

# COSA

## Introduction to UNIX

- OS - it is an interface between user & hardware
  - it does memory & hardware management.

UNIX - 1969 (dennis Ritchie)

LINUX - 1990-91 (Linus Torvalds)

WINDOWS - 1980-81

→ LINUX - Mostly used in world.

eg - Mobile (Android based on LINUX)

### UNIX :

→ it is portable.

→ it can install to any hardware

Multiuser - more than 1 user accessing sys  
at same time parallelly

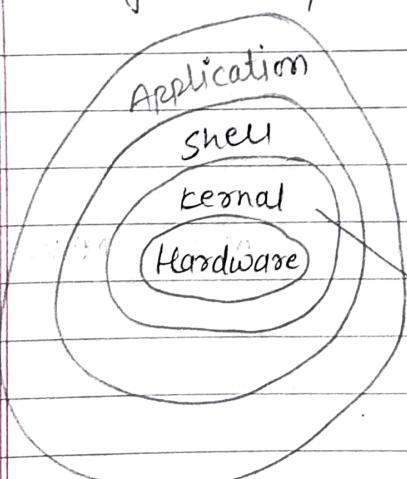
### Multitasking -

→ Multiprocessing OS

(Unix can support more than 1 CPU)

→ Written in 'C' language.

### 3 Major Component :-



→ Kernel - core of UNIX OS

→ Shell - bash, zsh, sh

→ Application

eg - firefox, open office  
gedit etc.

→ core of unix b/w soft & hard  
which control execution of OS

## # UNIX Distribution

- Commercial
  - [ it is paid version]
  - I Solaris (sun Microsystem)  
[oracle]
  - 2. AIX - only for IBM software
  - 3. HP-UX - only for HP system  
↳ user
  - 4. Mac OS -
  - 5. Tru64 - it is also for HP  
(supercomputer of HP)
- Free
  - [ free version.]
  - I Free BSD,
  - 2. Net BSD.
  - 3. Open BSD

- # LINUX
- it is like OS (Windows)
  - it supports networking [32bit / 64bit Unix like OS]
  - UNIX tools & Compiler are also in Linux.
    - ↳ grep, awk
    - ↳ C, C++, Java etc.
  - CLI - Command line interface
  - GUI - Graphical user interface
  - it has a complete development environment include compiler, tool & scripting
  - Multiuser, multitasking & multiprocessor
  - UNIX - CLI
  - LINUX - GUI
  - it can coexist with other OS in same hardware

Windows 11	LINUX
500 GB	500 GB

→ 1 TB

→ it supports multiple hardware & it is a open source software OS.



→ symbol of open source.

## # Open Source (Source code is available with you)

↳ free means related to freedom

→ which means you can download, study, change & modifying & share.

→ Source Code - open source soft<sup>ware</sup> gives it.

→ License - every software have some license key.

↳ it can be redistributed under some license.

## → Types of license :-

- GPL - GNU General Public License

[UBUNTU] 20 Oct 2013

- BSD - Berkley Software distribution.

## History of LINUX :-

→ 1985 - Minix Project

→ 1991 - LINUX (the first version of) - 1.0

→ 1992 - Soft Landing LINUX System

→ 1993 - Debian, Slackware & RedHAT

→ 1994 - LINUX kernel 1.0

→ 1996 - GNOME desktop environment

→ 2001 - UBUNTU

→ 2003 - LINUX kernel 2.6

→ 2007 - Android

## LINUX Distribution ( 200 distribution of LINUX)

- 1) Android (developed by Google)
- 2) Redhat (it is stable, reliable, enterprise)
  - ↳ fedora, Redhat
  - ↳ it is testing version for building of main redhat version
  - ↳ developed by Redhat
- 3) Open SUSE - suse linux GMBH
  - ↳ it is developer & system administrator
- 4) Debian - it is widely used, powerful OS
  - ↳ UBUNTU
  - ↳ KALI Linux - pen-testing, ethical hacking
- 5) Linux Mint - very easy to use
- 6) Chromium OS - developed by Google

## # LINUX & UNIX

↳ developed by vendor [IBM, HP]

IRIS Linux - developed by Silicon Graphics

LINUX - support all types of files & multiple hardware, multiuser, multi-tasking

- Redhat - founded in 1993
- Redhat Linux - it is popular & open source
- RHEL - redhat enterprise

## Difference bet LINUX & UNIX

LINUX	UNIX
→ it is freely downloaded & distributed	→ different cost structure a/c to vendors.
→ it is GUI Based	→ it is CLI Based.
→ it can co-exist with other OS in same machine	→ it can co-exist with other machine.
→ it is used by home user to developers	→ it is mainly for servers, workstation & mainframe.
→ kernel developed & maintained by community eg- centos, UBUNTU, debian etc.	→ Mostly maintained by commercial vendors. eg- RedHAT, AIXS etc.
→ Linux support NTFS, PAT, FAT32, EXT2/3/4, IPS, XFS, Btrfs.	→ It support file system ls, gls,
→ it support all file system	→ does not support all file
→ it able to access other OS files from same machine	→ it is not able to access both OS.
→ it support multiuser, multihardware like mobile, phone etc.	→ it support workstation & server only.

### # Redhat

- it is MNC software company.
- founded in 1993
- for enterprise version redhat has major market share
- RHEL (Redhat Enterprise Linux OS)

## # Redhat LINUX:-

- it is popular in 1993 but stopped in 2003
- it is community driven distribution.

## # Redhat Enterprise Linux [RHEL]

- After Redhat Linux stopped it comes in market.
- it is commercial supported software
- used in businesses, organization, and govt. agencies.
- it charge for support but OS is free.
- it is enterprise grade because, it is stable, secure & long term supported.
- it is commercially supported.

→

## o Hardware require of Redhat Linux :-

- Minimum 2 GB of free Hard disk

- 64 MB of RAM

## o CentOS → it is opensource code of RHEL

- Community Enterprise OS

- it is free & open source alternate of RHEL

- offer used in server environment include web server, database server & other enterprise level application.

- CentOS release 2000

- centos 9 stream 2022

## # CentOS Linux & CentOS Stream:-

- CentOS project is for two linux distribution centos linux and centos stream
- Centos is upstream version as it serve as a platform for upcoming RHEL

## # Few important command & their work:-

- ls : it show present working directory contents.
- # : for root user
- \$ : this is for normal user
- ls -l : display content of current directory in long listing format
- touch : it create new empty file
- ls -t : it show the newest file made in the working directory in descending order.
- cd filename : it will change directory & open that file.
- ls -a : it show hidden file in present working directory.
- touch filename : it will create hidden file.
- cd .. : it will take you to previous directory.
- cp : it help to copy file or folder.
- Kdump : create crash dump in kernel crash.
- cp -r filename : it will copy folder with content.
- ~ : (tilde) it is root
- pwd : it help to check present working directory.

- mv filename : it move the folder in specific document / folder.
- mv -r : copy & paste folder along with document.
- rm -r : it remove folder content & files in folder
- cp -r : it copy & paste in relative previous directory.
- pr filename : used for print
- cat filename : used to check content of file.
- gedit : to open text editor  
gedit vishal.txt - create txt file & open terminal.
- sort filename : show content of file in ascending order
- sort -r : it will sort content in reverse order.
- grep filename.txt : it will search a pattern inside the file
- grep -i contentname filename.txt  
-i ignore the case  
eg: Akshat name.txt  
Naresh name.txt
- Swap - The process use as virtual memory, when physical memory full, system use swap for virtual memory.

- more : it show full page of context of file in page by page or line by line.
- more .bashrc : read files & display the text one screen at a time.
- ps : it will show process shell the process running
- gedit & : run the text editor in background & means it will run processing in the background.
- ps -ef : it show running process of entire system
- man : it will show the manual of PS

UID : User ID

PID : Parent ID

- head -n 2 .bashrc : it will show first 2 header file lines.
- tail .bashrc : it will show last line [last 10 lines]
- head -n 20 .bashrc : show first 20 lines
- sudo : superuser do  
(it means admin execute root command)
- whatis command name : show you what will this command do.

- **mandb**: it will create entire database of what is ensure that you are in root.
- **whereis**: it will show location of directory it show the command where it is located.
- **locate filename.txt**: it search & locate the pattern of the text in file system  
- it is use for searching file & directory in a file system in database.
- **updatedb**: it update the file in database  
sudo updatedb
- **find -name filename.txt**: it will find the file in file system in the current directory.
- **find /home/user/.name filename.txt**: it will find within the specific directory path.
- **cp filename.txt**: it will copy the content of file system into another file
- **bash**:  
it is command line shell & scripting language used on linux, mac os & other UNIX OS
- **diff filename.txt filename2.txt**: it show difference between two files.

- Qa1 add context here which is different  
↓  
two files  
line no.

→ if we add an empty line at the begining of the first script, then we get the first line of the second script:

a : add

d : change delete

c : change

- file file-name : it shows type of file

file name.txt : ASCII text

file-name : directory

/dev/sda : block special like

- jobs : it will show task running / job running  
it is different from ps process.

- firefox & : help to open browser but in the background

- rmdir file name : to remove directory  
(only if directory is empty)

- mkdir : it will make the directory.

- ln -s filename.txt filename-link  
it will create link of the original file or shortcut of file.

- gzip : used to create zip file.
- du : used to get size of file.
  - du -sk anaconda-ks.cfg
    - ↓
      - show kilobyte
    - sh
      - ↓
        - human readable format
  - gzip filename : zip the particular file or folder
  - -d (decompressed) :
    - gzip -d :
      - ↳ decompress the file
  - gunzip : decompressed the zip file
  - tar : archive the file
    - (archive multiple file & folder in single file)
  - tar -cf archive.tar cdac.txt ditiss-1.txt
    - ↓
      - create a name of
    - tar file file
  - tar -tf archive.tar ;
    - ↓
      - list of name of file
    - file
  - tar -xf archive.tar ; extract the file
    - ↓
      - extract

- whoami : name of current login user
- zcat : showing actual content of extract file.
- cal : calendar of system
- bc : calculator for linux on cmd shell.  
(ctrl+c - exit)
- date : current system date & time.
- sudo date & -s : edit date & time
- wc : find word count in file  
wc ditiss.txt ↴
  - 6 ↴ line
  - 17 ↴ word
  - 413 ↴ characterwc -L ↴ line length (number of characters)  
wc -c ↴ character  
wc -w ↴ word
- echo "ditiss sept 2023"  
: it help to point which is written  
in " ". "
- who : how many user accessing the my system  
tty2 - terminal
- w : status of your system  
eg: 13:20:14 up 2:00 2 user load average
  - ↓
  - time
  - ↓
  - last2 hours

## • Network Command :

• ping :

telnet : remote login to system

SSH : remote login & more secure than telnet

FTP : connect only ftp server

SFTP : secure ftp

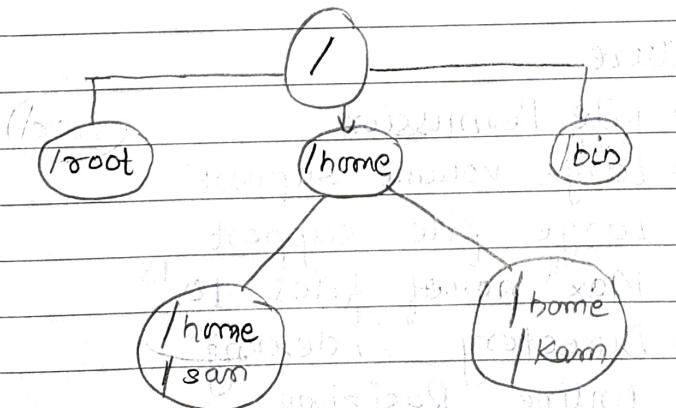
finger : user status

## # The linux file system & permission :-

- Structure
- Various type
- File permission
- To set or change permission of any file or directory.
- special permission

1) File System - software that allow how file are organised

- an OS store, retrieve & organize.



[C : D : E]

[linux - hierarchy is /root is followed by linux]

- Windows → NTFS, FAT, FAT16 or FAT32
- Macintosh → HFS

[linux - ext 2/3/4, xfs]

- 1) Ext [Extended file system] (1st file system)  
→ April 1992 & added to Linux kernel 0.96c

### # Limitations :-

- No journaling - if system crashes & if that journaling is active then we can access the data from getting crashed.
- No symbolic links - you cannot create a link here.
- Max size is 2GB -
- No extended attributes

### # To resolve limitation

- 2) Ext 2 [Remi Kart in Jan 1993] (2nd ext file)

#### Key feature

- file Permission
- Large volume support
- Large file support
- Max no. of files  $10^{18}$
- Directory indexing (locate & search file)
- Online Resizing

3) Ext3 [2001 - Stephen Tweedie in November] 3rd ext file  
key feature

- no need to create
- it is similar to extension 2
- journalized file system

(file cannot be removed or delete even if  
pc gets shutdown / crashes)

4) Ext4 [2008 - 2013] 4th ext file

key feature

→ large file support: (16TB single file size)

→ large volume supports (1EB)

→ online defragmentation (the deleted file  
free space will be removed & combine  
to use space again)

→ delayed allocation

→ encryption

→ max no. of files 4 billion

5) XFS [Silicon Graphics - 1994]

key feature

→ it is default file system in linux

→ large file support - 16 EB

→ large volume - 16 EB

→ scalability

## # Linux file system structure :-

1) /home : default location for all user home directories

→ in this files are kept for users

→ user data are here

→ user personal files.

2) /bin , /usr/bin : directory

- /bin (binary)

- it contain all the commands

3) ls ls - it will show the command in present in single user mode - they will go use /bin

→ only root/admin go for troubleshooting.

• /sbin - contain all command system user admin command.

• /etc - we to store configuration of files

• /var - it contain variable file [eg log files, spool files & temporary files]

• /boot - The machine BIOS or boot & run a boot loader

• /dev - it contain all the device files  
(in Linux everything treated as file)

• /tmp - contain all temporary file which are actually created by the program on system

- /opt - any additional or third party software install

- /boot - it contain info about loader & kernel

### # File & Directory Permission:-

- it is multiuser system, all the file of user stored in single file
- Mechanism is required, to access other user files.

- read (r) to read a file or program
- write (w) to write a file or program
- execute (x) to execute a file or program.

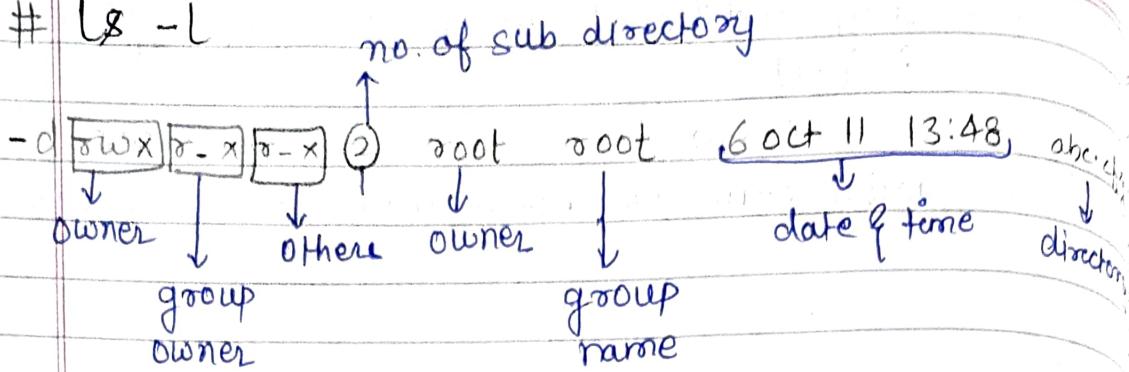
### → for directory :-

- r → Permission to get a listing of file directory.
- w → to create, del, rename directory
- x → to change directory or to use directory as an intermediate

### # Permission for different group of people:-

1) Owner (r,w,x)	Grp A	Grp B
2) Group Owner (r- -)	Ram (owner)	Sita ]
3) Others (- - -)	Radha Ravi } Group owner	Lakhan Deep ] Others

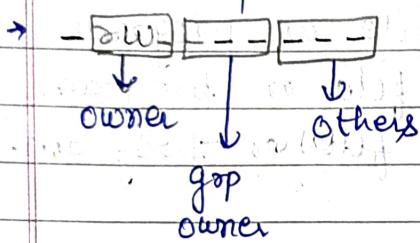
# ls -l



# Changing Permission

→ We can change the permission of directory.

chosen permission



read(r) - 4

write(w) - 2

execute(x) - 1

- chmod `rwxr--r-` anaconda-ks.cfg.

421,401,409

7 5 4

- chmod 754 anaconda-ks.cfg.

Whenever you create a file or directory or empty file this is a default permission

`-rwx r-- r--`

→ (644)

## # Changing Permission

→ U (owner) user

# chmod u+w anaconda-ks.cfg

# chmod o-x

↓  
Others  
execute

# chmod u=rwx, g=rw, o=r anaconda-ks.cfg.

# chmod a+rwx all

↓  
all (including owner, group & others)

## # Default Permission

\$ Umask can change default value

077 - more secure.

0	{ r w x }
1	{ r w }
2	{ r - x }
3	{ r - - }
4	- w x
5	- w -
6	- - x
7	- - -

- By default UMASK is in  
more/etc/login.defs

④	775 → SUID
②	- - - → SGID
①	- - - → sticky bit (t)

## # Special Permission :-

1) Set uid (set owner user id) -

↳ it can set only executable files

- passwd - use to change password of any user

/etc/shadow - - - - hashed version of password  
store here

(password have special permission  
called SUID)

-rwsr-xr-x

↳ SUID

# ls -l /usr/bin/ls

→ -rwxr-xr-x

2) SGID (executable + directories)

root # mkdir | shared-folder|

# ls -l |

# chmod 777 | shared-folder|

# chmod g+s | shared folder|

# mkdir | shared-folder2|

# ls

# su ripu

3) Sticky Bit (restrict who can delete files in a directory on Linux system)  
→ it apply only on directories.

eg Linux

user accounts

Ripul - ripulotxt /cp ripul.txt assignment  
Rahul

# chmod 0+t assignment  
( others + sticky bit)

# SGID: To add SGID bit to file we have to add the group to file.

Chown root: group.name filename change ownership

create group - cat /etc/group

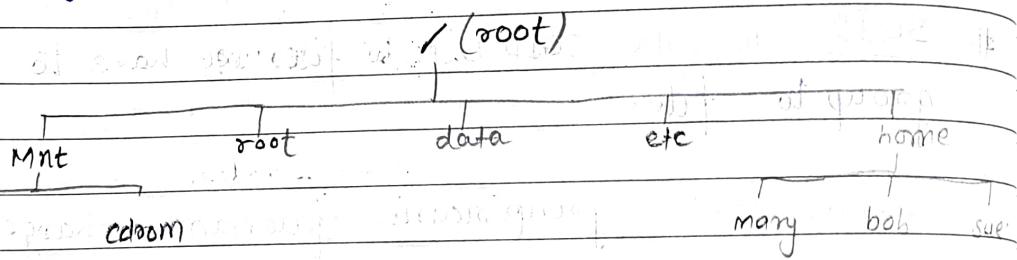
groupadd - group.name

usermod -aG member.name, member.name

chmod 4775 file name

## # Disk Management & Partitioning :-

- Concept of mounting in Linux
- Partitioning
- How to automatically mount a point
- Mounting - it refers to that mounting means the process of making a file system available for access on computer operating system
  - Mount Point - directory to which a device is attached.
  - Any existing directory can be mount point



- Mount Command in Linux :-
- it is used to mount file system & storage device to specific mount point with file system hierarchy.

## # How to mount an USB drive:-

→ Everything in LINUX treated as file

→ Every device is created in /dev folder.

Step to mount USB:

- 1- Every, insert the USB drive into in PC

- 2- find the device of USB.

(it is the directory, we have to create or use existing directory)

3. Create a mount point.

4. Mount the pendrive or mount point.

- In Linux we can automatically mount the USB/ external devices.

→ To mount manually: First stop the auto mount USB drive.

~~(Control+D) Disables auto mount for the device~~

# Systemctl stop, Udisks2.service

system command

name of the service

action

start, stop, resume

Systemctl: - this command start, stop or resume the services in the linux.

# for virtual box we have to give access to USB drive login as root.

CentOS-9 virtual box :- Go to setting > USB > add > choose the sandisk ultra > USB name

# Restart the centos system.

Go to device > USB > add USB with name > Reboot System.

# Partition - primary MBR  
- Extended

⇒ MBR - Master Boot Record (older version)

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

- Sda → it is the hard disk
- Sdb → it is pendrive/external device.

# Lblk - the command show the device's connect

- to fc - it is the respective block device with size of device.

# man Lblk - it is for block device.

- This show the disk size, name, partition details of the device.

[~] # mkdir /pendrive

[~] # mount /dev/sbd1 /pendrive

(name/partition of the USB-external device)

# mount -t /dev/sbd1 /pendrive -- [NTFS partition]

# mount -t vfat /dev/sbd1 /pendrive -- [fat file]

# dnf -y install epel-release  
# dnf -y install ntfs-3g

→ cd /pendrive/

[it will show the content]

• To unmount :-

# umount /usb-drive  
(directory of USB drive)

Then remove the USB drive from system.

## # Hard disk Partitioning:

- one harddrive is divided into multiple logical storage units referred as partition.

### Two types of partition:

- 1) Primary - it hold the actual operating system.  
- it store boot file which boot the system.
- 2) Extended - it is used to store multiple logical units.  
→ single can be extended.

# you can't separate primary partition.

### # Add disk in the virtual machine:-

centos > Go to settings > storage > controller SATA >  
Add hard disk > Create > UPI > Next > size > next.  
then boot the virtual machine.

### # Disk partitioning - it can be done by various types of tools.

e.g:

8 steps:-

- find the device file of the HDD for need to create the partition.
- create 1 primary partition & one extended partition in that multiple logical units.
- create partition using

### Command Steps :-

# `lsblk` - to check list of hard drive.

# `fdisk /dev/sdb`

→ it will give configuration of disk partition.

# `m -l`

→ for help in partition command.

# `n & d`

→ add a new partition

# `p`

→ print a particular table

#

# `P` - primary partition

# ~~e~~ `e` - extended

> `P`

Partition Number [1-4 default 1]

first sector -

Last sector -

> `w` → write table to disk & exit

> `mkfs.xfs /dev/sdb1` [mkfs -t ext4 /dev/sd1]

make file system [it will create file for partition]

> `mkdir /new-drive`

it will create mount point for the new drive

> `mount -t ext4 /dev/sdb1 /new-drive`

it is mounted the partition in the mount point you have created.

## # Auto mounting a partition when reboot

→ Auto mounting required, /etc/fstab file should be edited.

Steps:-

Go to root > ↲

> gedit /etc/fstab [where the configuration file of auto boot will be stored.]

Then in the text editor > fstab

Add your partition mount.

dev   sdb1	pen-drive	ext4	default	0	0
The hard drive name	type of folder			no error	
	mount point	↓			no backup
			file system		

↓  
default permission  
are read/write

→ Save file & reboot system.

## # Linux Volume Management [LVM]:-

∴ it helps to managing & manipulating the storage volume

- To manage two harddrive with logically & make single volume is logical volume management.
- Two harddrive make a single logical hard drive.

### • features:-

- create logical volume by merging two hard disk
- Resize the volume on the processing.
- Move data bet<sup>n</sup> volume without logging.
- Merge multiple disk in a single volume.

## • LVM Components:-

- PV's (Physical volume) - Physical hard drive
- LV's (logical ) - Combined physical HDD & make a single
- VG's (volume group) - Combined the physical HDD.

## • LVM Advantage

- Improved performance
- Easy maintenance
- Resizing of partition
- Adding new disk is also a easy task.

## LVM Configuration :-

# shutdown -h now ↵

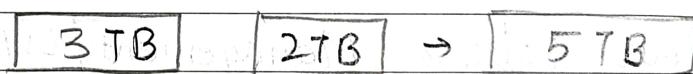
Go to setting > storage > SATA > Add > Create

# Lvm diskscan ↵

(it will show the new harddrive in the system)

# pvccreate |dev|sdb |dev|sdb

physical volume  
creation



→ it will create PV of the newly harddrive for the purpose of Lvm & mark the physical device for Lvm.

# vgcreate, D1T1SS, |dev|sdc |dev|sdd

create the name of new  
volume group

# prs (Physical volume status)

(it will show the physical volume scan)

# vgs (volume Group status)

(it will show the volume group status)

# lvcreate -L 4G -n lvm-part-1 -l /dev/sda1

(logical volume) |  
(create)

| (size)  
(length)

name of  
logical  
volume

| name of  
(logical volume)

| (name of  
volume group)

(it will create the logical volume)

# mkfs.ext4 /dev/dtiss/lvm-part-1

(device file) (volume group)

(logical volume name)

mkfs -t ext4 /dev/dtiss/lvm-part-1

(it will make file system for the lvm group)

# mkdir /lvm-dir

(it will create the directory)

# mount -t ext4 /dev/dtiss/lvm-part-1 /lvm-dir

# lsblk

# df -h

(disk usage in human readable format)

for auto booting the hard drive

# gedit /etc/fstab

## # Linux Authentication

→ it is a process of verifying identifying of a user in which system compares given username & password to system database.

- it uses the /etc/passwd & /etc/shadow for authentication
- system locate user account /etc/passwd
- password validation /etc/shadow
- session initialization

### # /etc/passwd - it will keep user account details

User-name : X : 1002 : 1002 :: /home/username : /bin/bash

```

graph TD
    X[User-name] --- X1["X :"]
    X1 --- X2["1002 :"]
    X2 --- X3["1002 ::"]
    X3 --- X4["/home/username :"]
    X4 --- X5["/bin/bash"]
    X5 --- X6["default shell."]
    X6 --- X7["location of home disc' of user"]
    X7 --- X8["GroupID"]
    X8 --- X9["user-ID"]
    X9 --- X10["when you create user(::) it is a comment of a user"]
    X10 --- X11["it shows password stored in shadow file"]
    X11 --- X12["password stored here"]
    X12 --- X13["( hashed fingerprint )"]
    X13 --- X14["( SHA512 is used )"]
    X14 --- X15["( password never change )"]
    X15 --- X16["( after 7 days get alert for password change )"]
    X16 --- X17["( after this days password created )"]
    X17 --- X18["( password is created for days )"]
    X18 --- X19["( hashed fingerprint ) stored here"]
  
```

The diagram illustrates the structure of a /etc/passwd entry. It shows the fields separated by colons: User-name, password (represented by 'X'), GroupID, User-ID, Home directory, and Default shell. Brackets with arrows point from the bottom to each field, explaining their meanings: 'it shows password stored in shadow file' points to the 'X' placeholder; 'password stored here' and '( hashed fingerprint )' point to the first field; '( SHA512 is used )' and '( password never change )' point to the second field; '( password is created for days )' and '( hashed fingerprint ) stored here' point to the third field; '( after this days password created )' and '( after 7 days get alert for password change )' point to the fourth field; and 'location of home disc' of user' and 'default shell.' point to the fifth and sixth fields respectively.

### # /etc/shadow: for password validation

- it will contain "Hashed version" of password.
- it also stores other password related info of user.

\$ : 19642:0:99999:7:::

```

graph TD
    S[$ : 19642:0:99999:7:::] --- S1["( hashed fingerprint ) stored here"]
    S1 --- S2["( hashed fingerprint ) stored here"]
    S2 --- S3["( SHA512 is used )"]
    S3 --- S4["( password never change )"]
    S4 --- S5["( after 7 days get alert for password change )"]
    S5 --- S6["( after this days password created )"]
    S6 --- S7["( password is created for days )"]
    S7 --- S8["( hashed fingerprint ) stored here"]
  
```

The diagram illustrates the structure of a /etc/shadow entry. It shows the fields separated by colons: User-name, password (represented by ':'), GroupID, User-ID, MaxAge, MinAge, Inactive, LastChange, and NextChange. Brackets with arrows point from the bottom to each field, explaining their meanings: '( hashed fingerprint ) stored here' points to the first field; '( SHA512 is used )' and '( password never change )' point to the second field; '( password is created for days )' and '( hashed fingerprint ) stored here' point to the third field; and '( after this days password created )' and '( after 7 days get alert for password change )' point to the fourth field.

## # User Identifier [UID] :-

The normal user created on the system is assigned.

e.g. # adduser Aman } UID minimum - 1000  
 # password Aman } UID maximum - 60000

## # Group ID [GID] :-

→ Every file have VID & GID owner of the file.

# cat /etc/group

→ it will show the group owner/group user belong to which group

## # Default Configuration for User login :-

Most user info comes from the two files

## a) # /etc/login.defs :

(.def) define the site specific config for shadow password, email location, password creation, password expiration, minimum password length, range of VID's & GID's.

→ login.defs you can set the umask/default permission for the user; group; other user.

## b) # /etc/default/useradd :

→ it will tell inform directory of user

→ /backup all user gets created.

→ default primary group, home directory location, password expiration info shell, selection directory.

## # User Identifier [UID] :-

The normal user created on the system is assigned.

# adduser Aman

} UID minimum - 1000

# password Aman

} UID maximum - 60000

## # Group ID [GID] :-

→ Every file have VID & GID owner of the file.

# cat /etc/group

→ it will show the group owner/group user belong to which group

## # Default Configuration for User login :-

Most user info comes from the two files

## a) # /etc/login.defs :

(.def) define the site specific config for shadow password

→ email location, password creation, password expiration, minimum password length, range of VID's & GID's.

→ login.defs you can set the umask/default permission for the user; group; other user.

## b) # /etc/default/useradd :

→ it will tell inform directory of user

→ /backup all user gets created.

→ default primary group, home directory location, password, expiration info shell, selection directory.

## # Skeleton directory :-

- When you created new user the default files copied from skeleton directory of user.
- if you created any file in the skeleton directory it will copied this all files into the new user created home file.

## # Linux User Administration:-

## • Locking user

(→ Locking user)

# usermod -L Aman

# grep Aman /etc/passwd

! - Added before hash value.

## • Unlock user

# usermod -U username

# grep username /etc/passwd

## • delete user

# userdel aman

→ Delete command with all the files

# userdel -r tom

# grep tom /etc/passwd (to delete home directory of user)

## # Managing Group:-

To add the user # groupadd student

# groupadd -g 10001 groupname

-g → specifies group ID groupadd -g 10001 groupname

-r → create the system group

# groupadd -g 10001 student

# grep -i student /etc/group

# useradd -u 10002 -g 10001 -c "Aman Shukla" Aman  
usedid groupID comment

# Modify a Group - groupmod -n

- groupmod command modify group  
groupmod -n group name

-g → change the GID group ID

-n → new name / change name of group

-o → allow the new GID to be in same group  
an existing group as the system.

-R → change the group ID Range

-k → it will set addition attributes for the group.

- Delete a group

# groupdel student

→ Before deleting a group no members should be  
in the group any more.

→ To check the user belong to how many group

# groups username

Package management in CentOS Linux:-

- Software must be in package which are easily send or receive from one point to another.
- it contain all the configuration, libraries, dependencies header file of the software.
- There are different package management system for CentOS most popular are RPM (Redhat Package Management).

#### • RPM

- it is a package management system used by Red hat based linux system.
- it is used for installation, upgrade, remove of the software.

: cd /var/lib/rpm (where database exist)

- RPM are archive files / bounding files.
- RPM contain the name, version, dependencies, installation scripts in package, software etc metadata.
- it provide the several tools to manage package - rpm, yum, dnf .

#### • rpm (small rpm)

- it is command line tool which is used to install, update, remove & query software package that are in rpm format:

-v: verbose (show installation process)

-b: bashed (show bashed format progress bar)

→ `rpm -i` → package name with its version.  
 ↓  
 (install)

→ `dnf`: it is command line tool package manager.

→ `dnf install package name`

- it will download a package from repositories, of the linux with all the dependencies
- it require internet to download the dependencies older version `dnf` is `yum`.

→ Some common used RPM command are:-

• `rpm -e [package-name with version]` (remove/uninstall software)

• `rpm -qa` (qa - query all)

(it will show the installed package in system)

• `rpm -qf package.name` (it shows package is installed or not)

• `rpm -qi package name`

(detailed info of package)

• `rpm -ql package name`

(it shows list of content of package)

• `rpm qa | grep samba` (show list of installed package)

(pipe to connect one to another command)

→ samba server is used to share file among clients

- : `rpm -qf etc/ command/file`  
(it shows the command/file belong to which package)
- : `rpm -ivh package name`  
(install an RPM package with hash value of package)
- : `rpm -Uvh package name.`  
(upgrade an rpm package)

## # YUM [Yellow dog Update Modified]

- A package manager is used for command line tool in Linux.
- YUM provide a convenient & efficient way to manage software package & dependencies in Linux.
- yum list all (list of all installed packages)
- yum list package name
- yum list package name1 package name2
- yum update "
- yum upgrade "
- yum group list (to show group list of package)
- yum group install "group name"  
(to install package of group like development tool)

: `cd /etc/yum.repos.d/`

(it contains the detail of repository server)

# DNF (Dandified Yum Package Manager) - it is next generation of yum package manager & it designed to provide better performance, scalability & compatibility with other package management tools.

• dnf list all

(it will show list of installed dnf package)

• dnf list package name

• dnf search package name

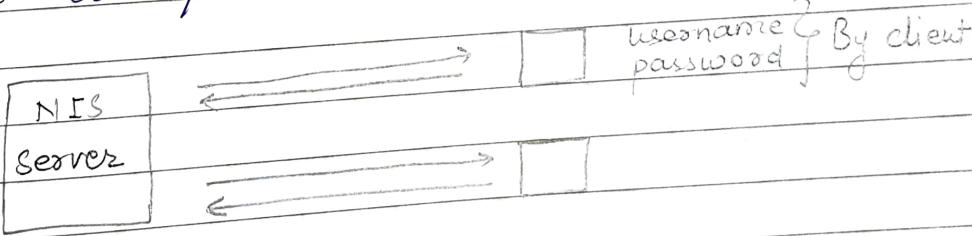
(it show package files)

# Linux System Administration:

## NIS/NFS Server Configuration

### Network Info Service [NIS]

- it is centralized authentication server, in this all the user accounts are created, managed, stored in the NIS server.
- All the user's are client of NIS server.
- If some issue with any user no need to every system, just go to NIS server & resolve issue.
- all users accounts & password are created under the NIS server
- All the system in the NIS network are client of the NIS server.
- On local system username & password will get downloaded from the NIS server.
- The downloaded file contain hashed version of password & it will match with the password type by the user in local system, it will authenticate the client / user.



- We have to keep the backup/multiple NIS servers because if one NIS server goes down or ran out of server/broken/damaged user cannot be logged into their account.
- NIS enable you to create user accounts that can be shared across all system on your network.
- User need to change password on their NIS server only.

#### → Disadvantages:-

- in this username & password in plain text.
- it will not encrypt the username & password.

#### NIS server-client management:-

1) Master Server

2) Slave Server

3) Client Server

GMP XX

classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

## # Configure the NIS - Server :- [CentOS - 8]

- for that we need two VM.
- Use one VM as NIS Server.

Note :- Change NIS server network to Bridge Network

① # Login as root ↵

② # dnf install ypbind , ypserv  
↓ ↓  
( software package ) ( software package )  
( for server ) ( for client )

→ check IP add. of NIS Server.  
↳ # ifconfig ↵

③ # gedit /etc/sysconfig/network

→ setup NIS domain

NISDOMAIN = "Ditiss" save & close

④ # ypdomainname "Ditiss"

→ it set domain name.

⑤ # systemctl start ypserv (service for NIS server)

# systemctl status ypserv (to check status)

⑥ # cd /usr/lib64/yp (initialize NIS Server)

# ./ypinit -m (Master Server)

ctrl + D > y ↵

(it created a directory /var/yp/Ditiss where  
NIS Server database is stored)

→ Now user add & their password.

# useradd Ram &

# passwd 12345 ↳ (add user & password on root)

After creating new user rebuilt database.

# cd /var/yp

# make

(adding user to NIS server using make command)

# setenforce 0 (selinux)

# systemctl stop firewalld (stop firewall)

# iptables -F (flush other routes route for traffic)

Deleted

Imp \*\*

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

# Configure the NIS-client Machine :- [centos-8]

Here we use another VM

Note: change NIS ~~client~~ to Bridge Network

# login as root ↴

# ifconfig ↴  
(check ip add of NIS client)

# ping 192.168.1.11  
(ping server from client to check network)

# dnf install ypbind  
(install NIS client package)

NISDOMAIN = "Ditii" save & close.

# ypmatchdomainname "Ditii"  
(setting domain name)

# gedit /etc/yp.conf.  
(domain ditii server ip add of server  
save & close.)

# authselect select nis --force  
(default authentication happen as local authentication)  
(this will forcefully select authentication from  
NIS on client)

```
# setenforce 0
```

```
# iptables -F
```

```
# systemctl stop firewalld
```

```
# systemctl start ypbnd
```

```
(start NIS-client Service)
```

```
# systemctl status ypbnd
```

```
# ypcat passwd
```

```
(yellow page cat)
```

```
(it shows content of file which is on NIS-Server)
```

- After that try to login user accounts on NIS-Server
- it will login & logout immediately

```
# mkdir /home/unknown
```

- for that create /home/username directory at NIS-Server and then use NPS service
- which help (NIS Server) you to share files among network of NIS servers & NIS client.
- if any user create file on NIS client, it store the NIS Server to /var/nis/

Gang XX  
X

## -o- Configure NFS Server at NIS-server

Go to NIS Server & login as root.

# dnf install nfs-utils  
(install NFS-server package)

# gedit /etc/exports  
(conf NFS to export /home directory of user  
in network)

share entire /home directory share on  
network.

(Ctrl + Alt + F2 = it will open CLI of login)

/home \*(rw) , -A this /home

(you want to export or share this  
folder to everyone in network)

(IP address can also be specify here to share this  
IP read & write permission)

# systemctl start nfs-server

# setenforce 0

# systemctl stop firewalld

# iptables -F

## o Configure NFS client at NIS client

# mount -t nfs 192.168.0.0:/home home  
mounting file type | NFS server IP Add export share folder  
filetype system | | | | mount point  
on client machine

client machine local /home folder get hidden & export  
/home folder will available.

# cd /home

# ls

[it will show NIS server user created  
su username  
(login user on client machine)]

## # NFS [Network file system]

→ it was developed to allow a computer system  
to access directories on remote computer by  
mounting them on a local file system.

→ it allows system administrator to store resource  
file of user in a central location on a network,  
providing authorized user access to them.

→ The system admin on NFS server has defined the  
directories that need to be activated or exported for  
access by NFS Client.

# Runlevel :- it is also called unit / target

few commands :-

① `Runlevel`

(it will show which runlevel is active)

② `man runlevel`

(it will show manual of runlevel)

③ `echo $runlevel`

(show you previous status of runlevel)

④ `echo $pre runlevel`

(list all previous status of runlevel)

⑤ `systemctl list-units --type target`

(show you list of target / tasks, which run after boot process)

⑥ `who -r`

(show you current status of runlevel, also runlevel starting date & time)

⑦ `systemctl show -p wantedby sshd.service`

⑧ `systemctl get-default`

(show default runlevel)

⑨ `systemctl set-default targetname`

(graphical, multि�user)

ls -l /lib/systemd/system/runlevel \*target  
(in this file all runlevel target are kept)

ln -sf /lib/systemd/system/target\_name /etc/systemd/system/  
/default-target  
(it can change target name & set default target)

sudo telinit runlevel\_no.  
(it will change state of current runlevel)

sudo systemctl isolate target name

## # Upstart:-

→ Canonical developed upstart

→ it is replacement of sysv init

## # System D - init System:-

→ from centOS 7 we use this.

→ it has a parallelization which means parallelly many processes starts.

→ it has dependency management.

★

## # systemctl start servicename

(it will start service)

## # systemctl stop servicename

(it will stop service)

## # systemctl restart servicename

(it stop service & start again from scratch)

## # systemctl reload servicename

(it will not stop service, only update configuration in service)

## # systemctl status servicename

(checking status of service)

## # systemctl enable servicename

(whenever system boot the service will start automatically)

## # systemctl disable servicename

(when system boot it will not start automatically)

## # CentOS package management.

- Repository :- A place or container where something is stored
  - it is a kind of database.
- |etc/yum.repos.d/
- Dependencies :- It means a soft<sup>u</sup> is require for run other software.

## # Debian Based OS package management