00:00:00 ps
ubuntu $ kill 23297
ubuntu $
Session terminated, killing shell... ...killed.
Terminated
ubuntu $
```

### 32. df—Display disk filesystem information



```
ubuntu $ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            998196         0    998196   0% /dev
tmpfs           203096     1008    202088   1% /run
/dev/vda1       20134592 4595532  15522676  23% /
tmpfs           1015468         0    1015468   0% /dev/shm
tmpfs            5120         0         5120   0% /run/lock
tmpfs           1015468         0    1015468   0% /sys/fs/cgroup
/dev/loop0       64768     64768         0 100% /snap/core20/1634
/dev/loop1       69504     69504         0 100% /snap/lxd/22753
/dev/loop2       49152     49152         0 100% /snap/snapd/17336
/dev/vda15       106858     5313    101545   5% /boot/efi
ubuntu $
```

33. **mount**—Mount file systems in Linux.

```
root@ubuntu:~ -->> mount /dev/cdrom /mnt
root@ubuntu:~ -->> df -h
```

34. **chmod**—Command to change file permissions

```
ubuntu $ ls -l
total 16
drwxr-xr-x 2 root root 4096 Mar 23 16:24 755
drwxr-xr-x 2 root root 4096 Mar 23 16:22 do-w
drwxr-xr-x 2 root root 4096 Mar 23 16:24 example.sh
lrwxrwxrwx 1 root root    1 Feb 23 12:40 filesystem -> /
drwxr-xr-x 2 root root 4096 Mar 23 16:22 importany.txt
ubuntu $
```

35. **chown**—Command for granting ownership of files or folders.

```
ubuntu $ touch important.txt
ubuntu $ chmod go-w important.txt
ubuntu $ ls -l
total 0
lrwxrwxrwx 1 root root 1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root 0 Mar 23 16:29 important.txt
ubuntu $
```

36. **ifconfig**—Display network interfaces and IP addresses

```
Editor  Tab 1  +
ubuntu $ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:fb:12:3b:47 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1460
    inet 172.30.1.2 netmask 255.255.255.0 broadcast 172.30.1.255
    ether f2:05:f6:3f:86:80 txqueuelen 1000 (Ethernet)
    RX packets 354854 bytes 495919765 (495.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36074 bytes 18302746 (18.3 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 330 bytes 41538 (41.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 330 bytes 41538 (41.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu $
```

37. **traceroute**—Trace all the network hops to reach the destination

```
ubuntu $ traceroute localhost
traceroute to localhost (127.0.0.1), 64 hops max
 1  127.0.0.1  0.002ms  0.002ms  0.001ms
ubuntu $ traceroute google.com
traceroute to google.com (74.125.24.101), 64 hops max
 1  172.30.1.1  0.073ms  0.033ms  0.039ms
 2  51.79.228.231  0.077ms  0.042ms  0.041ms
 3  51.79.228.252  0.488ms  0.273ms  0.301ms
 4  10.161.49.124  0.212ms  0.166ms  0.144ms
 5  10.133.3.56  0.502ms  0.372ms  0.366ms
 6  10.75.0.54  0.177ms  0.205ms  0.142ms
 7  10.75.248.14  0.885ms  0.638ms  0.757ms
 8  103.5.15.16  4.384ms  1.964ms  1.968ms
 9  10.200.0.198  1.480ms  1.330ms  1.347ms
10  * * *
11  * * *
12  142.251.241.0  0.795ms  0.680ms  0.727ms
13  108.170.240.241  2.985ms  0.986ms  0.895ms
14  216.239.35.154  1.570ms  *  1.501ms
15  142.251.229.230  26.301ms  8.491ms  1.493ms
16  66.249.94.149  2.482ms  2.588ms  3.031ms
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  74.125.24.101  1.264ms  1.213ms  1.261ms
ubuntu $
```

38. wget—Direct download files from the internet.

```
ubuntu $ wget https://www.putty.org/
--2023-03-23 16:42:37-- https://www.putty.org/
Resolving www.putty.org (www.putty.org)... 52.84.251.60, 52.84.251.59, 52.84.251.5, ...
Connecting to www.putty.org (www.putty.org)[52.84.251.60]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4384 (4.3K) [text/html]
Saving to: 'index.html'

index.html          100%[=====] 4.28K --.-KB/s in 0.001s

2023-03-23 16:42:37 (7.28 MB/s) - 'index.html' saved [4384/4384]

ubuntu $
```

39. ufw—Firewall command

```
ubuntu $ sudo ufw enable
Firewall is active and enabled on system startup
ubuntu $ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
ubuntu $ sudo ufw allow http
Skipping adding existing rule
Skipping adding existing rule (v6)
ubuntu $
```

40. **iptables**—Base firewall for all other firewall utilities to interface with

```
ubuntu $ sudo iptables -A INPUT -p tcp --dport 22 -j DROP
ubuntu $ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
ubuntu $ sudo iptables -P OUTPUT ACCEPT
ubuntu $ ufw allow 80
Rule added
Rule added (v6)
ubuntu $
```

41. **apt, pacman, yum, rpm**—Package managers depending on the distro.

- **Debian and Debian-based distros**—`apt install <package name>`
- **Arch and Arch-based distros**—`pacman -S <package name>`
- **Red Hat and Red Hat-based distros**—`yum install <package name>`
- **Fedora and CentOS**—`yum install <package>`

42. **sudo**—Command to escalate privileges in Linux.

```
ubuntu $ sudo su
ubuntu $ pwd
/root
ubuntu $
```

43. **cal**—View a command-line calendar.



```
ubuntu $ cal Januarry 2023
      January 2023
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

ubuntu $ cal March 2023
      March 2023
Su Mo Tu We Th Fr Sa
           1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

ubuntu $ █
```

44. **alias**—Create custom shortcuts for your regularly used commands.

```
ubuntu $ alias lsl="ls -l"
ubuntu $ lsl
total 8
lrwxrwxrwx 1 root root    1 Feb 23 12:40 filesystem -> /
-rw-r--r-- 1 root root    0 Mar 23 16:29 important.txt
-rw-r--r-- 1 root root 4384 Jun  6 2022 index.html
ubuntu $ █
```

45. **dd**—Majorly used for creating bootable USB sticks

46. **whereis**—Locate the binary, source, and manual pages for a command.

```
ubuntu $ whereis sudo
sudo: /usr/bin/sudo /usr/lib/sudo /usr/share/man/man8/sudo.8.gz
ubuntu $ █
```

47. **whatis**—Find what a command is used for

```
ubuntu $ whatis sudo
sudo (8)                - execute a command as another user
ubuntu $
```

#### 48. top—View active processes live with their system usage

```
top - 16:56:44 up 54 min, 0 users, load average: 0.08, 0.08, 0.03
Tasks: 123 total, 1 running, 122 sleeping, 0 stopped, 0 zombie
%Cpu(s):  0.0 us,  1.3 sy,  4.7 ni, 94.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 1983.3 total, 176.3 free, 356.2 used, 1450.8 buff/cache
MiB Swap: 1024.0 total, 1023.2 free,  0.8 used, 1435.6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
21578	root	39	19	858652	80168	34224	S	0.7	3.9	0:05.49	node
22499	root	20	0	0	0	0	I	0.3	0.0	0:00.11	kworker/u2:0-events_power_efficie
22596	root	39	19	842988	62776	30124	S	0.3	3.1	0:06.15	node
22669	root	39	19	559336	38880	29732	S	0.3	1.9	0:00.19	node
1	root	20	0	169472	12456	7864	S	0.0	0.6	0:13.83	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
9	root	20	0	0	0	0	S	0.0	0.0	0:00.53	ksoftirqd/0
10	root	20	0	0	0	0	I	0.0	0.0	0:00.55	rcu_sched
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.01	migration/0
12	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
16	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_kthre
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
19	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
20	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
21	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
22	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0

#### 49. useradd and usermod—Add new user or change existing users data.

```
ubuntu $ useradd priyanka
ubuntu $ usermod -s /bin/zsh john
usermod: no changes
ubuntu $ usermod -s /bin/zsh priya
ubuntu $
```

#### 50. passwd—Create or update passwords for existing users.

```
ubuntu $ passwd
New password:
Retype new password:
passwd: password updated successfully
ubuntu $
```

Credits : [digitalocean](#)

That's it, thank you for reading.

<https://github.com/cloudncloud> [www.linkedin.com/in/manitechy](http://www.linkedin.com/in/manitechy)

👉 In case you would like to continue the discussion, you can always reach out to me on Twitter or on LinkedIn for professional networking, if you feel like following me on GitHub you can also do that.

👉 Follow Cloudncloud Tech Community for more insightful knowledge & resources & CloudnLoud YouTube channel.