

Command Line BASICS

✓ **NOTE:**

WINDOWS CMD->NOT CASE SENSITIVE

LINUS BASH->CASE SENSITIVE.

FORMAT->

- *<command> <switch> <parameters>*

AutoFill->

- *Type initials of command & press tab*

Arrow Keys->

- *Scroll up->previous command*
- *Scroll down->current command*

Customize CLI->

- *Tap->Top left corner & select properties.*

Commands Lists:

1-Help:

- Type *help* & enter.
- It will shows all basic CLI commands
- For particular command-> **<command> /?or /help**

2- dir(directory):

- Displays all folders of a drive.
- Dir /b:displays all file names.

3- cd (change directory):

- To move inside folders.
- **Cd..** ->goes one directory backward.
- **Cd /d <drive name> :** \ -> to change drive .
- **Cd.. \..** ->goes 2 directory backward.

4-Command History:(by default no persistent history)

- F7 key(& press enter to exit).
- Type **doskey/history**.

5-Ctrl+C:

- Interrupt running process.

6-cls(clear screen):

- *Clears the command prompt.*

7-tree command:

- *Graphical representation of directory.*
- *Tree/f: show all files*
- *Tree | more:if list is too long(for reading purpose.)*
('l'->pipe)(press enter -> to move 1 line)
(press spacebar -> to move 1 page.)

8-md(make directory) : `D:\>md copy`

9-TYPE(avoid for binary/program created files).

- *Reads/Displays contents of file.*
- *Format->*`type [<drive>:][<path>]<filename>`
(pipe('|') more if text too long).

10-move:

- *Move files from source to target destination.*
- *Format: **move <source file> <destination file>.***

11-delete file:

- *Format ->**del <file name>.***
- *To delete Contents of sub folder : **del /s <subfolder name>.***

12-Echo:

- Return info to cmd.
- Combined with environmental variables(date,time,mssge,&etc) using % <var> %.
- Example:

```
D:\demo>echo "hello world!!"  
"hello world!!"  
  
D:\demo>echo time  
time  
  
D:\demo>echo %date%  
01-08-2021  
  
D:\demo>echo %time%  
13:31:33.29
```

✓ Note:

.bat file extension->batch files.

13-ICACLS(integrity control access control lists)

- Used to grant permissions of file.
- Format-> **icacs {file} /grant {user} :[permissions].**

14-FC(file compare):

- Compares 2 files and displays the differences.
- Format-> **fc <1st file><2nd file>**

15-Xcopy:

- *Copy & compares 2 files .*
- *Faster copying and increased buffer size.*
- *Many switches: /w , /p , /c & etc.*
- *Format-> **xcopy** <source> <destination> <switches>*

16-Robocopy:

- *Robust copy the files and displays the details.*

17-systeminfo:

- *Displays info like os version , patch level ,system type, physical & virtual memory &etc.*

18-date & time:

- *Used with echo-> **echo** %(date/time)%.*
- *Format-> (type) date or time.*
- *Also allows to change date/time.*

19-whoami:

- *Used in windows/linux.*
- *Displays current user info.*

20-tasklist:

- *Displays current running processes in form of tabular data with entries: session name, pid and memory usage.*
- *Used for troubleshoot-ing.*
- **Pid**-> process ID.

21-taskkill:

- *Stops running or struck process.*
- *Format->**taskkill/PID** { PID no. or Image Name}.*

22-SC:

- *SC is a command line program used for communicating with the Service Control Manager and services.*
- *Used to start, stop, raise query ,pause a service.*
- *Create failure actions.*
- *Format->**sc <server> (action) (service name).***

23-fsutil(file system utility):

- *Performs tasks that are related to file allocation table (FAT) and NTFS file systems*
- *Used to change / manage dirty bit.,object id management & etc.*
- *Very powerful tool & support multi commands.*
- *ADMIN controls/ permissions required.*

24-sfc(system file check):

- *Checks protected system files .*
- *Help to recover corrupted OS files.*
- *Allows scanning& validating of files.*
- *ADMIN controls/ permissions required.*
- *Sfc /scannow: scan and repair system files them as well.*

25-chkdsk(check disk):

- *Validates sectors on disk.*
- *Repairs damaged sectors.*
- *Useful in recovery mode .*
- *Required admin priviledges.*
- *Options: /r(locate bad sectors),/f(fix any error).&etc*

26 -ipconfig(IP-configuration):

- *Displays details like:ip address,mac address,netmask,gateways.*
- *(Type)->Ipconfig and enter.*
- *Ipconfig/all-> for complete and descriptive info.*

27-getmac:

- *Displays mac address.*

28-nslookup(name services lookup):

- *Displays DNS server info that device is currently using. .*

29-ping:

- *To validate connectivity.*
- *Easy to use.*
- *to verify that a computer can communicate over the network .*
- ***ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS] [-r count] [-s count] [-w timeout] [-R] [-S srcaddr] [-p] [-4] [-6] target [/?].***

30-tracert:

- *trace route*
- *displays all hops from root to destination*
- *helps in troubleshooting latency.*

31-Address Resolution Protocol (arp):

- *Used to match Ip address with Mac address.*
- *Switches used:-a(for cache),-v(for verbose mode),-s(adds static entry).*
- ***Syntax: arp -[switches].***

The basic protocol functionally divided into two parts:

- ***One part determines a physical address when sending a packet***
- ***Other part answers requests from other machines.***

32)Netstat(network statistics):

- *Displays all active connections along with connection result.(established or not) &protocol used(generally TCP).*
- *Also identifies malware(if any)*

- **Netstat -an:** to get details of all open ports.

33)ftp(file transfer protocol):

- **Used for transferring files ,and insecure by default.**
- **Syntax: open [ftp.\[domain\].com](ftp.[domain].com)** (to connect with any domain)
- **To get or send file through ftp:**

- `get myfile.htm`
- `send myfile.htm`

34)ssh(secure shell):

- *securely & remotely access the terminal.(encrypted comm..)*
- *ssh [user]@[address].*
- *SSH offers several options for user authentication and the most common ones are **passwords** and **public key authentication**.*

35)Net Commands:

Functions:

- *to add user, groups.*
- *Start/stop services.*
- *Connect to other computers.*
- *Used to share files remotely.*

36)assoc (list file association)

- **Syntax: assoc & enter.**
- *If used without parameters, assoc displays a list of all the current file name **extension associations**.*
- *This command is only supported within cmd.exe and is not available from PowerShell.*

37)cipher:

- *Encodes deleted files .*
- *Helps in deleting data by overriding them with 0's and 1's.*
- **Syntax: cipher & enter.**

38)driverquery:

- *Lists all the drivers installed in the system.*
- **Driverquery -v** :to get full details of all drivers installed.

39)color <alphabet>:

- *Changes color of cmd: Example: **color a**.*

40)secure folder: (secure folder prevents from deleting, copying & moving.)

- *To create secure folder : **md aux***
- *To delete secure folder: **rd aux***

Piping & Ampersand:

Single pipe(|):

- *takes standard output and pass it as input.*
 - *Links both commands.*
- `commandA | commandB` *Pipe the output from commandA into commandB*

Double Pipe(||):

- `commandA || commandB` : *Run commandA, if it fails then, run commandB.*

Single Ampersand:

- *Run commandA and then run commandB*
- *Doesn't check for errors*

Double Ampersand:

- *If commandA succeeds ,then only run commandB*
- *Check for errors.*

Windows Redirection:

Single ('>') :

- Redirect command output to a file
- If file exists, overwrite it.
- Else create new file & store output.
- **Example** : `echo "hello word" > hello.doc`

Double ('>>') :

- Redirect command output to a file
- **Will not overwrite file, but append it.**
- **Example** : `echo "hello word" >> hello.doc`

Single ('<') :

- Type a text file and pass the text to command
- Provides input from file (existing file)
- **Example** : `sort < hello.doc`

Command Loops:

- *Run specific task ,a number of times.*
- */f: for file input.*
- */l: list of numbers*

For Loop:

- *Take a set of data*
- *Make a FOR Parameter %%G equal to some part of that data*
- *Perform a command (optionally using the parameter as part of the command).*
- *Repeat for each item of data*
- **Example** : *for /L %G in(1,1,5) Do echo*

While loop:

- *Evaluates if the condition is true,false .*
- *Achieved using if/else/label statements.*

