# Linux

12/7/24

operating system:— it is a collection of 100 and 1000 of application and programs that collective run to gether that user can interactive with the Hardware.

Types of software

2) freeware

3) open source

open source:— it's a kind of software which is use by the users were the cost of the software is 0.(zero) but source code can be used by anyone to customize and publiesh as the hare.

linux → open source

Types of linux:—

Types of linux os

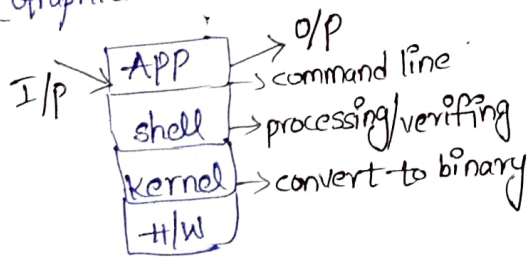
RHEL, elementary os, opensuse, Linux mint, centos, Arch linux, Debian, fedora, ubuntu, Gentoo, kali linux

15/7/24

## layers of operating system

it is a 2 layers of operating system.

GUI
(Graphical user interface)

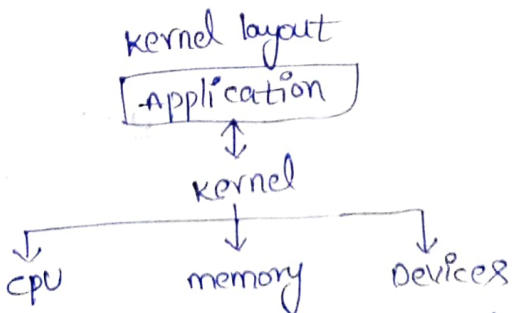



CLI
(command line interface)


Hardware layer:—

it constant of all the physical components. that is require to run a machine, (cpu, ram, Hardick).

Kernel:—




The main layer between the os and underlying computer hardware.

* A computer program at the core of a computer's operating system.

→ It consistant of ~~firewall~~ which converts high level language to low level language and vice versa shaw the user interact with the hardware. it's refere to core of the operating system.

shell :— it takes the input from the user and validation (check for instructions and also for the user permission to executed).

Application layer :—

it is present only GUI which converts graphical jesters (actions into command line instructions and using to the machine).

unix (1969) D.R/K.T (uniplexed information computing system).

- sun → solaries
- IBM → AIX
- HP → HP-UX
- Linux (1991)
  - → freeBSD
  - → RHEL
  - → centos
  - → ventix
  - → kali
  - → Backstack
  - → Sure/Android
  - → minix

* unix is a multi-user operating system.

main use of unix :— editing text
writing, compiling
controlling, programs,
processing.

3 parts

* The kernel, the shell, and user commands are applications.

16/7/24

## File system hierarchy

192.168.29.29

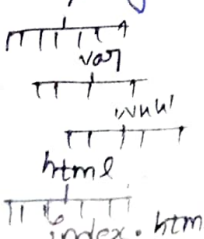
→ All files and directories are organize inverts single rooted inveted tree structure.

→ All file systems beings with root which is represented by "/" (root directries).

→ present and sub directries are separated using "/".

/var/www/html/index.html

- / → root
- / → Seperaters



* The named of files and directories are all case sensitive.

→ names of files of directories kindly the maximum of 256 characters in length. to check use alphabets, numbers and special characters. it expect of /

Imporpdent directories of linux

1) / bin
2) / sbin
3) / home
4) / etc
5) / dev
6) / boot
7) / tmp
8) / run
9) / user
10) / var

/proc
/ media
/ mnt
/ opt
} optional

17/7/24 basic commands

/bin :- it contains all the commands that can be executed by any user (Normal user, root user etc...,)

/sbin :- The contains all the commands that can be excuted only by root user by default.

/home :- it contains all the users default home directories

Note :- home directory is a personal directy allocated to every user.

/etc :- it contains all configuration files.

Note :- configuration files contains all the setting require to run the service. hence when ever we head of configuration file we restart the services.

/dev :- it contains all the hardware devices information.

/boot :- it contains Boot loader and kernel information. (GRUB 2)

/temp :- it contains are tempary files of the information / machine.

temp is special directory where any data that's not access. for last 30 days we will get automatically removed.

/run :-
↳ it contains are run time data optimis of the machine.

/var :- it contains all the variable data this has automatidy saved without user intergerence.

/usr :- it contains all the read only information of the install packages.

/proc :- classes created information.
* processes related to the information.
/media :- that's access point to all external storage devices attack to the to machine.
/mnt :- To the access point for network shared data.
/opt :- it contains all the third party packages install of the machine.

22/7/24
command :-

command is a instruction or set of instruction that has to be executed and also provide valid output.

different types of command

| # cmd | option | argument |
|---|---|---|
| what? | How? | where? |
| # ls | -a | /etc |

Note :- To list all files and directory including hiden and present working directory.

* The command is mandotry the option and arrgument on the option can be change for the required.

Note :- when arrgument is not provided were present working directy (current directy) we present on the directy.

* if option is not provided then command will executed normally.

[root@localhost ~]# → root user
$ → normal user

root → current user
localhost → local machine name
~ → present working directy

~ → home directy

# date → display current date time
cal → To display to current month calendar
# cal 2025 : Ex
[root@localhost ~]# cal 2025
# uname → To display the operating system name
# uname -r → To display the version of the kernel.
# uname -v → To display the use of the kernel.
# uname -n → To display the machines name
# uname -m → To display the machines architecture.
# uname -o → To get the os of the machine.
# uname -p → To get the processer architecture.
# uname -a → To get all the machine and os machine.
# whoami → To display all the current login users name.
# who → To display all the current login users name.
physically and remotely.
# history → To display the history of commands executed by a particular user. history is user specific.

23/7/24
listing commands :- (LS)
#ls space is to list all files and directly under the given directry.
# ls → To list out files and directy under the current directry.
# ls -t → The list files and directryes based on time (start) stamp. (new to old)
Note:- time stamp difines the last modify date and time of a file.
# ls -tr → To list out all files and directries based on time stamp (old to new).
# ls -i → To list out all files and directries with inode numbers (index number).
Note:- files as directry hoods names of a users referes were as inode numbers of a system referes.

# ls -a → To list out all files and directries including hidden.

# ls -F → To list out all files and directries and difference between them.

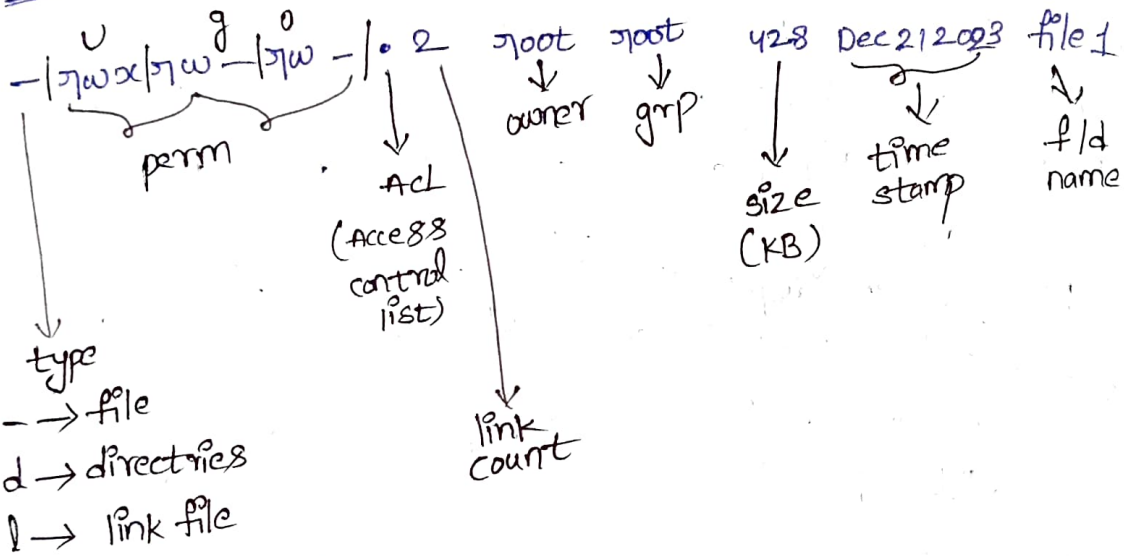
~ → Normal file

~ ★ → executable file ~ @ → link file

~ ~ / → directries

# ls -l → To list files and directries with complete details. (long list)

Ex:-




## change directry

# cd <d/N> → To change the cell access from present working directry to given directry.
directry name

# cd .. → To go to the parent directry of the given directry.

# cd → To go to the home directry are the current lockdin (login) user.

24/7/2024

unit-1

# File & directory management

3 ways of creating files

1) touch
2) cat
3) vim

touch:- touch files space new file name how many at bit file create a empty file (of root)

# touch < New f/N >

cat:- it is used only visit. it create a new file and it allowed the add data immediately.

# cat > <new f/N>

ctr+d → save & exit

# cat < existing f/N

# cat > < existing f/N

→ overwrite

ctr+d

Note:- deleting exciting data to ladding new data is called overwriting.

# cat >> <existing f/N

append

# ctr+d

To add data to existing data to the file.

Note:- the major dis advantage of cat is data can't be edited, data can either be overwriter or append. that use by cat.

vim:-

# vim < new f/N >

# vim < existing f/N >

esc+i → insert mode

esc+:+wq! → to save & exit

esc+:+q! → to exit w/o file

Vim is basically used as file editer.

## creating directerity

# mkdir < New d/N > create directy

# mkdir < dir 1 > < dir 2 > < dir 3 >

→ create multipule directy cymantenlly

king
↓
queen
↓
prince
↓
slave

# mkdir king
# cd king
# mkdir queen
# cd queen
# mkdir -p king/queen/prince/slave

# mkdir -p :- is used for creating multiple directry (or) creating simultaneously.

ex:- mkdir -p king/queen/prince/slave

ls king → To check the directory

26/7/24

## copying files and directeries

s → source file  d → destination file

→ # cp <s/f> <d/f>

To copying content from source file to destinator file. if the destination file doesn't exist it create new file then copyies the data. if the destination file already exists then it overwrites the data.

# cp -r <s/d> <d/d>

To coping content from source directry to destination directory recursively.

## moving file and directeries

# mv <source> <destination>

moving file as directory from source to destination.

# mv <oldname> < new name >

Renaming file and directeries

# mv <old name> < new name >

Renaming file as directory to old the new name.

Deleting files and directories

-f - filename

# rm <f/N>
To delete the file. it also ask permission to delete the file. it ask permission before deleting files.

# rm -f <f/N>
To delite file forcelly. it directly delete the file.

# rm -r <d/N>
To delete directry. it ask permission before deleting the directory.

# rm -rf <d/N>
To delete directory forcefully as recursively it directly delete the directry.

27/7/2024

## users and Group management

users:- Any body who has valid identials to login into the machine and access to services and application is refered as users.

→ user are mainly required for authentication defines who should be able login and who should be not.

→ users are also required for authentication.

✓ gid:-(29/7/2024)
It defines the primary group of a user.

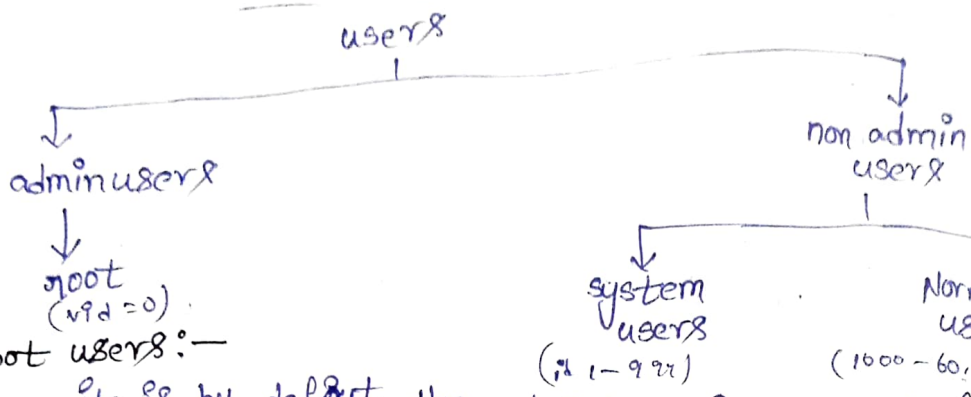
Note:- Every user can be root of one and only primary group but can be poot of upto is optional secondary groups.

✓ gecos:-
It is optional field of users where it describes the users. it is optional field where the user can define anything for this reference.

→ users are require to define the permission (ownerships) of filles we directries.

→ Every processes running on the machini must and should have owners which is defined by users.

# Types of users




root users :—
it is by default the administrative account of the machine which has complete access over the machine with impetive of the permission.

Normal users :—
these are the users account created by the root users so that other user login to the machine and access to the machine.

system users :—
These are the user account created by the operating system to take the ownership of process. these user cannot login in the machine access to machine. They do not have valid user names and password.

28/7/2024

## user id

It is unique nomber is allocated to every user for system reference.

→ They uid for root user is allocated '0'
→ uid for system user is '1-999'
→ uid for normal user is '1000 - 60,000'

Imp:— All the users information is stored under "/etc/password"

## fields of /etc/password

V/N : pwd : uid : gid : gecos : homedir : loginshell

V/N :
it is name allocated to a user for users systems.

pwd : [x]
it is defined as the password is set for the user.

vid: it is unique number allocated for every user for system references.

(gid: gecos) mandatory page fourth

homedir:
it defines the home directory allocated to a particular user. ex: /home/<user name>

loginshell:
it define a shell allocated to particular user.

⇒ #cat
To create new user home. #useradd <New V/N>

⇒ # password
To set the password activate user.
# password <New V/N>

30/7/2024 / 31/7/2024

→ Every user must and should have 6 mandatory fields namely username, uid, password, primary group, homedirectory, loginshell

→ whenever the delayout values are not defined the system automatically takes some delayout values.

Default values

uid → password uid+1

(primary gap) gid → <U/N>

homedir → /home/<V/N>

loginshell → /bin/bash

To create user with specific values

→ #userold -V <New V/d> -g <pri gap> -G <sec gap> -d <home dir> -c <geces coment> -s <login shell> <New/VN>

→ # password <U/N> → To set the password of new user.

Note:- whenever the userid is define the uid should always bean unique. the uid is above 1000.

→ #useradd -u 2000 -g solves -G hr -d/scholaggic -c testuser -s/sbin/nologin semreen.

Note:- example of creating user specific values.

→ # password sumreen → password set
→ # id sumreen → To check values of prigrp and sup grp.
# cat/etc/password
to check context of etc/password all user check

user modification :—

# usermod -u <New uid> -g <pri grp> -G <sec grp> -d <home dir> -c <gecos/ comment> -s <loginshell> <existing V/N>

Note:- usermod is command to modify the existing user.

# password <existing V/N> → To changes the password of existing user.

user deleting

→ # userdel <existing user> → To delete only user.
→ # userdel -r <existing user> → To delete all files and data of users.

Group management let/group

Group is collection of multiple simlar type of users. it is mainly used to managed multiple user simultenously.

creating group with default value

→ # groupadd <New g/N>

creating group with specific values

→ # groupadd -g <New gid> <New g/N>

modifining group

→ # groupmod -g <New gid> <existing g/N>

Deleting Group

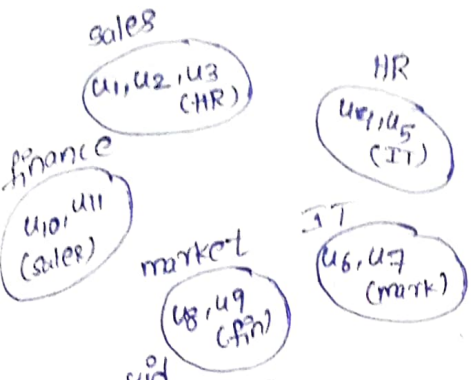
→ # groupdel <existing g/N>

A group can be deleted if and only it is empty.

1/8/2024

Requirement :—

1) first create a sales.




Sales $\xrightarrow{uid}$ 2000
HR $\longrightarrow$ 3000
IT $\longrightarrow$ 4000
market $\longrightarrow$ 5000
finance $\longrightarrow$ 6000

u1, u2, u3 $\xrightarrow{uid}$ 2001 – 2999
u4, u5 $\longrightarrow$ 3001 – 3999
u6, u7 $\longrightarrow$ 4001 – 4999
u8, u9 $\longrightarrow$ 5001 – 5999
u10, u11 $\longrightarrow$ 6001 – 6999

ux $\xrightarrow{\text{home dir}}$ /home/logic ux
ux $\xrightarrow{gecos}$ student x
u4, u6, u8 $\xrightarrow{shell}$ /sbin/nologin
u1 – u11 $\xrightarrow{pwd}$ redhat

2/8/2024

## permissions

permission are always apply only are files and directries that define the level access for a partiular group of user.

* 3 types of permissions.
  1) read (r)
  2) write (w)
  3) execute (x)

* In linux software called package.
* in linux folders called directries.

| | files | directries |
|---|---|---|
| r | to view/read the content of a file (data) | to list the contents of a directrie(files & sub dir) |
| w | to modify/edit the content of a file (data) | To create/delete(copy)/move/rename the content of directrie (files & sub directriex). |
| x | to install a package/ to execute a script | To execute the path of a directrie (ie cd <dir>). |

Note:- if have to delete the file & sub directry then parent directries should have write and execute permission if we want to delete content of a file the file should have a permission.

data/test → minimum permission
-wx ---

→ Thamb rule rwx

Note:- The permission should always be in order of r w x.
-to read write and execute (rwx).
were ever no permission replace with (-)

data/test -wx ---

| | |
|---|---|
| -wx | r--x |
| r-x | r-x |
| ✓ -wx | -wx |
| ✓ -wx | r-x |
| r-x | -wx |
| r-x | -w- |
| ✓ -wx | -w- |
| ✓ rwx | --- |
| r-- | -wx |
| ✓ rwx | rw- |
| ✓ rwx | -w- |

| | |
|---|---|
| ✓ rw- | --x |
| r-x | --x |
| --x | rwx |
| ✓ -wx | r-- |
| ✓ rwx | -wx |
| ✓ rwx | r-- |
| r-x | --x |
| --x | -w- |
| ✓ rw- | r-- |

* The permission identifications

rwx
rwx
r-x
r--
-w-
--x
---

5/8/2024

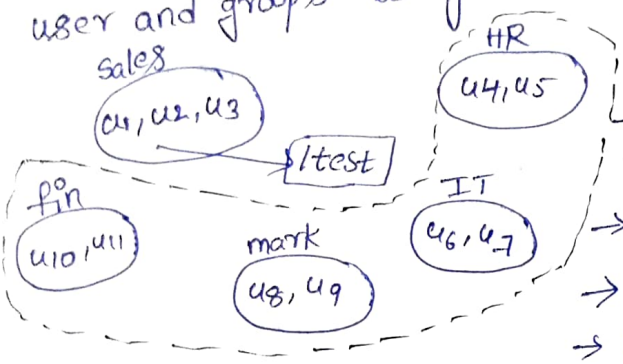
Permission can be defined as three major categories

owner(u), group(g) and others(o)

owner:-
By default the creator of file and directy is the owner(u).

group:- (g)
By default the primary group of the owner is refered to as the group of file as directy.

other(o):-
A pair from owner and group remebers all the user and groups belongs to other categories.




→ u1 is the owner
→ sales is group and
→ others is HR, IT, market, fin and their sub files and directy.

## changing permissions

chmod any apply for three categeries two is u, g, o.

Ex:- /test  u g o
r | r | rwx → existing file
rwx | rw- | r-- | → New file
(u)7 g(6) o(4)

## symbolic / relative

cmd → # chmod u+wx, g+w g-x, o-wx /test/

## Numeric / Absolute

chmod → use-ful changing permission of file

cmd → # chmod

r - $2^2$ → 4
w - $2^1$ → 2
0 - $2^0$ → 1

```
cat >/test
chown u1:sale/test
ls-l /test
chmod 764/test
```

```
ls-l /test
su u1
su u2
su u7
chmod 760/test
```

u g o
7 6 0 /test

rwx → 7 --x → 1
--- → 0
rw- → 6
r-x → 5
r-- → 4
-wx → 3
-w- → 2

## changing ownership

cmd = to change owner and group permissions one time only

→ # chown < new owner > < f/d name > chown → To change owner ship of file.

→ # chgrp < new gap owner > < f/d name > chgrop → To change group of file.

To change both owner and group of getter

→ # chown < new owner > : < new gap owner > < f/d name >

```
su-u1
cd/
cat/test
cat >>/test
logout
```

```
su-u1
cd/
cat/test
./test
exist
```

```
su-u2
cat/test
cat>>/test
./test
exist
```

```
su-u7
cat/test
cat >> test
./test
chmod 760 /test
su 7
cat/test
```

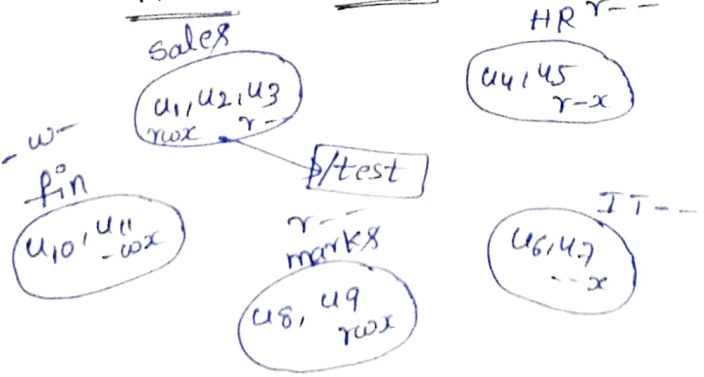
# su < f/name > → To switch to users.

# cd → To go root directry

# ./test → Is used for directries of file.

6/8/24

## ACL → Access control list



→ The major drawback of chmod is permission can be define only for three catogeries. i.e, u, g, o owner, group, other.

→ using ACL we can chose individual user & gap & give them permission bassed on specific user and grap without distributing other user and group.

```
# chmod  764 /test
          u g o
#setfacl -m u:u3:r-- /test  → To change user permission.
#setfacl -m u:us:r-x /test
#setfacl -m g:IT:--- /test  → To change group permission.
#setfacl -m u:u7:--x /test
#setfacl -m u:u9:rwx /test
#setfacl -m g:fin:-w- /test
#setfacl -m u:all:-wx /test
```

#getfacl <fld name> → To check all ACL (append) Applied user and group permission.

```
ex: #setfacl -x u:u9 /test
ex #setfacl -x g:fin /test
```

→ #setfacl -x is used for delete, remove particular user and group permission.

→ #setfacl -b is used for delete/remove all ACL permission of user & group.
ex: #setfacl -b /test.

ACL # ls -l

. → ACL is not applied

+ → ACL is applied

ex: -rwxrwxr--+ 1 u1 sale 36 Aug 5 21:34 test } ls-l
                ↑ 1st Applied ACL

Ex: -rwxrwxr--+ 1 u1 sale 36 Aug 5 21:34 test }
                ↳ Applied ACL

7/8/2024

## Imp umask / (user mask)

umask is mainly used to define the default permission that has to been apply on to the file and directry when created.

> Default permission = full permission − user mask
> DP = FP − umask

FP of directry → 777 (rwx/rwx/rwx) (owner group other) (red and excecute one of the other group of execution).

FP of file → 666 (rw-/rw-/rw-) (red write one of the other group of redwrite).

→ umask value of root user (022)

→ umask value of normal user (002)

### Default user of mask

root user (022)

| dir | file |
|---|---|
| 777 | 666 |
| -022 | -022 |
| 755 | 644 |
| rwx/r-x/r-x | rw-/r--/r-- |
| u g o | u g o |

Normal user (002)

| dir | file |
|---|---|
| 777 | 666 |
| -002 | -002 |
| 775 | 664 |
| rwx/rwx/r-x | rw-/rw-/r-- |
| u g o | u g o |

umask → To view the existing umask value.

umask -value → To change the umask of the given value.

| | |
|---|---|
| 666 | 777 |
| -024 | -024 |
| 642 | 753 |
| rw-/r--/-w- | rwx/r-x/-wx |