Help → ① command --help :- will show all avail. options
② man command :- displays manual page of each command

SSH → remote connection to any system

Syntax :- ssh [username]@[Machine_name/Hostname/IPaddress]
Eg :- ssh student@Service

Case 1 Simple Remote Connection (with ~~Out Without~~ Passphrase)

steps ① Generate Key Pairs (Public & Private)
→ ssh-keygen (id_rsa & id_rsa.pub)
Note :- New Keys can be found in .ssh folder
you can Provide Passphrase in this step

② Copy the Public Key to remote server
→ ssh-copy-id [username]@[Machine_name]

③ Connect to remote server
→ ssh [Username]@[Machine_name]

Case 2 Remote Connection without Password/Passphrase
{Runtime / Temporary method} (Or) Connection with agent

steps ① Same step as of Case 1's first step

② deploy an Agent
→ eval $(ssh-agent)

③ assign Agent with your Passphrase with your Private key
→ ssh-add .ssh/[Private_Key_name]
here you will then add your Passphrase

④ Connect to remote server

↓
Note :- Agent stores the Passphrase in runtime & hence after deleting your terminal, Agent will release its memory

Case 3] Remote Connection to server without Password or Passphrase Permanently

steps ① Switch to root user in remote server

② Navigate to etc/sshd

③ Edit the file such that you explicitly allow the connection from our server

---

Non Interactive Key (or) Non default Key

step ① ssh-keygen -f .ssh/mykey

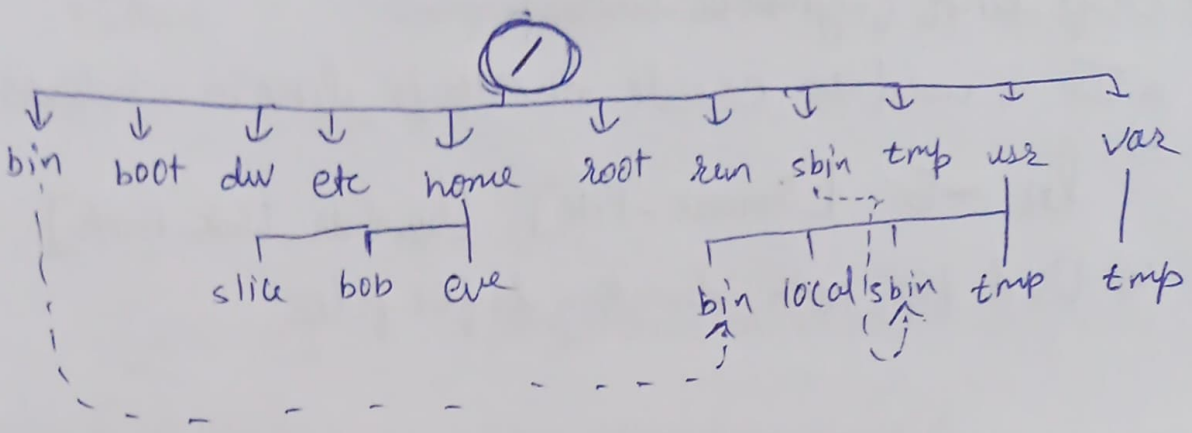
here we specify the file location by force (-f) explicitly

② Copy to remote user

ssh-copy-id -i .ssh/mykey.Pub [username]@[machine]

↑ To ignore the default key

Chap - 2



/boot :- Files to start boot process

/dev :- special device files that the system uses to access hardware

/etc :- System - specific configuration file (IMP)

/home :- Home directory, where regular user store their data & config file

/root :- Home directory for the administrative superuser. root.

/run :- runtime data for processes that started since last boot.

/tmp :- A world - writable space for temporary files

/usr :- installed software, shared libraries, files & read only data

/var :- files that dynamically changes, eg: database, cache dir etc

/usr/bin :- A dir that contains regular commands & utilities

---

sudo -i ⇒ Change user from root ⇒ student

Links (ln)

- Hard Link
- Symbolic link (Soft)

⇒ Soft link (Symbolic link)

↳ It is used to create shortcuts like in windows

↳ `ln -s [Source-File] [Symbolic_Link_Path]`

⇒ Used for both directory & for files

⇒ Hard link

⇒ Its like a connected copy of a file

⇒ Advantage :- we can still access the file even if the original file is lost unlike soft link

⇒ Disadvantage :- Supports only for files

⇒ To identify hardlink we check "inode number"

↳ `ln [source-file] [Symbolic-link-Path]`

Two files or directories are said to be same if their inode numbers are same

To check inode number ⇒ `ls -i`

Pattern	Example/Explanation
*	echo "v*"
?	echo "ved???"
[abc...]	Any character in box
[!abc]	None characters from box
[^abc]	" " " "
[[:alpha:]]	Any alphabetic character
[[:lower:]]	Any lowercase character
[[:upper:]]	Any uppercase character
[[:alnum:]]	Any alphabetic or digit
[[:punct:]]	Any printable character that is not a space or alphanumeric
[[:space:]]	Any white space character.

CH-3

Users → Superuser (root)
→ Regular user
→ System user
} Unique UID (User Identification Num)

By default

root → UID = 0

System user ⇒ UID = 1 - 999

Regular User ⇒ 1000 - 60K

* /etc/Passwd ⇒ Basic information of all users

→ Eg: root : x : 0 : 0 : /root
{User's name} {Password Pointer} {UID} {gid} {Home dir}
: /bin/bash
{Assigned shell to user}

↳ Useradd [user's name] ⇒ Creates a new user

↳ Passwd [Username] ⇒ Assigns Password to the user

↳ Usermod ⇒ Used to change any Parameter of already created user
(usermod --help)

↳ Group → Primary group (default)
→ Secondary group

* /etc/group ⇒ all information related to groups
(groupmod --help)

⇒ use id or id [username] to see user's basic details

⇒ ls -l [filename] ⇒ To view owner of the file

lsattr ls -ld [Directory] ⇒ To view the owner of the directory

---

⇒ su - [username] :- To switch user with new user's env/filesystem

su [username] :- To switch user without new user's filesystem / env.

⇒ sudo ⇒ ~~Switch~~ super user do

⇒ sudo -i ⇒ switch to root user

⇒ /etc/sudoers ⇒ config file for sudo command

eg:- ① To enable full sudo access for [username]

Syntax :- [username] All = (ALL) ALL

User has access to all machines = (all members) all commands

② To give full sudo access for [group Name]

Syntax :- %[group Name] ALL = (ALL) ALL

all machines = (all members) all commands

③ To setup Passwordless execution of sudo command by any user

Syntax :- NOPASSWD : ALL

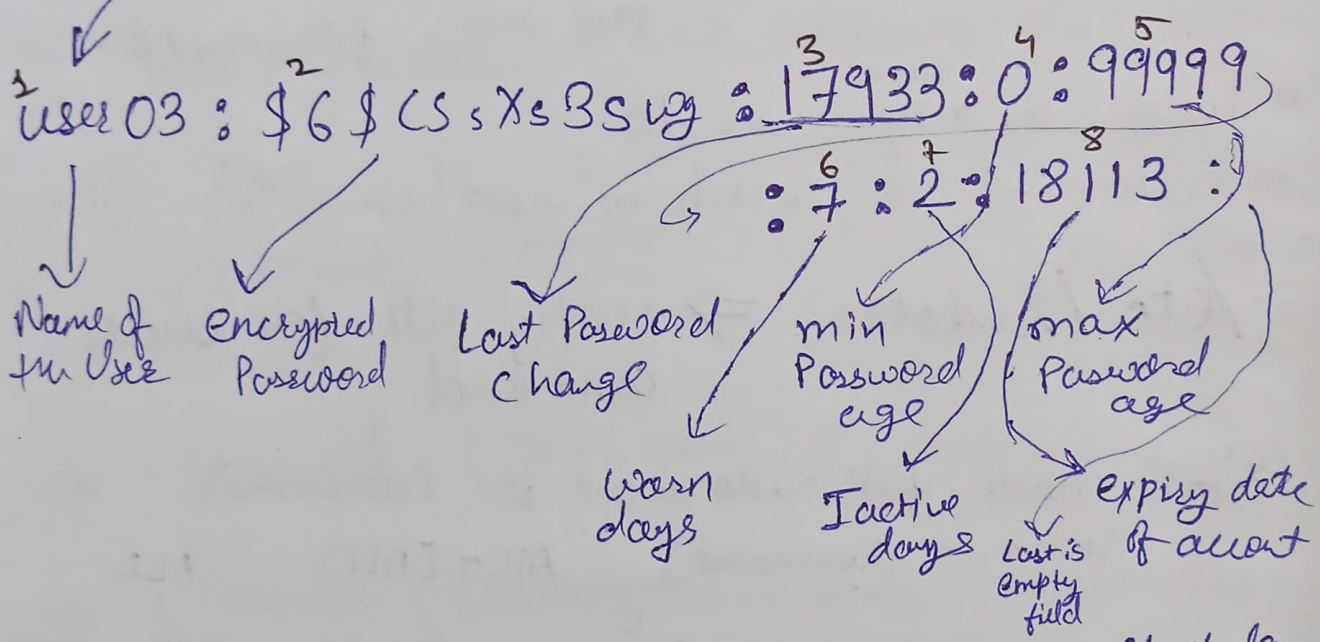
⇒ userdel [username] ⇒ deletes the user

⇒ userdel -r [username] ⇒ deletes the user with its home directory

⇒ To add a [user] [userName] into a [groupName]

usermod -aG [groupName] [UserName]

★ etc/shadow ⇒ all user's Passwords are stored here



③ Last Password change (-d) :- show the days since epoch day i.e 1st Jan 1970

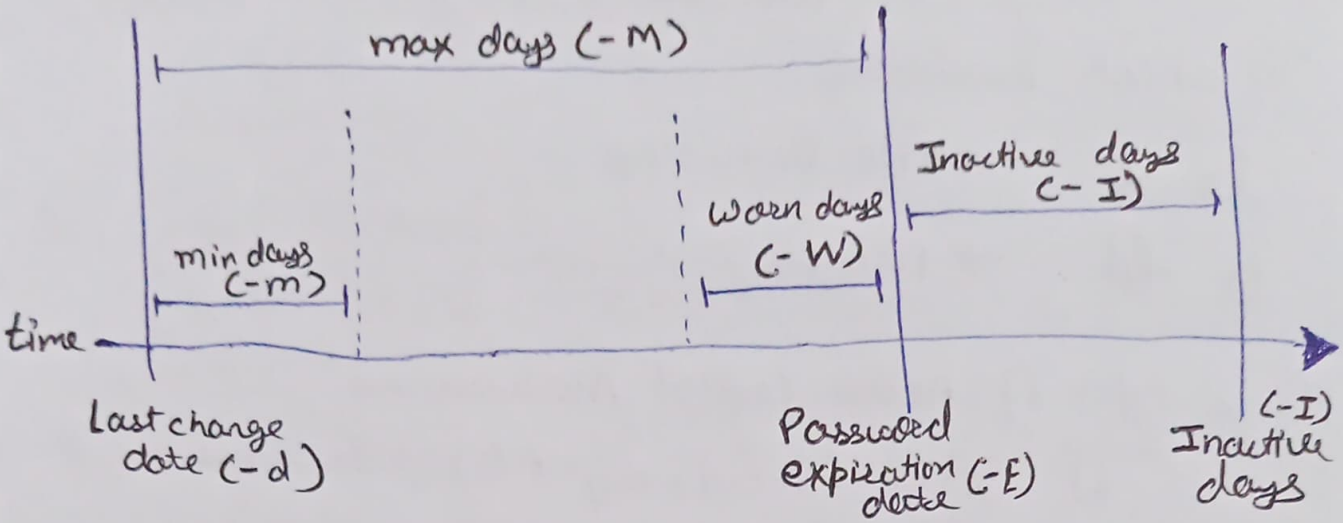
④ Min Password age (-m) : If 0 then a user can change Pass anytime

⑤ Max Password age (-M) : A user's Pass is valid for Provided no. of days

⑥ Warn days (-W) :- A user recieves warning before specified no. of days before the expiration date

⑦ Inactive days (-I) :- If 2, then Password will work 2 days more even if it is expired

⑧ Expiry date (-E) :- The day when the account expires it can be days or in YYYY-MM-DD formed



↳ Chage command is used to change Password aging Policies of a user

↳ eg :- chage -m 0 -M 90 [username]

⇒ To Prevent a user from interactive shell

Syntax :- usermod -s /sbin/nologin [username]

⇒ Force the [Username] to change the Password on first login

chage -d 0 [username]

⋇ To change Password Policy for users that supposed to be created in future

↳ For that we have to configure Password Policy in config file :- /etc/login.defs

(login definitions configuration file)