

Unix & Shell programming language => B (predecessor of A)

1969 AT&T created Unix

→ Unix - 1973 - C language

→ portable & efficient

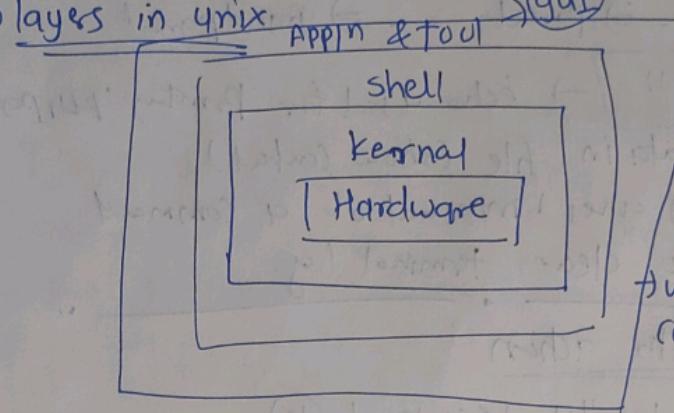
→ Linux is free version of Unix & Easy to install

(#) Unix system organisation

→ can work with multiple user system

→ computing power increased & cost decreased

(#) layers in Unix



kernel do task
on behalf of user

use shell to give
command to kernel
(Programm)

⇒ End user only works on Application & tool

(#) Coding Grammer

We have Linux so we can use VSCode

(#) Unix commands

ls ⇒ list the file & directory

cat ⇒ show/print content inside a file

whoami ⇒ our root OS

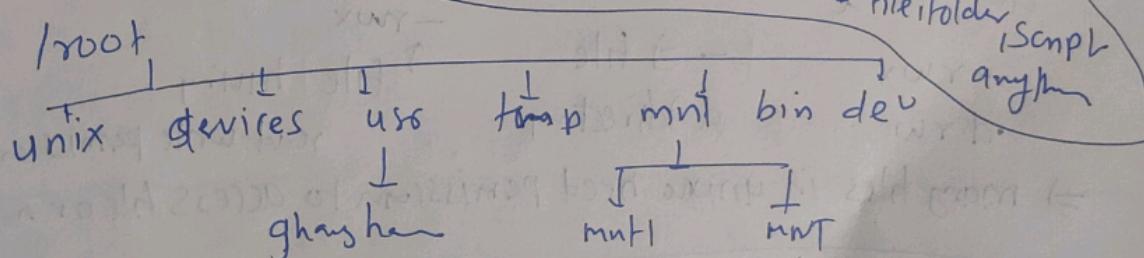
It is case sensitive command & filenames

(#) Unix file system

everything in Unix is file

↓ file, folder, script

anythin



such like that we have structure of Unix

⇒ tree command

Feature of unix file system

- Files can grow dynamically
- Files & directory has access permission
- All devices implemented as files in unix

Basic unix commands

- ① ls list file & folder , pwd ⇒ current working folder
(cd ⇒) change directory
 - .. ⇒ current
 - .. ⇒ parent
 - .. ⇒ grand parent

[echo "Hello print"] → echo used for printing purpose

- cat ⇒ print all data in file (show content)
- ⇒ man command ⇒ gives manual of a command
- clear ⇒ used to clear terminal log

② Commands with action

- ① ls -q ⇒ show detailed version of ls

options give after command

-d show dir hidden com

ls * .class ⇒ show .class file

tab
↓
autocomplete

- ② cp filepath targetpath

file permission & how to change it

do ls -l

-rwx

-d rw

- ⇒ file

d ⇒ directory

-rwx

↑ file having

read-write permission

⇒ many files in unix need permission to access file or not

[chmod used to change file permission]

Ex

chmod 0=rw testfile

others = write

→ give write permission to anyone

$\Rightarrow \text{chmod } 0 = \text{rw testfile}$
 $\downarrow \text{other} = \text{read and write}$

$\Rightarrow \text{chmod } ug = \text{rw testfile}$
 $\downarrow \text{User group}$

\Rightarrow Such like that we can change permissions
 \Rightarrow read write + execute

wildcard & file descriptors com

* \Rightarrow wild card

? \Rightarrow wildcard

ls * \Rightarrow show all file

ls *.txt \Rightarrow all txt file

* \Rightarrow all what

? \Rightarrow only one character

\Rightarrow use them like regex ? \Rightarrow used to search or change
so say you want to file ram.txt

ls ?q.*

find klpd.htm

ls t.html

ls h?m*

pwd \Rightarrow
cd

mkdir \Rightarrow new folder creation

\Rightarrow cp * ./main \Rightarrow copy all files to main folder

\Rightarrow rm -r folder \Rightarrow to delete a whole folder at once

Miscellaneous commands print

whoami, date \Rightarrow find date, cal \Rightarrow show calendar

\Rightarrow ps \Rightarrow list processes running on system

\Rightarrow kill \Rightarrow kill a process kill -9 \Rightarrow force kill

Inode & storage of file

UFS \Rightarrow Unix file system

Inode \Rightarrow Unix inode to store a file

\Rightarrow Each tiny file, owner, grp \Rightarrow has 1KB

chmod
can help us to handle permissions over file

Inode has pointers to files

⇒ See full Inode structure there is large size of Inode & it is also so much spread

max file size 1 indirect block \Rightarrow 256 kB

double -ll \rightarrow 64 MB

triple -ll \Rightarrow 16 GB

Such like that inode have indirect block which use & provide a large file sizes

④ Disk related Commands

⇒ df : Show info about disk also show usage

⇒ du : Show no. of subfolders in directory & no. of blocks used for file

⇒ mount & umount we can mount & unmount the hardware to harddrive

wc file.txt ⇒ it gives lines words character in file

⇒ sort ⇒ Sort data in file

⇒ file ⇒ it provide info about each file that is it a txt file or directory or pdf or else

⑤ Redirection highlights

You can give multiple command Semicolon Separation

⇒ ls & pwd ; cd (executed one by one)

You can redirect input of command

ls > file.txt

ls >> file.txt

write alpha file

append alpha file

> link

>> append

Such like we can give input using <

(cat < ls.txt) not used commonly

④ Piping/highres

You can redirect output to another program

Ex:

date | wc

so output of [date to wc]

=> cat xyz.txt | wc

Input that is piping

| => pipe piping

< , > => redirection

=> head top (10)

=> tail bottom (10)

=> grep => find

so ps -ef | grep 'apache'

from process we find apache

grep used
to shortlist

=> ps -ef | grep 'network' → filter

grep mostly used in pipe commands

⑤ Processes & cron jobs

ps -ef => command you

~~ps -ef~~ =>

PS => show process

kill -kill process meth

kill -9 PID

ls & => this give process id

ls &.class &

=> At last line you can see
PID of process

Cron job => scheduled job

=> Crontab

this will schedule a job which run at that time

min > 05 => hours

every day 5) →
at time 5 hours
in week
min hour in day

④ Vi editor (also nano editor is also there)

Basic :q => quit :w => write
:q! : quit & exit :wq => write & exit

a => append

⇒ Vi has Text Mode & Command Mode

esc + : => and you can write commands.

(!) is → ! ~~also used to write~~
↑ used to write commands