

1. **login** Access computer; start interactive session

**Syntax:** **login** [username]

2. **logout** Exits the current session

**Syntax:** **logout**

3. **shutdown** Shuts down your Linux system in a way that prevents damage to the file system

**Syntax:** **shutdown** [options] time [message]

#### Options

- f reboots quickly without checking the filesystem
- h halt the system after shutdown
- r restart the computer

4. **su** Creates a new shell session with a different user id and privileges

**Syntax:** **su** [username]

5. **tty** Displays the number of terminal devices that are currently in use

**Syntax:** **tty**

6. **cal**        Displays a calendar of the current year

**Syntax:** **Cal** [option][month][year]

**Options**

- j        displays the calendar using Julian dates, with the days numbered from 1 sequentially to the end of the year
- y        Display a calendar for the entire year

7. **date**        Displays the current date and time as traced by the system clock

**Syntax:** **date** [option][date]

**Options**

- u        displays the date and time in GMT format
- s        set the date

8. **hostname**    Displays the system's network identification name (hostname)

**Syntax:** **hostname** [option][hostname]

**Options**

- a        displays the hostname alias
- i        display the ip address
- d        displays the name of the domain to which the host belongs

9. **man** Displays text only manual pages and is the quick way to get information of most of the utilities installed to the system.

**Syntax:** **man**[options][section][title]

**Options**

- a list all the man pages that match the title
- d displays debugging information
- k lists short description of all the man pages that match the specified string

10. **pwd** This command shows the full path of the current directory

**Syntax:** **pwd**

11. **uname** Provides the wealth of information about the system you are using i.e it returns the name of the OS

**Syntax:** **uname**[option]

**Option**

- a all available information
- m type of processor in use
- n Displays the computer's host name

12. **uptime** Displays the current time, The amount of time the system has been running in the current session

13. **users** Displays the total number of users currently working in the current session

14. **who** Displays the name of the users currently logged into the system.

**Syntax:** **Who**[option]

**Options**

-h display column heading

--help lists available options

15. **whoami** Displays the username of the user currently logged in to the terminal session

16. **bash** Starts the bash shell.

**Syntax:** **Bash**[option][filename]

**Options**

-c Read commands from the specified string

-i Starts bash as an interactive shell

17. **bg** places the process in the background

**Syntax:** **bg**[jobid]

18. **env** It displays or sets the specified variables.

**Syntax:** **env**[option][variable=value][command]

**Options**

- u unset the specified variable
- i ignore the current environment

19. **jobs** Lists all running or suspended jobs

**Syntax:** **Jobs**[option][jobid]

**Options**

- l list job Ids and Process ids
- n lists all altered jobs
- p lists process ids only

20. **kill** Terminates the specified process

**Syntax:** **kill**[option][id]

**Options**

- l list the available signal names and numbers
- s Specifies the signal by name

21. **killall** kills all the processes

**Syntax:** **killall**[option][name]

**Options**

- e Require an exact match of long names
- i Asks for confirmation before killing

22. **ps** Displays the list of running processes

**Syntax:** **Ps**[option][sort key][output field]

**Options**

- a shows all processes on the current terminal
- e shows all processes
- h Shows the process hierarchy

23. **suspend** suspends a command

**Syntax:** suspend

24. **tee** accepts output from the specified command and “splits” the output to the standard output

**Syntax:** command|tee[option]filename

**Options**

- a Append to the file
- i ignore interrupt signal

25. **cd** Change the current directory

**Syntax:** **cd** [directory|path name]

26. **dir** Lists the files in the current directory in case-sensitive, alphabetical order, using a columnar format.

**Syntax:** **dir** [option][pattern]

**Options**

- l list entries using one line for each filename
- a shows all files including hidden files
- c lists files sorted by the time of the last modification to the file's status

27. **find** This command searches the current directory for files with names that match the specified shell pattern.

**Syntax:** **find** pattern

28. **ls** Lists the files in the current directory in case sensitive alphabetical order using a columnar format.

**Syntax:** **ls** [option][pattern]

**Options**

- l Lists files in long listing format

29. **chgrp** Change the ownership of the specified file to the specified group.

**Syntax:** **chgrp** [option] group file

**Options**

- c Display message only when changes are made
- f hide messages
- help display available options

30 **chmod** changes the permission settings of the filename to the mode.

**Syntax:** **chmod** [option] mode filename

**Options**

- c Display message only when changes are made
- f hide messages
- help display available options

1. **chown** Change the ownership of the specified file to the specified user.

**Syntax:** **chown** [option] user [.group] file

**Options**

- c Display message only when changes are made
- f hide messages
- help display available options

2. **cp** copies one source file to the destination file

**Syntax:** **cp** [option][source][destination]

**Options**

- a create archive copies of the files.
- b makes a backup copy
- f remove existing destination file

3. **dd** Copies the file and performs various conversions at the same time.

**Syntax:** **dd** [option]

**Options**

- bs specify the size of the input and output bit streams in bytes.
- Cbs specify the number of bytes to convert at a time

4. **ln** Creates a hard link to the specified target file.

**Syntax:** **ln** [option] target [link name]

**Options**

- b make a backup copy
- d creates hard links to directories
- f remove existing destination files.

5. **Mkdir** Creates a specified directory.

**Syntax:** **mkdir** [option] directory

**Options**

- m creates the directory with the specific permissions.
- p makes parent directory.

6. **mv** renames or moves one source file to a destination file or moves multiple source file to a destination that must be an existing directory.

**Syntax:** **mv** [option][source][destination]

**Options**

- b backup copy
- f remove existing destination file
- i prompt before overwriting existing destination file.

7. **Rm** removes the specified file

**Syntax:** **rm** [option] [file]

**Options**

- d unlink the directory even if it is non empty.
- f ignore non existent files
- i prompt before overwriting existing destination file.

8. **Rmdir** removes the specified directory but only if it is empty.

**Syntax:** **rmdir**[option] directory

**Options**

-p removes associated parent directory

9. **Touch** It changes the time of the last access or modification of the specified filename to the current time.

**Syntax:** **touch** [option] filename

**Options**

-a changes the access time but no other times

-c do not create any file.

10. **Undelete** restores files deleted earlier using safedelelete command

**Syntax:** **undelete** [option][filename]

**Options**

-i Displays information about the file

-l Display a list of safedeleted files that can be restored.

11. **Wc** Display line, word and character count for the specified filename.

**Syntax:** **wc** [option][filename]

**Options**

- c Show the character count only
- l Show the line count only
- w Show the word count only

12. **Df** Displays the amount of disk space used and remaining on all mounted filesystems.

**Syntax:** **df** [option] [filename]

**Options**

- a include all filesystems
- h displays sizes in a human readable format

13. **Du** Displays the amount of disk space used in the current directory.

**Syntax:** **du** [option][filename]

**Options**

- a Show sizes of individual file
- c Print the grandtotal of all arguments after all have been processed.

14. **Fdisk** Launches a menu driven program that partitions a hard disk

**Syntax:** **fdisk** [option] device

**Options**

- l List the current partition table
- s Display the size of the specified partition
- v Display the version number.

15. **Mount** Attaches the device to the specified directory, which will serve as the filesystem's mount point.

**Syntax:** **mount** [option] device [directory]

**Options**

- a Mount all the filesystems listed in /etc/fstab, except those set to noauto
- r Mount the device as read only
- h Display the available options.

16. **Cat** Displays the specified filename on the standard output.

**Syntax:** **cat** [option] [filename]

**Options**

- e Display control and non-printing characters
- n Number all output lines

17. **Cmp** Compares the two specified files to determine whether any difference exist.

**Syntax:** **cmp** [option] filename1 filename2

**Options**

- l Print the Byte numbers of each difference and show the differing values
- s Indicate nothing but generate the exit codes.
- c Print the differing bytes as characters.

18. **Cut** Displays a range of characters from the specified filename.

**Syntax:** **cut** [option][filename]

**Options**

- b output only the bytes specified by range
- c Output only the character specified by range

19. **Grep** Searches filename for lines that match a regular expression.

**Syntax:** **grep** [option] regexp [filename]

**Options**

- A Display n lines of trailing context after matching lines
- b Display the byte offset in the input file before each line of output.

20. **Sort** Sorts the specified filename line by line in character order.

**Syntax:** `sort [option][filename]`

**Options**

- b Ignore leading blanks
- d Use only alphanumeric characters in keys
- f converts lowercase to uppercase characters in keys

21. **Uniq** Removes duplicate lines from a sorted input file and writes to the output file.

**Syntax:** `Uniq [option][input file][output file]`

**Options**

- c Prefix lines by the number of occurrences
- d print the duplicate lines
- i ignore case

## PARAMETER EXPANSION

A shell parameter is an entity that stores a value(which is null). Among the various types of shell parameters are variables. You can create your own variable. Besides this there are several Built in variables as well.

### How to create a variable?

Name=value Create a variable called name and assign value to this variable.

\$name Insert the value of name

## BUILT IN VARIABLES

\$BASH	The location of bash
\$BASH_ENV	The location of the current <b>.bashrc</b> file in use
\$BASH_VERSION	The version number of bash
\$CDPATH	The path to be used when cd is used
\$DIRSTACK	An array variable containing the current contents of the directory stack
\$FCEDIT	The text editor used by default for the fc command
\$FIGIGNORE	A colon-separated list of suffixes to ignore when performing filename completion

## Standard Directory Structure

<b>Directory</b>	<b>Purpose</b>
/	The root directory
/bin	Start-up programs and commands used in single-user mode
/boot	Files used to start the system
/dev	Special files representing system files
/etc	System – level configuration files and scripts
/home	The user's home directory
/lib	Shared library files
/mnt	The mount point for temporary file systems
/opt	The installation directory for commercial software
/root	The home directory for home user.
/tmp	The storage space for temporary file
/usr	The storage space for files that need to be made available system wide
/var	Data file of variable length.