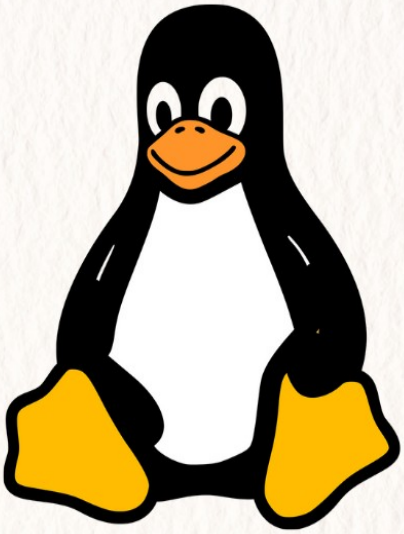

125 LINUX COMMANDS & EXAMPLES

125 Linux Commands and Examples

Ebook created by : Dhanasekar T
Email ID : tkdhanasekar@gmail.com
LinkedIN : https://www.linkedin.com/in/tkdhanasekar/

Table of Contents

1 date

To display date

```
$ date
```

To display the time in GMT/UTC time zone

```
$ date -u
```

To display past dates

```
$ date --date="3 year ago"
```

```
$ date --date="1 month ago"
```

To display future date

```
$ date --date="next wed"
```

```
$ date --date="next month"
```

To set the system date and time

```
$ date --set="Wed Apr 27 14:20:55 IST 2022"
```

2. echo

echo - display a line of text

```
$ echo [string]
```

with double quotes

```
$ echo "Welcome to Linux"
```

with single quotes

```
$ echo ‘Welcome to FOSS’
```

without quotes

```
$ echo Welcome  to Ubuntu
```

3. cat

cat - concatenate files and print on the standard output

To display contents of file

```
$ cat /etc/group
```

To view contents of multiple files

```
$ cat file3.txt file4.txt
```

To create a file with cat command

```
$ cat > file5.txt
```

some contents typed here

`CTRL+D` to save the file

To view cat command with large file size

```
$ cat /proc/cpuinfo | more
```

```
$ cat /proc/cpuinfo | less
```

To display line numbers in file

```
$ cat -n number.txt
```

4. ls

ls - list directory contents

To list files and directories

```
$ ls
```

To long listing of files

```
$ ls -l
```

To view hidden files

```
$ ls -a
```

To list files with human readable format

```
$ ls -lh
```

recursively list Subdirectories

```
$ ls -R
```

To sort files by file size

```
$ ls -lS
```

To order files based on last modified time

```
$ ls -lt
```

5. rm

rm - remove files or directories

To remove or delete file

```
$ rm file.txt
```

To delete a directory recursively

```
$ rm -r old_data/
```

To delete the files interactively

```
$ rm -i file.txt
```

To Delete files forcefully

```
$ rm -f file.txt
```

To delete all the .txt files

```
$ rm -f *.txt
```

6. cp

cp - copy files and directories

To copy a file old_file.txt

```
$ cp old_file.txt new_file.txt
```

To copying multiple files to a directory

```
$ cp file1_name file2_name file3_name /opt
```

To copying a directory or folder

```
$ cp -r /home/klug /opt/backup
```

To preserve mode, ownership and timestamps when copying

```
$ cp -p file.txt /opt/backup/
```

To copy the files and directory forcefully

```
$ cp -f file.txt /opt/backup
```

7. mv

mv - move or rename files

```
$ mv [Option] source destination
```

To rename a file1.txt to file2.txt

```
$ mv file1.txt file2.txt
```

```
$ mv file1.txt /home/venus/Documents/file2.txt
```

To move multiple directories from one location to another

```
$ mv dir1 dir2 dir3 /opt/
```

8. history

history - displays a list of commands used in the terminal session

To display the list of commands used

```
$ history
```

To show only the latest 10 entries

```
$ history 10
```

To run the 100th command again in history

```
$ !100
```

To repeat the last command

```
$ !!
```

To remove a command from history

```
$ history -d event_number
```

```
$ history -d 100
```

To remove whole history

```
$ history -c
```

To view the last 10 commands

```
$ history | tail
```

9. whoami

whoami - print effective userid

```
$ whoami
```

10. hostname

hostname - show or set the system's host name

To display the system hostname

```
$ hostname
```

To get all IP addresses

```
$ hostname -I
```

To set the hostname

```
$ sudo hostname <new_hostname>
```

To set kaniyam as hostname

```
$ sudo hostname debian
```

11. uname

uname - print system information

To print uname without options

```
$ uname
```

To print all information

```
$ uname -a
```

To print the kernel name

```
$ uname -s
```

To print the kernel release

```
$ uname -r
```

12. uptime

uptime - Tell how long the system has been running

uptime command without any options

```
$ uptime
```

To show uptime in pretty format

```
$ uptime -p
```

To display the date/time since when the system has been running

```
$ uptime -s
```

13. pwd

pwd - print name of current/working directory

To get working directory path

```
$ pwd
```

14. mkdir

mkdir - make directories

To create a directory

```
$ mkdir dir1
```

To display verbose message for every directory created.

```
$ mkdir -v directory_1 directory_2 directory_3
```

To create multiple directories

```
$ mkdir {dir1,dir2,dir3}
```

To create parent directories

```
$ mkdir -p /dir_1/dir_2/dir_3
```

```
$ mkdir -p -v /dir_1/dir_2/dir_3
```

To set permissions for the directories

```
$ mkdir -m a=rwx [directory_name]
```

```
$ mkdir -m777 dir_1
```

```
$ mkdir -m755 dir_2
```

```
$ mkdir -m766 dir_3
```

15. rmdir

rmdir - remove empty directories

To remove a single empty directory

```
$ rmdir sample_dir1
```

To remove multiple directories using rmdir

```
$ rmdir sample_dir1 sample_dir2
```

16. cd

cd - change directory

change current directory to /usr/share

```
$ cd /usr/share/
```

To change current directory to parent directory

```
$ cd ..
```

To change to home directory

```
$ cd
```

17. locate

locate - find files by name, quickly

To locate a file name

```
$ sudo updatedb
```

```
$ locate file_name
```

create a file secret.txt in somewhere in system

```
$ locate secret.txt
```

18. man

man - an interface to the system reference manuals

```
$ man df
$ man du
$ man uptime
```

1 Executable programs or shell commands
2 System calls (functions provided by the kernel)
3 Library calls (functions within program libraries)
4 Special files (usually found in /dev)
5 File formats and conventions, e.g. /etc/passwd
6 Games
7 Miscellaneous
8System administration commands (usually only for root)
9 Kernel routines [Non standard]

19. who

who - show who is logged on

To print who command output without options

```
$ who
```

To print same as -b -d --login -p -r -t -T -u

```
$ who -a
```

To check the current runlevel

```
$ who -r
```

To view the time of last system boot

```
$ who -b
```

20. wc

wc - print newline, word, and byte counts for each file

wc without options will display (number of lines),(number of words) and (number of bytes) of the file

```
$ wc file.txt
```

To Count Number of Lines

```
$ wc -l file.txt
```

To Display Number of Words

```
$ wc -w file.txt
```

To Count Number of Bytes and Characters

```
$ wc -c file.txt
```

```
$ wc -m file.txt
```

21. | piping

To find wc of file /proc/cpuinfo

```
$ cat /proc/cpuinfo | wc
```

```
$ cat filename | wc
```

To filter a keyword from a file.txt

```
$ cat filename | grep <keyword>
```

```
$ cat /proc/cpuinfo | grep vendor_id
```

22. vim

vim - Vi IMproved, a programmer's text editor

To create a file

```
$ vim filename
```

To go Insert Mode

```
press I
```

Once the editor is in insert mode, start writing the content in the file.

To Save the file and exit from the editor

```
[Esc] SHIFT+ :wq!
```

or

```
[Esc] SHIFT+ :x
```

To quit from the file without saving

```
Esc SHIFT+ :q!
```

23. find

find - search for files in a directory hierarchy

To find all the files whose name is secret.txt in current working directory

```
$ find . -name secret.txt
```

To find files in home directory

```
$ find /home -name secret.txt
```

To find all python files in a directory

```
$ find . -type f -name "*.py"
```

24. env

env - run a program in a modified environment

To print out a list of all environment variables

```
$ env
```

25. export

export - It is used to mark variables and functions to be passed to child processes

To display all exported variables

```
$ export
```

To view all exported variables on the current shell

```
$ export -p
```

The variable ‘community’ has been assigned the value ‘fedora’

```
$ community=fedora
```

```
$ export community
```

check with

```
$ printenv community
```

26. df

df - report file system disk space usage

To display all the file system

```
$ df -a
```

To display size in human readable format

```
$ df -h /home/
```

To get complete grand total

```
$ df -h --total
```

To display file type

```
$ df -T /home/venus
```

To display disk space usage of current dir

```
$ df -Th
```

27. less

less - used to read the contents of a text file one page(one screen) at a time

```
$ less filename
```

```
$ less /proc/cpuinfo
```

28. sort

sort - is used to sort a file, arranging the records in a particular order

```
$ cat sort.txt
assam
tamilnadu
chattisgarh
delhi
gujarat
himachal pradesh
kerala
bihar
```

To sort arrange the sort.txt

```
$ sort sort.txt
```

To Save Output to File

```
$ sort file.txt > sort_output.txt
```

To Sort Multiple Files

```
$ sort file1.txt file2.txt
```

To Sort in Reverse Order

```
$ sort -r sort.txt
```

To Remove Duplicate Entries

```
$ sort -u file.txt
```

29. uniq

uniq - it is used to report or filter out repeated lines in a file.

```
$ cat uniq.txt
redhat
debian
ubuntu
ubuntu
centos
fedora
fedora
fedora
fedora
```

To report or filter out for lines that are adjacent and repeated

```
$ uniq uniq.txt
```

To only print unique lines

```
$ uniq -u uniq.txt
```

To prefix lines by the number of occurrences

```
$ uniq -c uniq.txt
```

To only print duplicate lines, one for each group

```
$ uniq -d uniq.txt
```

To print all duplicate lines

```
$ uniq -D demo.txt
```

30. cut

cut - remove sections from each line of files

```
$ cat cut.txt
Alpha is first line
Beta is second line
Charlie is third line
Delta is fourth line
```

To display 1st character from each line of a file

```
$ cut -c1 cut.txt
```

To display 2nd character from each line of a file

```
$ cut -c2 cut.txt
```

To extract first 3 characters of each line from file.txt

```
$ cut -c1-3 cut.txt
```

To extract 7 characters from the beginning of each line

```
$ cut -c-7 cut.txt
```

31. fmt

fmt - simple optimal text formatter Reformat each paragraph in the files, writing to standard output

```
$ cat fmt.txt
Hai
all Welcome
to
Ubuntu

$ fmt fmt.txt
Hai all Welcome to Ubuntu
```

32. head

head - output the first part of files

To display the first 10 lines default of head command

```
$ head /proc/cpuinfo
```

To show the first 6 lines

```
$ head -n 6 /proc/cpuinfo
```

To display multiple files

```
$ head file1.txt file2.txt
```

To redirect output to a text file

```
$ head /proc/cpuinfo > head_output.txt
```

To display head with Pipeline

```
$ ls /etc | head
```

```
$ ls -t | head -n 4 | sort
```

33. tail

tail - output the last part of files

To display the last 10 lines of a file

```
$ tail /proc/cpuinfo
```

To show the last 6 lines

```
$ tail -n 6 /proc/cpuinfo
```

To Display the last n lines from multiple files

```
$ tail -n 4 file1.txt file2.txt
```

To Save the output of tail command to a text file

```
$ tail -n 10 /proc/cpuinfo > tail_output.txt
```

To use pipes

```
$ tail /var/log/messages | sort
```

```
$ tail /var/log/messages | tail -n 6 | sort
```

To monitor real-time log files

```
$ tail -f /var/log/messages
```

34. nl

nl - used for numbering lines, accepting input either from a file or from STDIN

```
$ cat nl.txt
Apache
Squid
Samba
DNS
DHCP
```

To display a file with line numbers

```
$ nl nl.txt
```

To number all lines including empty lines

```
$ nl -b a file.txt
```

To add a string after line numbers

```
$ nl -s "..." file.txt
```

35. split

split - is used to split large files into smaller files

syntax

```
$ split {options} {file_name} {prefix}
```

```
$ cat split.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
This is line 6
This is line 7
This is line 8
This is line 9
This is line 10
```

To split split.txt with verbose option

```
$ split split.txt --verbose
```

To split files with customize line numbers

```
$ split -l2 split.txt --verbose
```

To split file with customize suffix

```
$ split -l2 split.txt my_file
```

36. tac

tac - is used to concatenate and print files in reverse

```
$ cat tac.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
```

To print files in reverse

```
$ tac tac.txt
```

37. last

last - show a listing of last logged in users

To list last five users logged in

```
$ last -5
```

To display information like system down entries and run level changes

```
$ last -x
```

38. tr

tr - is used to translate and/or delete characters from stdin input and writes to stdout

```
$ cat tr.txt
linux OS is powerful
linux os is versatile
linux os is best
```

To change all lowercase letters in the text to uppercase and vice versa

```
$ cat tr.txt | tr [a-z] [A-Z]
```

To save the results written to stdout in a file

```
$ cat tr.txt | tr [a-z] [A-Z] > tr_output.txt
```

To send input to tr using the input redirection and redirect the output to a file

```
$ tr [a-z] [A-Z] < tr.txt > tr_output.txt
```

39. sed

sed - stream editor for filtering and transforming text

Basic text substitution using ‘sed’

```
$ echo "Fedora is linux OS" | sed 's/Fedora/Ubuntu/'
```

```
$ cat linux.txt
linux is a very popular os.
linux is easy to use. linux is easy to learn.
linux is a versatile os
```

To make all occurrences to change from linux to unix

```
$ sed 's/linux/unix/g' linux.txt
```

To replace words or characters with ignore character case

```
$ sed 's/linux/unix/gi' linux.txt > sed_output.txt
```

40. paste

paste - used to join files horizontally

```
$ cat name
apache
nginx
mysql
ftp
jenkins

$ cat server
webserver
webserver
db server
file server
integration server
```

To merge the files in parallel with default delimiter as tab

```
$ paste name server
```

To merge files with delimiter as any character

```
$ paste -d "|" name server
```

41. join

join - join lines of two files on a common field , join combines lines of files on a common field

```
$ cat file1.txt
1 andhra
2 tamilnadu
3 kerala
4 karnataka
5 Delhi

$ cat file2.txt
1 101
2 102
3 103
4 104
5 105
```

To join the 2 files

```
$ join file1.txt file2.txt
```

To create a new file with the joined contents

```
$ join file1.txt file2.txt > file3.txt
```

42. file

file - determine file type

To determine file type

```
$ file -b filename.py
```

```
$ file -b file.img
```

```
$ file -b file.txt
```

```
$ file -b file.pdf
```

To display all files's file type

```
$ file *
```

To display the file type of files in specific range

```
$ file [a-d]*
```

```
$ file [e-h]*
```

43. touch

touch – create empty files and change file timestamps

To Create an Empty File

```
$ touch file.txt
```

To Create Multiple Files

```
$ touch file1.txt file2.txt file3.txt file4.txt
$ mkdir test && cd test
$ touch file{0..1000}.txt
```

To Set File Timestamp Using Date String

```
$ touch -d tomorrow demo.txt
```

To Explicitly Set the Access and Modification times

```
$ touch -c -t YYMMDDHHMM demo.txt
$ touch -c -t 2412311159
```

44. cal

cal - displays a calendar

To Show current month calendar

```
$ cal
```

To Show calendar of selected month and year

```
$ cal August 2025
```

To Show the calendar of current year

```
$ cal -y
```

To Show calendar of previous, current and next month

```
$ cal -3
```

45. rev

rev - reverse lines character wise

```
$ cat rev.txt
```

This is sample test file

To reverse the text

```
$ rev rev.txt
```

```
$ echo This is sample file | rev
```

```
$ rev
linux
foss
debian
```

46. >

> I/O redirection

write to a file

```
$ cat > sample.txt
line1
line2
line3
line4
ctrl+D
```

write output to file

```
$ wc file_name > file_output.txt
$ free -h | wc > free_output.txt
$ df -Th | wc > df_output.txt
```

47. <

< - I/O redirection

Input to a command and writing output to a file

```
$ wc < input_sample.txt > wc_output.txt
```

48. >>

>> - append

To append lines to a file.txt

```
$ echo "This is append line1" >> file.txt
```

```
$ cat >> file.txt
```

This is added line2

This is added line3

This is added line4

ctrl+D

49. tee

tee - read from standard input and write to standard output and files

To append a line of text to a file

```
$ echo "This is demo msg " | tee -a demo.txt
```

To display output of df -Th and write to a df.txt

```
$ df -Th | tee df.txt
```

To display output of free -h and write to a free.txt

```
$ free -h | tee free.txt
```

50. xargs

xargs - build and execute command lines from standard input

To find a file and remove using xargs

```
$ find /home/venus/ -name "test.py" -type f | xargs rm -f
```

To find a file and grep a particular keyword

```
$ find /home/venus/ -name "free.log" -type f | xargs grep
"04:08:01"
```

To read items from file

```
$ xargs -a file_name
```

To list number of lines/words/characters in each file

```
$ ls free.* | xargs wc
```

51. grep

grep - print lines that match patterns

```
$ cat grep_example.txt
This is line number one
this is line number two
THIS is line number three
this is line 4
This is line 5
```

To search for the given string in a single file

```
$ grep "this" grep_example.txt
```

To check for the given string in multiple files

```
$ grep "this" grep_example.txt file2.txt
```

To search case insensitive using grep -i

```
$ grep -i "4" grep_example.txt
```

To count the number of matches using grep -c

```
$ grep -c this grep_example.txt
```

To show line number while displaying the output using grep -n

```
$ grep -n "this" grep_example.txt
```

52. jobs

jobs - used to list the jobs running in the background

To run some jobs in background

```
$ ping google.com
```

CTRL+Z

```
$ man ls
```

CTRL+Z

To lists jobs running in background

```
$ jobs
```

To display jobs with process id

```
$ jobs -l
```

To display the process ID or jobs for the job whose name begins with “p” and “m”

```
$ jobs %p
```

```
$ jobs %m
```

To display PIDs only

```
$ jobs -p
```

53. fg

fg - is used to put a background job in foreground.

First list the jobs running in background

```
$ jobs -l
```

To make the job with id [1] to run in foreground

```
$ fg %1
```

To make the job with id [2] to run in foreground

```
$ fg %2
```

54. bg

bg - is used to place foreground jobs in background.

```
$ ping google.com
```

press CTRL+Z

To view running jobs

```
$ jobs -l
```

To resume the job ping google.com job with job number 1

```
$ bg %1
```

To kill the job # ping google.com

```
$ kill -9 <pid>
```

55. runlevel

runlevel - Print previous and current SysV runlevel

To see the current runlevel of the system

```
$ runlevel
```

0 - Halt

1 - Single-user mode

2 - Not used (user-definable)

3 - Full multi-user mode

4 - Not used (user-definable)

5 - Full multi-user mode (with an X-based login screen)

6 - Reboot

56. init

init - initializes and controls processes

To restart the system

```
$ init 6
```

To shut down system

```
$ init 0
```

0 - Halt

1 - Single-user mode

2 - Not used (user-definable)

3 - Full multi-user mode

4 - Not used (user-definable)

5 - Full multi-user mode (with an X-based login screen)

6 - Reboot

57. ps

ps - report a snapshot of the current processes.

To display processes for the current shell

```
$ ps
```

To Display processes in BSD format

```
$ ps aux
```

To print user running processes

```
$ ps -x
```

To print all processes in different formats

```
$ ps -A
```

To display full-format listing

```
$ ps -ef
```

58. pstree

pstree - is used to display the parent-child relationship in a hierarchical format

To print pstree without any option

```
$ pstree
```

To Display the tree hierarchy of a user processes

```
$ pstree -p <username>
```

59. top

top - display Linux processes

To list all processes

```
$ top
```

To exit after n repetitions

```
$ top -n 3
```

```
$ top -n 5
```

To display all user-specific processes

```
$ top -u <username>
```

```
$ top -u root
```

```
$ top -u venus
```

To save the running top command results output

```
$ top -n 1 -b > top_output.txt
```

60. htop

htop - interactive process viewer

To view the running processes

```
$ htop
```

To view the processes of a user

```
$ htop -u <username>
```

61. kill

kill - send a signal to a process

To display all the available signals

```
$ kill -l
```

To use PID with the kill command

```
$ kill pid
```

To kill multiple processes at once

```
$ kill <pid1> <pid2> <pid3>
```

To forcefully kill single process

```
$ kill -9 <pid>
```

To forcefully kill multiple process

```
$ kill -9 <pid1> <pid2>
```

To find signal name

```
$ kill -l 3
$ kill -l 9
$ kill -l 15
```

62. killall

killall - kill processes by name

To kill a program by name

```
$ killall <program_name>
```

To killall firefox

```
$ killall firefox
```

To get a list of signals that killall can send

```
$ killall -l
```

To Kill multiple processes interactively

```
$ killall -i <program1> <program2>
```

63. pidof

pidof - find the process ID of a running program

To find the PID of the SSH

```
$ pidof sshd
```

To find the PID of firefox, top

```
$ pidof firefox
```

```
$ pidof top
```

64. nice

nice - run a program with modified scheduling priority

- ‘nicer’ processes require fewer resources
- Nice value ranges from +19(very nice) to −20 (not very nice)
- Non-root users can only specify values from 1 to 19
- the root user can specify the full range of values

To check all nice values of all processes

```
$ top
```

To check the nice value of htop process

```
$ ps -el | grep htop
```

To set the priority of a process

```
$ nice -n <number><process name>
```

```
$ nice -10 htop
```

To set the negative priority for a process

```
$ sudo nice --n <number><process name>
```

```
$ sudo nice --10 htop
```

65. renice

renice - alter priority of running processes

renice changes the niceness of existing processes

To change the priority of the running process.

```
$ sudo renice -n 10 -p <PID>
```

66. useradd

useradd - create a new user or update default new user information

To add a new user without home directory

```
$ sudo useradd user_name
```

To create user with home directory

```
$ sudo useradd -m user_name
```

To create a user with a specific User ID

```
$ sudo useradd -u 1004 user_name
```

67. adduser

adduser - add a user to the system

To add a new user

```
$ adduser user_name
```

68. passwd

passwd - change user password

To change system user’s password

```
$ passwd
```

To change password for root

```
$ sudo passwd root
```

To display user status Information

```
$ sudo passwd -S <user_name>
```

To display information of all users

```
$ sudo passwd -Sa
```

69. userdel

userdel - delete a user account and related files

To delete a user account

```
$ sudo userdel user_name
```

To remove the user’s home directory and mail spool

```
$ sudo userdel -r user_name
```

To forcefully remove the user account

```
$ sudo userdel -f user_name
```

70. deluser

deluser - remove a user or group from the system

To delete an user account

```
$ sudo deluser user_name
```

To delete or account including deleting home directory

```
$ sudo deluser --remove-home user_name
```

To delete account even while the user logged in

```
$ sudo deluser --force <user_name>
```

71. groupadd

groupadd - create a new group

To create a group

```
$ sudo groupadd group_name
```

To create a group with specific groupid

```
$ sudo groupadd <group_name> -g 1234
```

72. addgroup

addgroup - add group to the system

To add a new group

```
$ sudo addgroup <group_name>
```

To add a new group with specified group id

```
$ sudo addgroup group_name --gid 6789
```

73. groupdel

groupdel - delete a group

To delete a group

```
$ sudo groupdel group_name
```

74. delgroup

delgroup - remove a group from the system

To remove a group

```
$ sudo delgroup group_name
```

75. groups

groups - print the groups a user is in

```
$ groups [username]
```

To display group membership for the current user

```
$ groups
```

To find groups of root

```
# groups
```

76. id

id - print real and effective user and group IDs

To print your own id without any options

```
$ id
```

To find a specific users id

```
$ id -u <user_name>
```

To find a specific users GID

```
$ id -g <user_name>
```

To find out UID and all groups associated with a username

```
$ id <user_name>
```

77. usermod

usermod - modify a user account

To add a user to sudo group

```
$ sudo usermod -aG sudo <user_name>
```

To add group to an existing user

```
$ sudo usermod -aG group_name user_name
```

78. ln

ln - creates the hard and symbolic links between the files.

To create hard link with the name sample_hardlink_file.txt

```
$ ln sample_file.txt sample_hardlink_file1.txt
$ ln sample_file.txt sample_hardlink_file2.txt
$ ln sample_file.txt sample_hardlink_file3.txt
```

even the original file name sample_file.txt is deleted we can access the file with sample_hardlink_file1.txt, sample_hardlink_file2.txt, sample_hardlink_file3.txt

To create symbolic or soft link to a file

```
$ ln -s /home/venus/Documents/file.txt softlink_file.txt
$ ls -al softlink_file.txt
```

To create symbolic or soft link to a directory

```
$ ln -s /home/venus/music/ music
$ ls -al music
```

79. unlink

unlink - call the unlink function to remove the specified file

syntax

```
$ unlink filename
$ unlink dir_name
```

To create hard link with the name sample_link_file.txt

```
$ ln sample_file.txt sample_hardlink_file1.txt
```

To delete the hardlink

```
$ unlink sample_hardlink_file1.txt
```

To create symbolic or soft link to a file

```
$ ln -s /home/venus/Documents/file.txt softlink_file.txt
```

To delete the symbolic link

```
$ unlink softlink_file.txt
```

To delete the symbolic link for directory

```
$ ln -s /home/venus/music/ music
$ unlink music
```

80. stat

stat - display file or file system status

To view the file details

```
$ stat file.txt
```

To Show only octal file permissions

```
$ stat -c %a file.txt
```

```
$ stat --format="%a %n" file.txt
```

To Show the owner and group of a file

```
$ stat --format="%U %G" file.txt
```

81. chmod (symbolic & Numeric)

chmod - change file mode bits

Symbolic Method

u - The file owner.

g - The users who are members of the group.

o - All other users.

a - All users, equal to ugo.

r – read w – write x – execute

- Removes the specified permissions.

+ Adds specified permissions.

= Changes the current permissions to the specified permissions

To set user , group and others full permissions

```
$ chmod ugo=rwx file.txt
```

To remove write and execute permission for others

```
$ chmod o-wx file.txt
```

To remove write permission for group

```
$ chmod g-w file.txt
```

To set sticky bit to a given directory or file

```
$ chmod o+t <dir_name>
```

```
$ chmod o+t file.txt
```

chmod (numeric mode)

numeric method

r (read) = 4

w (write) = 2

x (execute) = 1

no permissions = 0

rwx=4+2+1=7

rw= 4+2=6

rx= 4+1=5

To set read , write , execute permission to users , group and others

```
$ chmod 777 file.txt
```

To set read , write , execute permission to users and read permission only for group and others

```
$ chmod 744 file.txt
```

To set read , write , execute permission for user and no

permission for group and others

```
$ chmod 700 file.txt
```

To set permission recursively for folder

```
$ chmod -R 755 /var/www/
```

To set read, write, and execute permissions, and a sticky bit to a given directory

```
$ chmod 1777 dir_name
```

82. chown

chown - change file owner and group to another existing owner and group

To change the owner of a file

```
$ sudo chown frappe file.txt
```

To change the group ownership of a file

```
$ sudo chown :frappe file.txt
```

To change both owner and the group

```
$ sudo chown frappe:frappe file.txt
```

To change the owner/group of the files by traveling the directories recursively

```
$ sudo chown -R venus:venus directory_name
```

83. chgrp

chgrp - change group ownership

To change a directory group ownership

```
$ sudo chgrp <group_ownership> <dir_name>
```

To change group ownership of a file

```
$ sudo chgrp <group_ownership_name> file.txt
```

To recursively change group ownership

```
$ sudo chgrp -R frappe directory_name/
```

84. umask

umask - is used to set default permissions for files or

directories the user creates.

To calculate umask value

```
$ umask
```

To Displays the current mask

```
$ umask -p
```

To set the default permissions for all new files or folders to 644 and 755 then umask value is

```
$ umask 022
```

for folders 777-022 = 755

for files 666-022 = 644

85. gpasswd

gpasswd - administer /etc/group and /etc/gshadow

To add user user1 to the group mint

```
$ sudo gpasswd -a user1 mint
```

To remove user user1 from the group mint

```
$ sudo gpasswd -d user1 mint
```

86. whatis

whatis - display one-line manual page descriptions

```
$ whatis free
```

```
$ whatis cp ls df du free
```

To get debugging information

```
$ whatis -d pwd
```

To use regular expressions with this

```
$ whatis -r free
```

To use wildcard with this

```
$ whatis -w du
```

87. w

w - Show who is logged on and what they are doing

To show who is logged on and what they are doing

```
$ w
```

88. which

which - locate a command

syntax

```
$ which -a [argument]
```

To locate a command

```
$ which -a touch
```

```
$ which -a free
```

```
$ which -a du
```

```
$ which -a df
```

```
$ which python3
```

89. whereis

whereis - is used to find the location of the binary, source, and manual page files

To find the directories where the whereis command search

```
$ whereis -l
```

To get information about the commands

```
$ whereis du
```

```
$ whereis free
```

```
$ whereis bash
```

To get output for multiple commands

```
$ whereis du free bash
```

To search only for sources

```
$ whereis -s free
```

To search only for man files

```
$ whereis -m du
```

To search only for binaries

```
$ whereis -b cp
```

```
$ whereis -b free
```

90. apropos

apropos - search the manual page names and descriptions

```
$ apropos <command_name>
$ apropos useradd
$ apropos adduser
$ apropos df
$ apropos free
```

91. chattr

chattr - change file attributes on a Linux file system

To add attributes on files and immutable to secure from deletion create file sample.txt

```
$ sudo chattr +i sample.txt
```

To list the file attributes

```
$ lsattr sample.txt
```

To unset attribute on Files

```
$ sudo chattr -i sample.txt
```

To open the file only in append mode

```
$ sudo chattr +a sample.txt
```

To secure entire directory important_folder and its files

```
$ sudo chattr -R +i important_folder
```

To unset it

```
$ sudo chattr -R -i important_folder
```

92. lsattr

lsattr - is used to list the attributes of a file or directory

To display all the files and directories in the current directory along with their file attributes

```
$ lsattr
```

```
$ lsattr file.txt
```

```
$ lsattr dir_name
```

To list all files in directories

```
$ lsattr -a
```

To Recursively list attributes of directories and their contents

```
$ lsattr -R /etc/ssh/
```

93. zip

zip - package and compress (archive) files

Create files for archiving

```
$ touch file{1..5}.txt
```

```
$ zip zipfile *.txt
```

To list zip file contents

```
$ zip -sf zipfile.zip
```

94. unzip

unzip - list, test and extract compressed files in a ZIP archive

To extract all files from the zip archive

```
$ unzip zipfile.zip
```

To display the content of the zip file without extracting

```
$ unzip -l zipfile.zip
```

To extract zip files with suppressing output

```
$ unzip -q zipfile.zip
```

95. sudo

sudo - allows a permitted user to execute a command as the superuser or another user

To run command as a root user

```
$ sudo <command>
```

```
$ sudo chmod
```

To add a user to the sudo group

```
$ sudo usermod -aG sudo <user_name>
```

96. su

su - run a command with substitute user and group ID

su command without any option

```
$ su
```

To switch to root user

```
$ su -
```

su command to make the shell a login shell

```
$ su - frappe
```

To Use su with sudo command

```
$ sudo su - frappe
```

97. ulimit

ulimit - allows viewing or limiting system resource amounts that individual users consume

To find the resource amount that the current user has access to use

```
$ ulimit
```

To get a detailed report with all resource limits for the current user

```
$ ulimit -a
```

98. enable

enable command is used to enable or disable the shell built-in commands.

To list the shell builtin commands which are enabled

```
$ enable
```

To disable the shell builtin command alias

```
$ enable -n alias
```

check the list with

```
$ enable
```

```
$ alias c=clear
```

To make the alias command to enable again

```
$ enable alias
```

```
$ alias c=clear
```

```
$ c
```

To disable history command

```
$ enable -n history
```

check with

```
$ enable
```

```
$ history
```

To enable the history command

```
$ enable history
```

```
$ history
```

99. type

type - is used to display information about the command type

To find the type of ls command

```
$ type ls
```

To find the type of wc command

```
$ type wc
```

```
$ type type
```

To display more than one argument

```
$ type df free sleep head
```

To display the command is an alias, keyword or a function and path of an executable

```
$ type -a pwd
```

```
$ type -a ls
```

100. shutdown

shutdown - is used to shutdown the system in a safe way

To shutdown the system at a specified time 6 P.M

```
$ sudo shutdown 18:00
```

To schedule a system shutdown in 30 minutes from now

```
$ sudo shutdown +30
```

To cancel a scheduled shutdown

```
$ sudo shutdown -c
```

To shutdown the system immediately

```
$ sudo shutdown now
```

To halt your system

```
$ sudo shutdown -H
```

To make shutdown power-off machine

```
$ sudo shutdown -P
```

101. reboot

reboot - is used restart or reboot the system

To restart system

```
$ sudo reboot
```

```
$ sudo shutdown -r now
```

To restart remote server

```
$ ssh root@remote-server /sbin/reboot
```

To force immediate reboot

```
$ sudo reboot -f
```

102. help

help - displays the information about the built-in commands present in the Linux shell

To display information about help command

```
$ help help
```

```
$ help cd
```

To display short description about commands

```
$ help -d help
```

```
$ help -d ls
```

```
$ help -d cd
```

To display usage in pseudo-manpage format

```
$ help -m help
```

```
$ help -m pwd
```

103. at , atq , atrm

at, batch, atq, atrm - queue, examine, or delete jobs for later execution

To execute a command at 13.00 hours

```
$ at 13.00
```

warning: commands will be executed using /bin/sh

at Thu Aug 3 13:00:00 2023

```
at> df -Th > df.txt
```

```
at> CTRL+D
```

To list the jobs in queue

```
$ atq
```

To cancel the jobs in queue

```
$ atrm <job_number>
```

104. nologin

nologin – To add user with no shell access

To add a user without a login shell

```
$ sudo useradd -s /sbin/nologin user_name
```

To check

```
$ cat /etc/passwd | grep nologin
```

105. chsh

chsh - change login shell

To enable the shell access for the user

```
$ sudo chsh -s /bin/bash <username>
```

To change the login shell for user

```
$ chsh
```

106. crontab

crontab - maintain crontab files for individual users

To list crontab entries

```
$ crontab -l
```

create a script name sample.sh to display amount of free and used memory in the system with timestamp

```
$ cat > sample.sh
```

```
#!/bin/bash

free -h

echo "this is memory available"

current_time=$(date)

echo "time is: $current_time"
```

To execute this script for every 2 minutes and write to file called free.log

create free.log file

```
$ touch free.log
```

then

```
$ crontab -e
```

```
*/2 * * * * /bin/bash /home/venus/sample.sh >>
/home/venus/free.log
```

restart the cron.service

```
$ sudo systemctl restart cron.service
```

```
$ crontab -l
```

107. wget

wget - is used for non-interactive download of files from the Web

To download a file with wget

```
$ wget http://path/to/url
```

To download pdf from

```
$ wget
https://github.com/tkdhanasekar/linux_commands_examples/blob/main
/500%2BLinuxCommands%26examples.pdf
```

To download multiple files from a file

```
$ cat downloads.txt
```

http://path/to/url1

http://path/to/url2

http://path/to/url3

```
$ wget -i downloads.txt
```

To download multiple files with http and ftp

```
$ wget http://path/to/url1 http://path/to/url2
```

108. du

du - estimate file space usage

To check the disk usage summary of a directory

```
$ du /etc
```

```
$ du /home
```

To check disk usage in a human-readable format

```
$ du -h /etc
```

```
$ du -h /home/venus
```

To check the total usage size of a particular directory

```
$ du -sh /etc
```

To check the total usage size of current directory

```
$ du -hs *
```

To print the grand total for a directory

```
$ du -chs *
```

```
$ du -hsc /home/venus
```

109. systemctl

systemctl - Control the systemd system and service manager

To start service

```
$ sudo systemctl start mariadb.service
$ sudo systemctl start apache2.service
```

To stop service

```
$ sudo systemctl stop mariadb.service
$ sudo systemctl stop apache2.service
```

To restart service

```
$ sudo systemctl restart mariadb.service
$ sudo systemctl restart apache2.service
```

To check status of service

```
$ sudo systemctl status mariadb.service
$ sudo systemctl status apache2.service
```

To enable service

```
$ sudo systemctl enable mariadb.service
$ sudo systemctl enable apache2.service
```

To disable service

```
$ sudo systemctl disable mariadb.service
$ sudo systemctl disable apache2.service
```

To see the status of all services

```
$ sudo systemctl list-units --type=service
```

To List services by status

```
$ sudo systemctl list-units --type=service --state=active
$ systemctl list-units --type=service --state=running
$ systemctl list-units --type=service --state=stopped
$ systemctl list-units --type=service --state=enabled
$ systemctl list-units --type=service --state=disabled
$ systemctl list-units --type=service --state=failed
```

To Kill a service with signal 9

```
$ sudo systemctl kill -s 9 <service_name>
```

To reload daemon

```
$ systemctl daemon-reload
```

110. tar

tar - an archiving utility

```
$ mkdir tar_examples && cd tar_examples
$ mkdir files && cd files
$ touch file{0..1000}.txt
$ cd ..
```

To make archiving using tar

```
$ tar cf myfiles.tar files
```

To list files without extracting

```
$ tar tf myfiles.tar
```

To extract files myfiles.tar

```
$ tar -xvf myfiles.tar
```

To make archiving using gunzip tar.gz

```
$ tar cf myfiles.tar.gz files
```

To list files without extracting from tar.gz

```
$ tar tf myfiles.tar.gz
```

To extract files from myfiles.tar.gz

```
$ tar -xvzf myfiles.tar.gz
```

111. apt

apt - command-line interface for the package management

To Update System Packages

```
$ sudo apt update
```

To install packages

```
$ sudo apt install <package_name1> <package_name2>
```

```
$ sudo apt install vsftpd apache2 mariadb-server
```

To Check All Dependencies of a Package

```
$ sudo apt depends bind9
```

```
$ sudo apt depends vsftpd
```

To Search for a Package

```
$ sudo apt search apache2
```

```
$ sudo apt search vsftpd
```

To View Information About Package

```
$ sudo apt show apache2
```

```
$ sudo apt show vsftpd
```

To Upgrade System

```
$ sudo apt upgrade
```

To Remove Unused Packages

```
$ sudo apt autoremove
```

To Clean Old Repository of Downloaded Packages

```
$ sudo apt autoclean
```

To Remove Packages with its Configuration Files

```
$ sudo systemctl stop apache2
$ sudo apt purge apache2
$ sudo systemctl stop vsftpd
$ sudo apt purge vsftpd
```

To List Packages

```
$ sudo apt list
```

112. add-apt-repository

add-apt-repository - Adds a PPA repository into the /etc/apt/sources.list

To add a php PPA repository

```
$ sudo add-apt-repository ppa:PPA_REPOSITORY_NAME/PPA
$ sudo add-apt-repository ppa:ondrej/php
$ sudo apt update
```

Then install package

```
$ sudo apt install php
```

To remove the PPA repository

```
$ sudo add-apt-repository --remove ppa:PPA_REPO_NAME/PPA
$ sudo add-apt-repository --remove ppa:ondrej/php
```

113. rsync

rsync - a fast, versatile, remote (and local) file-copying tool

create a backup folder in Documents

```
$ mkdir -p /home/venus/Documents/backup
```

To backup .txt files to backup folder using rsync

```
$ rsync -v *.txt /home/venus/Documents/backup
```

To copy files from local to remote

```
$ rsync -av --progress *.txt
your_username@remote_server:/home/kaniyam/
```

To copy files from remote to local

```
$ rsync -av --progress kaniyam@ip:/home/kaniyam/*.txt .
```

114. scp

scp - OpenSSH secure file copy

To copy a file from local to remote server

```
$ scp *.txt kaniyam@remote_server:/home/kaniyam/
```

To copy a file from remote server to local

```
$ scp kaniyam@remote_server:/home/kaniyam/file.txt .
```

To copy a folder from local host to remote server recursively

```
$ scp -r example_folder kaniyam@remote_server:/home/kaniyam/
```

To copy a folder from remote server to localhost recursively

```
$ scp -r kaniyam@remote_server:/home/kaniyam/example_folder .
```

115. curl

curl - is a tool for transferring data from or to a server

To transfer a url

```
$ curl
<https://github.com/tkdhanasekar/linux_commands_examples/blob/mai
n/500%2BLinuxCommands%26examples.pdf>
```

To transfer a url and write output to a file

```
$ curl
https://github.com/tkdhanasekar/linux_commands_examples/blob/main
/500%2BLinuxCommands%26examples.pdf > curl_output.txt
```

To display a progress meter during use to indicate the transfer rate, amount of data transferred, time left, etc

```
$ curl -# -O
https://github.com/tkdhanasekar/linux_commands_examples/blob/main
/500%2BLinuxCommands%26examples.pdf
```

116. free

free - Display amount of free and used memory in the system

To Display system memory

```
$ free
```

To Display memory in Bytes/KB/MB/GB

```
$ free -b
```

```
$ free -k
```

```
$ free -m
```

```
$ free -g
```

To display system memory in human-readable format

```
$ free -h
```

To Refresh the output every 2 seconds

```
$ free -s 2
```

To write the output to a file

```
$ free -h > free.log
```

117. ifconfig

ifconfig - configure a network interface

To display all the interfaces available

```
$ sudo ifconfig -a
```

To display a short list

```
$ sudo ifconfig -s
```

To View network settings of wlp2s0

```
$ ifconfig wlp2s0
```

118. ip

ip - show / manipulate routing, network devices, interfaces and tunnels

To displays info about all network interfaces

```
$ sudo ip a
```

```
$ sudo ip -4 a
```

To show running interfaces

```
$ sudo ip link ls up
```

To check

```
$ sudo ip addr show
```

To check route table

```
$ sudo ip route show
```

119. netstat

netstat - Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

To list all ports:

```
$ netstat --all
```

To list all listening ports

```
$ netstat -l
```

To list listening TCP ports

```
$ netstat -at
```

TO list listening UDP ports

```
$ netstat -au
```

To List only listening TCP ports

```
$ netstat -lt
```

To List the statistics for all ports.

```
$ netstat -s
```

To print the netstat information continuously

```
$ netstat -c
```

To use grep with netstat

```
$ sudo netstat -tulpn | grep 80
```

```
$ sudo netstat -tulpn | grep https
```

120. ping

ping - send ICMP ECHO_REQUEST to network hosts

To check whether a remote host is up

```
$ ping google.com
```

```
$ ping ilugc.in
```

To limit the number of pings

```
$ ping -c 5 google.com
```

```
$ ping -c 10 ilugc.in
```

To print only summary statistics

```
$ ping -c 5 -q google.com
```

121. alias

alias - Creates aliases -- words that are replaced by a command string. Aliases expire with the current shell session unless defined in the shell's configuration file, e.g. ~/.bashrc.

```
$ alias c=clear
$ alias u=uptime
$ alias f=free -h
$ c
$ u
$ f
```

if shell is closed the alias will not work next session

To make it permanent

```
$ vim ~/.bashrc
alias c='clear'
alias u='uptime'
aliad f='free -h'
```

`:wq!` Save and exit

To make it to operational

```
$ source ~/.bashrc
```

122. unalias

unalias - Remove aliases

To remove the aliases in the current shell

```
$ unalias c
$ unalias u
$ unalias f
```

when the shell is closed and opened again the alias will work , to make the changes permanent we need to remove the alias in ~/.bashrc file

123. source

source - Execute commands from a file in the current shell

```
$ source ~/.bashrc
$ source /etc/profile
```

```
$ cat > example.txt
free -h
pwd
date
time
uptime
```

```
$ source example.txt
```

124. sh

sh - is a command language interpreter that executes commands read from a command line string, the standard input, or a specified file.

To Invoke the Bourne shell

```
$ sh
```

To run the bash script

```
$ sh example.sh
```

125. ssh

ssh - is a program for logging into a remote machine and for executing commands on a remote machine

To access remote server

```
$ ssh user@192.168.122.50
```

```
$ ssh user@my.server.in
```

To use a different port number for ssh connection

```
$ ssh user@my.server.in -p 2222
```

```
$ ssh user@192.168.122.50 -p 2222
```