

Unix

It is used in Telecom domain

Unix O.S.

Unix - Fedora, Ubuntu, Redhat, ...

like windows → Win XP, Vista, 7, 8, 10

My system OS was windows 10, on that there was a virtual terminal for unix as "Putty terminal".

* Unix → Server Side OS (CLI - command line interface like subset dos os)

* Linux → Personal PC/laptop (CLI + GUI like windows 10)

* Terminology in Unix Vs Windows

Files ↔ Files

Directories ↔ Folders

Virtual interface of unix → <https://cocalc.com/app/login>

* Create project → Home → create

New → Linux terminal window

bash (E)

Default directory - Home

* Rwx access →

File - - - - - rwxrwxrwx → " - "

Directory - d rwxrwxrwx → "d"

link - l rwxrwxrwx → "l"

binary files - b rwxrwxrwx → "b"

Type of file

User/owner

Groups

Others

rwx
meaning

{ r → read(view)

 w → write(edit)

 x → execute(compile, copy, Paste, delete)

* Command - press enter

1) ls

List files and/or directories from current working directory.

(-l)

ls -l

eg - velocity vel.txt Pune.txt
long listing / list with details.

eg - drwxr-xr-x 2 user user 2 Oct 31 6:30 vel
drwxr--xr-x 1 user user 17 Dec 10 5:14 velocity
dr-xrwxrwx 1 user user 7 Mar 21 08:14 Pune.txt

(-a)

ls -a

Include hidden files also in listing

(-r)

ls -r

List files in reverse order

2) cd

Change directory to home directory

cd directory name Change directory to concerned mentioned directory

cd ..

change directory to immediate parent directory! ← tracing steps *

3) Pwd

Present working directory.

4) mkdir

Used for directory creation

mkdir xyz A directory named xyz will be created

" " * Case-1 directory creation at sublevel (tree) hierarchy

" " " " mkdir dir1/dir2/dir3/dir4

" " " " * Case-2 directory creation at same level (branch)

" " " " mkdir dir1, dir2, dir3, dir4

* Case-3 If directory name which we want to create is already present use "-P"

mkdir -P dir1/dir2/dir3/dir4

* Case 4 If want to know action performed by "mkdir"

mkdir -PV dir1/dir2/dir3/dir4

* Cases creating one directory containing multiple directories

`mkdir mkdir -pv dir1/{dir2,dir3,dir4,dir5}`

type of command **Case 6**

5) ~~use~~ man manual for command.

`man command syntax`

eg - `man mkdir`

6) `rmdir` Remove empty directory (If a directory contains at least 1 file it will not be removed)

`rmdir directoryname`

7) `wc` word count

`wc filename` displays lwc (lines, words, chars)

eg - `wc velocity.txt`

2 7 27 velocity.txt
no.of lines → no.of words → no.of chars

8) `chmod` change mode

It is used to change max access for a file of owner, user and groups & ~~user~~ others

r - read (view)

w - write (edit)

x - execute (compile / copy / paste / edit)

for user/owner → 4

add access → "+"

groups → 8

remove access → "-"

others → 0

assign → "="

chmod ↑↑↑
person ↑ Task access

chmod g+w,o+wx velocity.txt

change mode add write

access for group

add write

& execute

file in which

access changes

others

& u is absent so no change of access level for user/owner

By using numbers

for any access $x/w/r$ number assigned = 1

for non access out of $x/w/r$ number assigned = 0

ex- $rwxrwxrwx$

eg- $\begin{array}{|c|c|c|c|c|c|} \hline & \text{owner} & \text{group} & \text{others} \\ \hline r & 1 & 0 & 0 \\ w & 1 & 0 & 0 \\ x & 1 & 0 & 0 \\ \hline \end{array}$

110100101

= suppose number is xyz

xyz
 $x \times 4 + y \times 2 + z = \text{decimal number}$

0 000

4 100... permission code

1 001

5 $101 \rightarrow eg \rightarrow 1 \times 4 + 0 \times 2 + 1 \times 1 = 5$

2 010

6 110

3 011

7 111

chmod 647 velocity.txt

$\begin{array}{ccc} 110 & 100 & 111 \\ \downarrow & \downarrow & \downarrow \\ rw- & r-- & rwx \end{array}$

$\begin{array}{ccc} & & \\ \downarrow & \downarrow & \downarrow \\ g & & \end{array}$

equivalent command $+t +w -x -r$

chmod 475

9) less / more Display file content file by file

page by page

less filename

eg- less velocity.txt

space

- jump to next page

b

- back/previous page

g

- 1st page

shift+g

- last page

q

- quite from page display

more filename

eg- more velocity.txt

less \rightarrow displays step by step file content on screen by spacebar

more \rightarrow displays file content page by page along with %file read,

10) head / tail →

head - Displays by default first 10 lines from file

tail - Displays by default last 10 lines from file

head filename

eg - head velocity.txt

o/p 1st 10 lines

tail filename

tail velocity.txt

o/p last 10 lines

(-n) **head - n filename**

o/p 1st n lines from file

eg - head -7 filename

o/p 1st 7 lines from file

head -22 filename

o/p 1st 22 lines from file

tail - n filename

last n lines from file

~~head tail -7 filename~~

o/p last 7 lines from file

tail -22 filename

o/p last 22 lines from file

11) **Vi Command** → File editor command

vi filename create filename and edit if file is previously not present or in case of previously available file open for edit.

Steps for vi:

1) vi filename.ext Type in terminal window (S1)

2) enter Press enter to open editor window

3) press i Insert mode in editor window

4) write content

5) Press "ESC" To quit insert mode in editor window

6) press "Shift + ;"

7) type wq (Write & quite)

8) we will be back in terminal window from editor window.

cursor motion (other than arrow key options) (01)

J - down

K - up

h - left

L - Right

W - Shift to next word

Delete option keys

X - Delete single char at which cursor is present

dd - Delete all content from line in which cursor is present (delete current line)

12) **touch** - to create file (it will create file but not open it for edit)

touch vs vi

create file but not open for editing	create file & open immediately for editing
--------------------------------------	--

touch filename

e.g. touch xyz.txt → It create xyz.txt blank file

(in root zone)

13) **grep** → Global regular expression pattern

eg - grep "pattern to be searched" filename

eg - grep "use" velocity.txt

↳ Highlight use word in velocity.txt

Cases - insensitive tip of

(-n)

① grep -n "use" velocity.txt

It includes all lines having use word.

(-v)

② grep -nv "use" velocity.txt

It shows all lines which not containing "use" word

(-i)

③ grep -ni "use" velocity.txt

It ignores case sensitivity & list all "use" words may be lower/upper/mix case

14) * pipes in unix It is used to combine/merge two commands in to perform in one action by separating with vertical line " | "

② - Syntax - command 1 | command 2

eg - cat velocity.txt | wc

* Output redirection → print file names of a directory to a file
 " > " & " >> "
 " > " - If file is not existing, it will create new file &
 If file is existing it will overwrite file

eg - ls > velocity.txt
 op - print file list names to velocity.txt file

" >> " - It will create file if not existing but if it exists then it will update existing file & not overwrite it (old data is also present)

eg - ls >> velocity.txt -
 op → print file list names to velocity.txt file

15) * Cat → Concatenate files & print on the standard output.

① Cat filename | Display content from file

② Cat filename1 filename2 filename3

Display content from all three files combinedly

③ Cat filename1 filename2 > filename3

Display content from first two files & save this combined content to filename 3

16) ~~cp~~ cp copy file (Creating replica)

CP sourcefile destinationfile

cp sourcelocation destinationlocation

eg - cp velocity.txt pune.txt

17) mv move file

mv sourcefile destinationfile

eg mv velocity.txt pune.txt

18) kill To stop process/program forcefully.

kill processid or kill pid

eg - kill 1

kill 728

19) top It will display all running processes with processid (pid).

Syntax - top

20) pidof To know pid of a particular process
(top gives all pid & process name list)

Syntax - pidof processname

OP - 22

21) Soft link & hard link

links works as pointer in Unix

It is used for backup of file

purpose Hardlink of file from softlink to

syntax In originalfile linkfile In originalfile linkfile.

Changes in originalfile Auto updated in Linkfile Auto updated in Linkfile
"inode". Same for originalfile linkfile different for originalfile linkfile

Original file delete Link file will be remains Link file also delete.

Hard Link - Copy / backup of original file

Syntax - In originalfile hardlinkedfile

✓ changes made in original file will be reflected in backup file automatically.

✓ "inode" number of original file & backup files are same

✓ If we delete original file, backup file will remain as it is.

Softlink - symbolic link / shortcut of a file

(-s) Syntax - In -s originalfile softlinkedfile

✓ Changes made in original file will be reflected in backup file

✓ "inode" number of original file & softlinked file are same

✓ If we delete original file, backup file will get deleted.

How to find inode

ls -li Displays file details with inode number

Eg- octsl.txt → oct.txt

145 -rwx--r-- 2 user user 61 Feb 27 15:02 oct.txt

145 -rwx--r-- 2 user user 61 Feb 27 15:02 octh1.txt

324 1wxwxwxwx 1 user user 7 Feb 27 15:01 octsl.txt

* **log in unix**

Directory location → /var/log

- Types-**
- 1) Success 4) Reboot
 - 2) Failed 5) Error msg
 - 3) Pending 6) Config logs

ls list files (Alphabetically)

ls -l long list files (in details)

ls -a list files with hidden files

ls -r list files in reverse

cd change directory to directly home directory

cd .. change directory to immediate parent

cd directoryname change directory to given child dir

pwd present working directory

mkdir dirname make directory

mkdir dir1/dir2/dir3 - if - exists (2)

mkdir -P {dir1, dir2, dir3}

man manual to know command purpose

rmdir remove directory

wc word count (lines - words - chars)

chmod change mode (rwx - access) of owner, group, other

less/more filename - Display file content step by step (more - %)

space

b short brief of woh

shift + g

q exit from text editor

head/tail filename first/last 10 lines of file (-n = n lines)

vi filename Edit command → file creation & open for editing

enter open editor

i insert mode of editor

ESC quite insert mode

Shift + j

Esc | Cntr V - write & quite

x - char at cursor delete

dd - cursor's line delete

touch filename Edit file → file creation but not open for editing

~~grep "Searchword" filename~~

grep Global regular expression pattern (all words in file)

- n *highlight all lines containing word in file*
- nv *" not containing "*
- ni *containing word (ignoring case) in file*

pipes (→) Command 1 | command 2

Output redirections

ls > filename > list file names to file & overwrite.
 >> append

cat concatenate files - Display content from many files

cp sourcefile destfile copy

mv move

kill processid kill process

top Display all running processes

pidof processname.

~~Backup~~

softlink Backup (inode diff, softlink = shortname)

hardlink Backup (inode same)

→ log in unix → /var/log
 commands of Unix

