

Linux 1.2

Monday, September 8, 2025 9:28 AM

Name_Batchname_ModuleName(subject)
MCQ=20 marks 30minutes (submit after atleast 20 minutes)
5 medium to hard (2 mark each)
10 easy direct (1 each)
COD= 1.5 hr
30 marks 5 questions(scenarios)
Paste all screenshots in pdf form

LINUX: Open source- code(can change it) copyLeft, (Windows, Unix- Proprietary codes) copyright -hP unix, solaris-oracle-Sun

Linux and Unix are same architecture
UI advanced, more user friendly due to open source contrib
Multix- multiplex info complement systems on mainframes

PDP
C-Denis retche- Rewrite- usable on different architectures.

FSF- Free Software Foundation - freedom of speech- open source software
Provided for new OSS.

Kernal -core of OS- 1991
LinuX- X due to similarity with Unix

Distributions of LINUX-
Free USD linux, bsd -barcley Software Distribution, Fadra- for new applications , Redat/Suse- enterprise levels

Redat -Linux major share

Package management

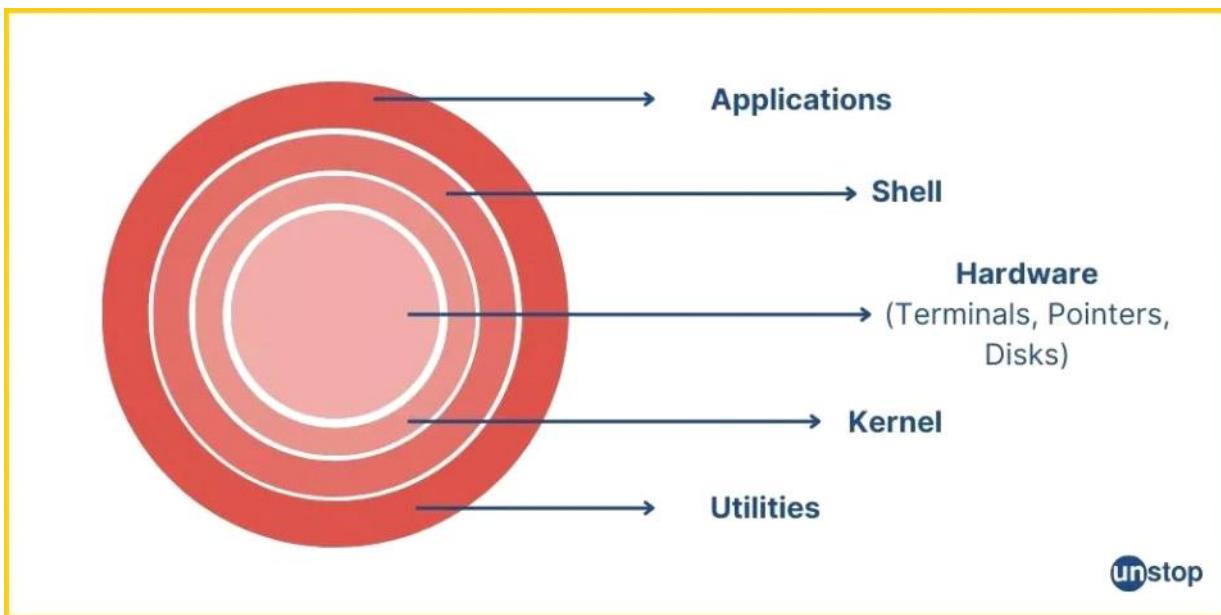
RHEL, SHEL: Redat/Suse enterprise levels
Ubuntu- desktop/ User friendly GUI and Windows also
Linux- open source, stable and robust systems , less prone to cyber hacks, compared to windows
MAC- OS of Apple like Linux

GNU- GPL- General Public License- some no of developers only

RHA 9 is going to be used, RHA 10 is out but not very much wide used, no much advancements required. Not used in production environment
LINUX- MultiUser OS, Many user login at same time
ROOT- parent right
Pseudo- super user do, write the rights to that user
Kernels let Multiple users in same Server, different terminals.

Linux is a :
Multiuser, Multitasking, stable and robust os, secure os, less prone to virus attacks.

Onion architecture- layers



USER- GUI- Application interact with shell

Shell is a UI between User and kernel where user execute the commands and get the output
It is a command interpreter- Interpreter Translates HLL to Machine level language line by line
Whereas compiler does this in one go

- Core function of OS- Resource MGMT (CPU, Memory, HD, H/W, I/O devices)
- Kernal is core of the os
- It performs all core functions of OS like resource mgmt (cpu, memory, Harddisk, I/O devices, Hardware etc)
- Process mgmt
- Handling interrupts
- Booting etc

Linux Principles-

Everything into file

Modular

File system is hierarchical- INVERTED TREE

Avoid captive user Interfaces- (

prevents a user from accessing a system's primary functions until they interact with a specific part of the interface)

Interaction after LOGin

Config data is stored in text in /etc directory

System and server config files are stored in /etc

All device files are stored in /dev directory

Important directories

Home: /root/home/username

User Executables- /bin, /usr/bin, /usr/local/bin

System Executables- /sbin, /usr/bin, /usr/local/bin

Mountpoints- It is a directory on which the device is mounted under /--- /media, /mnt

Proc- process

Config- /etc

Temp files- /tmp

Kernels and Bootloader: /boot

Server data- /var,/srv

System Info- /proc, /src

Shared Lib- /lib, /user/lib, /usr/local/lib

Monolithic kernel

Contains Modular Kernel however
 6 primary subsystems

- Process mgmt
- Inter-process communication
- Memory mgmt
- File system mgmt
- i. VFS- Virtual File system which provide a single interface to multiple file systems
- I/o mgmt
- Networking

HW req-

RHEL7- processor 64 bit, Ram- 4 GB, HDD-10GB+, NW card, keyboard Monitor mouse etc
 RHEL9- processor 64 bit, Ram- 4 GB, HDD-30GB+, NW card, keyboard Monitor mouse etc

SW requirement- DVD or ISO 9 or higher images

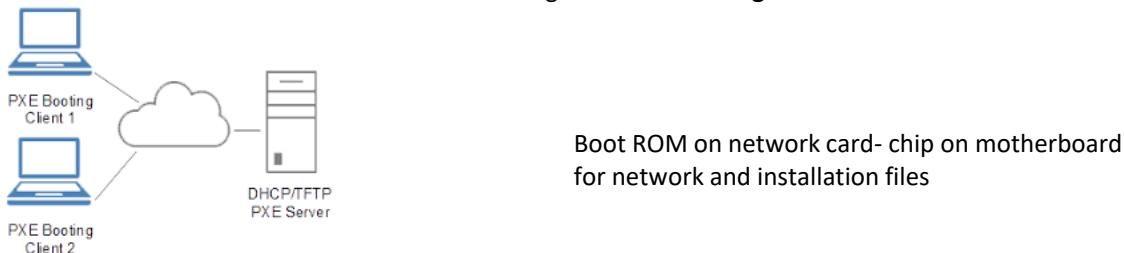
Installation methods-

locally dvd, pendrive, HDD

Over the networks- NFS, FTP,HTTP

Kickstart- **automated/Unattended** installation

PXE Installation- Pre Execution Environment done through **network booting**



Can be installed in GUI, Text-only file/s

Planning - HCL-

HCL- Hardware Compatibility List available on redhat website

It is the list of all h/w tested and certified to be working with Linux OS for that version

Daemon: Service that run persistently particularly a server or service

Network methods- ftp, http, nfs

Basic Important Commands-

>Cat= concatenate= used to display the contents of the file

#Cat /etc/os-release /etc/redhat-release

#Uname Operating system info- linux etc

UNAME:

-r= kernel release

-m= machine

-p= processor

-i = architecture

x86_64= intel 64 bit processor

-a= all system info

free= how much RAM Present(empty in system)

-g GB

-m MB

swap- virtual memory= Part of HDD is used as RAM = Inactive applications are moved from ram to SWAP- HDD(Swap).

Swapping- Moving from Ram to SWAP, Paging- Moving from SWAP to RAM

paging

#comment

history

#tac= to display line by line starting from bottom to Top

White- Files

Is colors

white= regular files

Dark Blue = Directories

Light Blue = Soft link or symbolic link

Red = tar and rpms

Green = Executable files

Yellow = Device Files

Proc- process =Pseudo or virtual filesystem used by kernel for its processes

We refer process by process name refers by Process ID (PID)

PPID- Parent Process ID

Human Reference for:

file = filename

OS reference = inode number or index number

We refer users by Username or uid, we refer group names by gid

#Ls= files

-i gives inode/Index number

#PS= Process state

whatis ps

whatis ls

#Whatis gives single line info regarding the command - display one-line manual page descriptions

#mandb= updates the whatis database //Run this command for updating the db, than whatis works fine

#man= Manual Pages, Help in Linux

Basic command Syntax: Command [Options] <arguments> arg(file or directory name or any info for that command)
example: ls -l /root

#Ls -a will show all files including hidden files,

. Indicates current directory

.. Indicates Parent Directory

Any file or dir beginning with . ls hidden in UNIX as well as Linux

#Ls -l = long listing

drwxr-xr-x. 2 root root 6 Oct 25 15:13 testFile

Type of | u | g | o link owner groupOwner size dateModification file/directory file

d= directory

r= read

w= write

x= execute

= no permissions

Permissions- first 3 owner permissions (users) u

Next 3 group g

Next 3 others o

mkdir /nd= make directory

```

# cd /nd= change directory (nd=NEW directory)
Absolute Path= full path from / to that file or dir, absolute path with /
    Ex= cd /usr/share/doc
Relative Path= Relative to your position in that directory structure, does not begin with /
    Ex= cd zip
#cd .. = Change to parent directory
#cd = change to logged in user's home directory
#cd - = changes to immediate previous directory
#cd ~student = change the directory path to the home directory of that student

```

Creating a file with content:

```

#cd /nd
#cat > filename to create a file
#cat filename to display the content
#cat >> filename to add filecontent

```

(Enter than ctrl+d when you complete with content)
If we use cat>filename again it overwrites

#pwd= Present Working Directory

```

# -l= long Listing, -i= inode/index number, -R= recursive
So, -lR= Long listing Recursive
#ls -li
# | Pipe= output of first command is given as input to next command for further processing
# more= display one page at a time
Ex: ls -lR | more = Output of ls -lR is given to the pager mode i.e. one page at a time

```

Copy a file, move, Rename:

```

#-i: interactive prompt before over writing
#cp= copy
# cp [Options] <source> <destination>
#cp -i file1 / = no dest file name= copy to same file
#cp -avri /nd /td =
    -a: all files including hidden files
    -v: verbose(gives explanation)
    -r: recursive (including sub directories and its contains)
    -i: interactive

```

```

#alias c=clear d=date h=history l="ls -l"
#unalias d
#alias to list aliases

```

```

#mv= move
# mv [Options] <source> <destination>
# -i= interactive
#1. destination=exists and it is a file. It will overwrite
#2. destination=is a directory. It will move the file into that directory
#3. destination=does not exists. It will rename source with destination

```

#touch creates a empty file if not exists, if exists it will update the date and timestamps and contents are not lost.
#touch f{1..10} =>same 10 files with name f1, f2..

#rm -i <filename> to delete the file
#rm -i f1

#rmdir <dirname>

#rm -ri <dirname> will delete a dir and its contents recursively

// Move /nd to /root
ANS: Mv -i /nd /root

->**PUTTY** is a tool to connect windows to LINUX for remote administration
->SSH is Secure Shell , It is used to securely administer a remote machine. Port:22

Softlink or Symbolic Link	Hardlink
Shortcut to the file	Another file which points to the data
If Original file is deleted, Data is lost	If Original file is deleted, Data is not lost

<< given in the file provided

Vi is a powerful visual editor
Vim is a visual Improved editor

#ln -s filename softlink //create soft link
#ln filename hardlink //create hard link
#ls -li filename softlink hardlink
#Dangling link= Softlink pointing to a file which does not exists
LINK Count = no of hard links
Hardlink inode no same as original file, also link count is same
Softlink inode no is not same as original file, also link count is different generally 1
Softlink file size is no of char is link name+file size
Hardlink size same as filesize

///MCQ-VIVA
#df=disk free space for partitions (displays all the active or mounted partitions)
#-T= filesystem, -h=displays all the sizes in human readable format i.e GB, MB, KB etc
#-h= divides by 1024 and gives accurate size in GB, MB, KB etc
#-H= divides by 1000 not accurate

(Block size in linux = 4kb) readable but space waste

#du= disk Usage by file on dir
#-s=SUMMARY

#owner is the one who creates the file
Group is the group of the owner
#CHMOD to change permissions
= assign
+ append
- remove
Ex: chmod u+x

Useradd student
Groupadd prod
Groupadd sales

#chown <username> <fileOrDirname> = change owner of file/dir
#chgrp <grpname> <fileOrDirname> = change group of file/dir
Both at once:
Chown root:sales f2= changes both owner and grp in one command

#use -ugo or -a for all 3=3 user group and others for same alteration to all

#Numerical/Octal permissions - easier to remember and execute

r=4, w=2, x=1

Ex= owner-rwx, group-rw, other-rx

==> chmod 765 f2

#whoami= which user you have logged in as

#who= All users logged in

#w= all user logged in and what they are doing

#stat= Shows Metadata

#echo= printing

echo "Hello World" > testfile //Overwrite

echo "Hello World" >> testfile //Append

""=Interpolate --Expand

"= No interpolation

\$= Get the value of variable //System variable in upper case

``= to substitute the output of that command inside a command //Other way \$(COMMAND)

#grep= to search a string

userID, GrpId of Root User is 0

1. Intro
2. Installation
3. Diff b/w different Linux Oss
4. Basic Commands
5. cp, mv, rm
6. Diff between soft and hard link
7. Permissions for files and directory
8. df
9. du
10. Basic System Commands: cat/etc/os-release, cat/etc/redhat-release, uname, uname -r,-p,-m,-i,-a, free, free -g, useradd, groupadd, alias, whatis, whoami, who, w

9/9/2025 9:00 AM

Booting Process (PPT Topic):

Bios= Basic Input Output System

CMOS tell weather to do bios via hD or DVD

POST- Power on Self test (TO CHECK h/w)

GRUB- Grand Unified Boot loader

MBR- Master Boot Record contain partition table 001-Head track/orCylinder sector

Boot Loader- ~LILO:reload~ so Grub: no reload required

SystemD= Parent Process initializes parallelly (rha <7 has init same but sequential) PID=1

Run level- state of OS, tell what services are running and what are not

The RHEL 7 boot process starts with the system's firmware (BIOS/UEFI) performing the POST, then loading the GRUB2 bootloader from the Master Boot Record (MBR). GRUB2 loads the Linux kernel and [initramfs](#) into memory and passes control to the kernel. The kernel initializes hardware, executes systemd as process ID 1, and systemd mounts the root filesystem, starts necessary services, and brings the system to its configured target state (like a graphical or text-based login prompt).

MBR= 1 Sector: has= 512 bytes

446 bytes Bootloader

64 Partition table= 16X4 Primary Partitions

2 bytes= magic no to validate Partition

Run level 1= Trouble shooting Single user mode

Run level 2= No use due to no NFS

Run Level 3= Client system/ Work stations

#runlevel

n=no previous run level 5=current run level

#init/telinit 3 to change

#systemctl get-default : to get default target

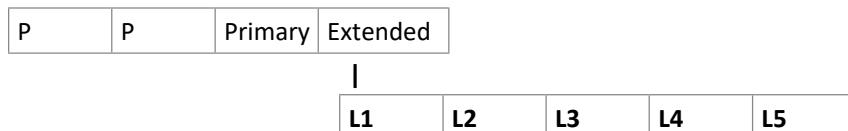
Graphical.target

#systemctl set-default multiuser.target : to set the target, don't set shutdown or reboot

#Ctrl+alt+del= reboot system //don't do

PARTITIONING-

SDA: practice ,(SDB,SDC: exams) and create fresh partitions FOR SERVER, don't delete and create partitions in exam, create fresh



Primary Partition: Designed to contain an operating system and can be made "active" to boot the computer.

Extended Partition: A special type of primary partition that acts as a container for [logical partitions](#)

If you create 4 primary partition, one of which is extended

Extended= not format,mount,store data

#fdisk -l =display partition table

#fdisk /dev/sda : tell which disk partition

p= print partitions

n= create partition -- 3 primary we already have, press e for 4th and press enter for taking all remaining space

Again n= for new logicals, take 40MB space for each, these are logical partitions- and are 5th and onwards

w= to save

d= delete the partition

#partprobe = updates the partition table that is loaded in memory w/o reboot

Else we need to reboot

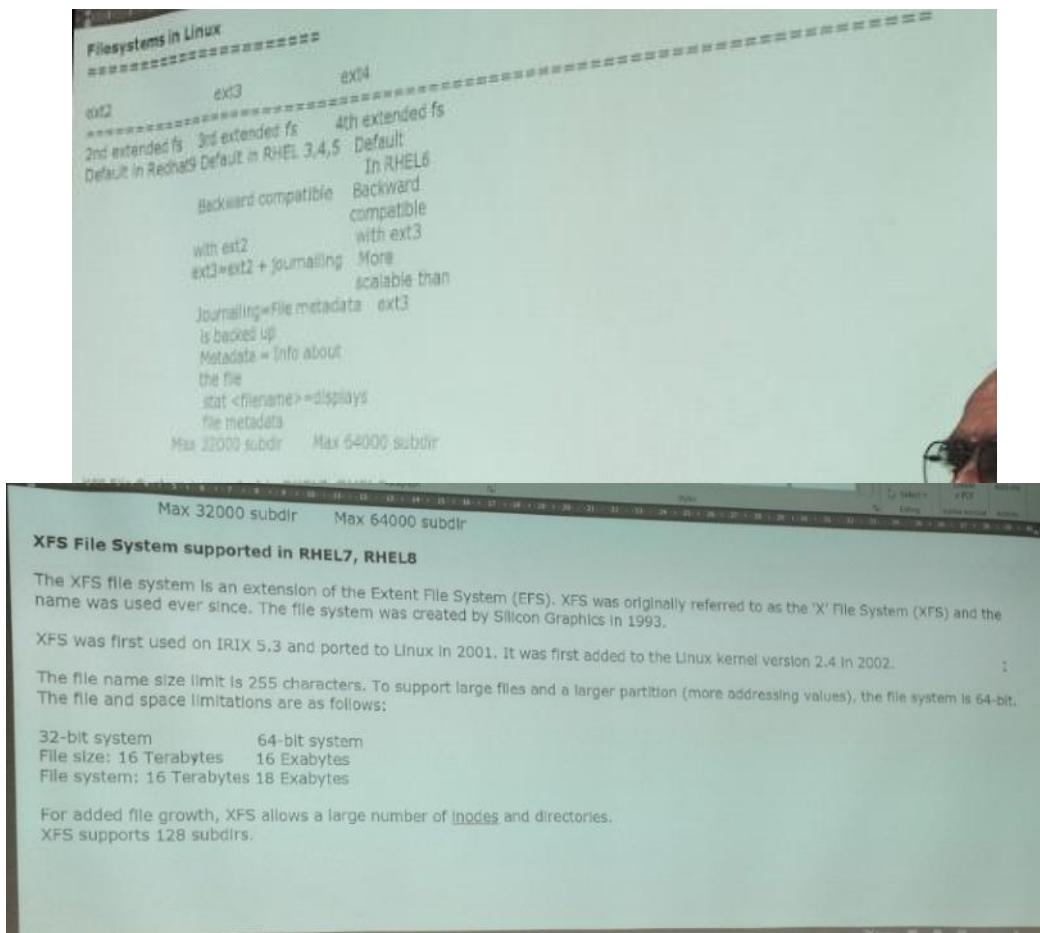
-ID for sda1 and sda5 is same

83=linux, 6=fat16, 8e=LinuxLVM, 5=extended

->FILESYSTEMS(from ppt)- Diff linux filesystems

Ext2,3,4

XFS



#Format Disk to store the data

```
#file <filename> == type of file  
#mkfs -t xfs /dev/sda5 = to format and give file type xfs MKFS= make file system  
#mkdir /data- create a dir and mount it to sda5, that you want to use  
    mkdir /data  
    mount /dev/sda5 /data
```

To check mounted partitions : df -Th

```
#mkfs.xfs -f /dev/sda5 to format the disk and assign the file type xfs to /dev/sda5
```

You cannot delete a mounted partition, delete the mount by #umount and then delete

```
umount /data
```

#Permanent Mount:

Device MountPoint filesystem mountoptions dumpfreq() filesystemcheck (fsck like scan disk in windows0,1,2-9)

```
#/dev/sda5    /data      xfs defaults 0 0
```

VI=Visual interface

#Vi /etc/fstab

```
:i
```

Than add last 2 lines

Than after writing esc,

```
:wq (w=write and quit)
```

```
:w write and save
```

```
:wq! = write and quit forcibly
```

```
:q! = abondoned changes and quit without saving
```

mount -a : to check if errors
#Tail -2 /etc/fstab (Tail shows 10 line last, -2 we took only 2)

We went back to vi again and commented the last line so that it do not get mount again during system reboot

```
/etc/fstab = Devices and partitions we need to add in this file so that they are mounted at boot time.
Fields in /etc/fstab:
 1           2           3           4           5           6
Device     mountpoint   filesystem   options   dumpfreq   fsck

dump frequency = how frequently it is backed up
 0 = never backed up
 1 = backed up daily
 2 = backed up on alternate days

fsck        = filesystem check
 0= never check
 1= check on priority
 2-9= equal priority checked parallelly

Default Mount Options
-----
1. rw   = read, write
2.suid=special permissions are honoured
3.exec=execute permissions are honoured
4.dev=device files are honoured
5.nouser=only root can mount the partition
6.auto=entries in /etc/fstab are mounted automatically with mount -a without rebooting
```

RAID: Redundant Array of Inexpensive/Independent Disks (Raid concepts pdf)

Fault tolerance/Redundancy

Performance

For organization to recover data

Suppose main Disk fails, the Childs are used to recover

Some for performance,

Others for redundancy- fault tolerance

Some for both

Types:

Hardware Raid- Card like h/w has a chip,

Adv: Own processor, memory and much faster

Dis: Expensive

Software Raid- Within the OS commanding

Raid 0= 2-32

Raid 1= 2-2

Raid 5=3-32 // a has bc parity and b has ac parity so fault tolerance

Raid 6=4-32

Raid10=

Concepts= viva, ppt and Mcq

COD= practical

#Raid 5:

fdisk /dev/sda

N create 3 logical partition and change the type by

t Toggle the partition id from 83=Linux to **fd(hx id for this)**-Linux Autodetect

W and enter

Partprobe to save

fdisk -l /dev/sda to check

mdadm -C /dev/md1 -a yes -l 5 -n 3 /dev/sda{6,7,8}

mdadm= Raid Administration command //to initialize raid

/dev/md1 raid device (starts with md)

similarly Disk starts with sd

-C= create raid

-a yes to create the RAID device file,

-l Raid Level

```

-n no of RAID Components
-x spare device
Mkfs -t xfs /dev/md1 //Make file system for format
Mkdir /raiddata
Mount /dev/md1 /raiddata/
Systemctl daemon-reload
Mount /dev/md1 /raiddata/
Df -Th
Cd /raiddata
Ls
Touch file{1..10}
Cat > myfile
    This is my file
Ls
Cd
//check the status
    Cat /proc/mdstat
    Mdadm --detail /dev/md1
//Simulating a raid fault and tolerating it
    mdadm -f /dev/md1 /dev/sda7 :FAIL
Soln:
    Remove the HD and create and add {we already have one to direct add}
    mdadm -r /dev/md1 /dev/sda7 :REMOVE
    mdadm -a /dev/md1 /dev/sda7 :ADD
#mdadm --stop /dev/md1
#man mdadm //Help

```

#lsblk =List all the blocked devices

#blkid =Block id i.e It shows the Uuid (partition) for each device (**UUID** Universal Unique ID)

#SSH= secure administer to remote machine

--> Shell:

Shell is configured using variables, aliases and functions

A=10 b=20 c=hello

Echo a=\$a b=\$b c=\$c

Ps: which shell we are using Bash Shell is the main program

Sh is a **bourne shell** command to execute a script //Execute in bourne shell

Bash is **bash** command to execute a script //execute in bash

Local variable not accessed in shell

Both at the methods to run a script without giving execute permissions

Export the variable to be used. **Export makes it Environmental variable**

Chmod +x script1 //Excute permissions ./script1

Su - student to change the user to student

Whoami tells the current user //who am I logged in as

//Not available for the

Ctrl+d/logout/exit to escape from current user

//It is still a local variable

//now lets make it a global variable

Add variable in export command

Vi /etc/profile

Than run the etc/profile by:

source /etc/profile

Different Types of Shells in Unix/Linux

Shells: Command Interpretation

RHEL: RedHat Enterprise Linux

Different types of shells in UNIX/Linux

- 1) Sh-Bourne Shell by Bourne of Bell Labs AT&T
- 2) Ksh- Korn Shell by Korn of Bell Labs AT &T
- 3) BASH = Bourne again Shell =By Brian FOX for GNU
- 4) Csh = C Shell by Bill Joy of Univ of Berkeley
- 5) Tcsh = Extended C Shell =Community effort

Note: GNU=GNU Not UNIX (One of the projects of Free SW Foundation, Richard Stallman)

Provided UNIX like shell and applications for the New OS.

Kernel for new OS was provided by Linus Torvalds.

The NEW Open Source OS was called Linux.

BASH Shell

Bourne again shell

Default in RHEL (Red Hat Enterprise Linux)

Most user friendly shell (We can recall previous commands and edit, we can use tab to complete file and Dir names)

Shell is configured using:

- a) Variables
- b) Aliases
- c) Functions
- d) Umask

PS1=Primary prompt

Echo \$SHELL

Echo \$BASH_VERSION

Recall by up arrow

Ctrl+a begin, ctrl+e end : moving cursor

Ctrl+u delete from cursor position to begin

Ctrl+k delete from cursor position to end of command

PS1: default/Primary prompt

[\u@\h \W]\\$ == user@hostname\pwd(~(root))

PS1=lti changes it

ex= PS1="\t\\$\h:"

PS2= Secondary Prompt

Shell startup scripts or shell login scripts:

Shell Startup scripts or Shell login Scripts

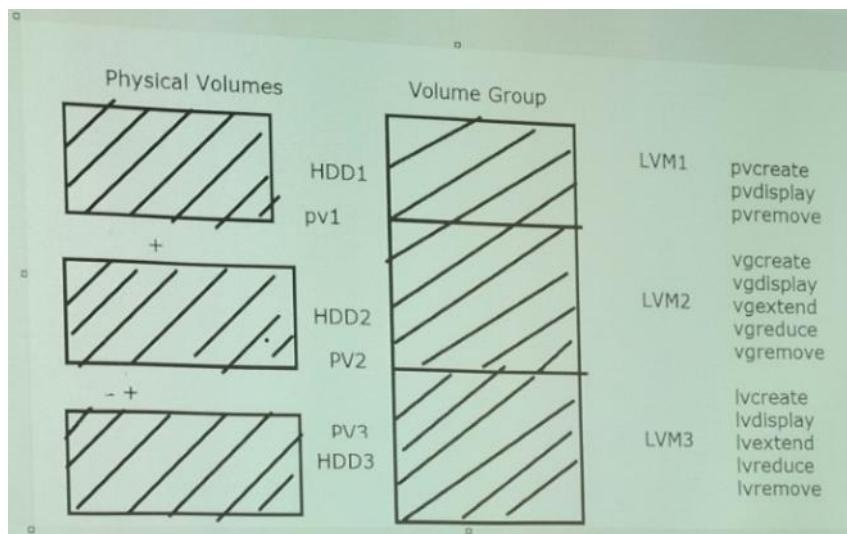
1. /etc/profile= Contains variables which are Exported (Global Variables)
TZ='Asia/Kolkata'; export TZ
export a=10 b=hello
source /etc/profile for it to take effect immediately
2. ~./.bash_profile = hidden under the home Dir of that user. This is the user specific Profile
3. /etc/bashrc= Contains aliases and functions For all users
4. ~./.bashrc = User specific bashrc
5. /etc/profile.d = Dir Contains scripts for your shell environment
6. /etc/skel=Dir Contains scripts that are copied into the new user's home dir

10/9/2025 9:00 AM

LVM-Logical Volume Manager:

Resizing the memory

In LVM we can resize because physical extents(PE) are created. Regular Partition don't have that PE is the minimum size by which we can resize the logical volume. Standard and Raid partition can not be resized.



Steps to create LVM

1. create a standard partition and change the partition ID to 8e. Sda3 is a LVM in our case

#fdisk /dev/sda

Create partition using n

Change pid using t

Save w

Partprobe to take effect

Check using fdisk -l /dev/sda

2. Create Physical Volumes

#pvcreate /dev/sda{9,10,11}

#pv display to check

#pvs to summary

3. Combine physical Volumes and create volume group

#vgcreate vg1 /dev/sda{9,10,11}

#vgdisplay vg1

4. Create Logical Volumes inside volume group

#lvcreate -L 40M -n /dev/vg1/lvm1

Small l goes for physical extent , capital L for size of LVM

Lvm1 created in vg1

Check using VGdisplay

5. Format and mount the logical volumes

- Format one using xfs and other using ext4 because xfs cannot be reduced but can be extended . So for reducing also we format one using ext4

```
#mkfs -t xfs /dev/vg1/lvm1  
#mkfs -t ext4 /dev/vg1/lvm2
```

Create mkdir-

```
Mkdir /lvm{1,2}  
#mount /dev/vg1/lvm1 /lvm1  
#mount /dev/vg1/lvm2 /lvm2
```

Check mount using = #df -Th

System gave /dev/vg1/lvm1 /dev/mapper/vg1-lvm1

Now you are good to add data and file in /lvm1,2 folders.Hence LVM is made...

6. Resize as required

a. Extend

- #lvextend -L +20M /dev/vg1/lvm1
- Vgs shows new size 60Mib partition
- Df -Th shows older size of the file system
- We have only extended the lvm partition and not the file system, for this we need:
- Xfs_growfs /dev/vg1/lvm1
- Check new and older file size by
- Df -Th shows new size

b. Reduce

- Opposite steps
- First umount that partition
1) Umount /lvm2
- Reduce the file system
- Resize2fs /dev/vg1/lvm2 20M
- Run the file system check first
- E2fsck -f /dev/vg1/lvm2 --> Resize2fs /dev/vg1/lvm2 20M
- lvreduce -L -20M /dev/vg1/lvm2
- Mount again
- mount /dev/vg1/lvm2 /lvm2
- Df -Th to check

c. Extend EXT4 Filesystem (same as xfs only xfs_growfs changes to resize2fs)

- #lvextend -L +20M /dev/vg1/lvm2
- Vgs shows new size =40Mib partition
- Df -Th shows older size of the file system
- We have only extended the lvm partition and not the file system, for this we need:
- resize2fs /dev/vg1/lvm2 40M
- Check new and older file size by
- Df -Th shows new size

Vgs and lvs for summary

Vgreduce to remove a PV into VG

VGExtend to add a PV

VRREMOVE will remove the entire volume group

//**Note:** Forcefully removing a volume group if any allocated memory

```
#vgreduce --removemissing vg1
```

```
#vgremove vg1
```

Creating a Snapshot in LVM: #lvmsnapshot= is to backup of LVM at a point in time

```
#lvcreate -s -L 8M -n /dev/vg1/snapoflvm2 /dev/vg1/lvm2
```

```

-s= Snapshot, -L=size, -n=name
#to check the data backup:
    Create a dir and mount the snapshot partition to it
#mkdir /snapoflvm2
#mount /dev/vg1/snapoflvm2 /snapoflvm2
Note: copy commands- cp <source> <dest>
      But for backups- command <dest> <source>

```

#NMAP= Port scanner- It tells you what all ports are opened on your network. It is a N/w Exploration tool and security/ port scanner

```

#nmap localhost //shows ports of server
#nmap <Server IP> //shows server ports on client

```

#SWAP- virtual Memory

Swap Partition= Partition in HDD=Inactive processes and applications in RAM are moved to swap so that RAM can accept new applications or processes
 Same steps as creating a partition in hd
#fdisk /dev/sda
 -p to print
 -n for new
 -t to toggle PID to **82**
 -L to list the PIDs
#mkswap /dev/sda12 = to format the swap
#swapon -a /dev/sda12 = to **activate** the swap
#swapon -s = for summary of swaps

RPM- Redhat Package manager:

Difference between rpm and yum method of package installation

From <https://ltimindtree-my.sharepoint.com/personal/sagar_10843112_ltimindtree_com/Documents/Documents/Linux>All/Difference%20between%20rpm%20and%20yum%20method%20of%20package%20installation%20-%20Copy.docx>

-->RPM:

```

#cd var/ftp/pub/ =
#ls
#rpm -ivh zsh-5.8-9.el9.x86_64.rpm
==>name-version-release.archietecture.rpm
    -i=install
    -v=verbose
    -h=hash(#)
    Archiectecture=x86_64=intel prcoessor
#rpm -q zsh : query whether the package is installed
#rpm -qa : what all packages are installed
#rpm -qi : what it does
#zsh is like your bash
#rpm -qf /etc/samba/smb.conf: if conf got corrupted, qf will show the rpm which provided that package file, so that we can delete and reinstall it using rpm.
#rpm -e zsh : e=erase/ remove the package
#rpm -U zsh/tab: upgrade the existing package to newer version, and if old version is not installed, it will install the old version
#rpm -F zsh/tab: (Freshen)upgrade the existing package to newer version, and if old version is not installed, it will not install the old version

#Cd /var/ftp/pub/AppStream/Packages/
#rpm -ivh httpd-2.4.62-4.el9.x86_64.rpm
    Not installed because it has it has dependencies
    --nodeps: install without dependencies using RPM

```

```
rpm -ivh httpd-2/tar --nodeps  
Install without dependencies using rpm=  
Rpm -ivh package_name --nodeps
```

-->YUM:

Server-Side-

```
#cd /etc/yum.repos.d = for moving to yum packages  
#pwd  
#ls  
#mv centos*/ =centos repos are the files which connects directly to internet  
So it moves internet to root  
#ls  
#pwd  
#vi server.repo  
[BaseOS] //repename  
name=BaseOS rpms //description  
baseurl= file:///var/ftp/pub/BaseOS //path to install  
gpgcheck=0 //gnu check  
enabled=1  
  
[AppStream]  
name=AppStream rpms  
baseurl= file:///var/ftp/pub/AppStream  
gpgcheck=0  
enabled=1  
//GPG= GNU private guard (0=disabled, 1=enabled) it checks against the key If it is a valid package  
#cat server.repo //to check  
#yum clean all //clear cache  
#yum list all  
#yum repolist  
#yum install httpd (fail we already installed by rpm w/o dependencies, so del and install)  
#yum remove httpd  
#yum install httpd
```

In vi editor **5yy** copy 5 line from top where the cursor is
Esc **p** for paste

Types of Packages:

Binary (Ready to use, given by provider)
Source(Raw code and prepare your own application yourself)

FTP SERVER:

```
#Vsftpd= very secure file transfer protocol daemon  
#yum install vsftpd  
#rpm -qi vsftpd  
• The default Red Hat Enterprise Linux FTP server  
• No longer managed by xinetd  
• Allows system, anonymous or virtual (ftp-only) user access  
• The anonymous directory hierarchy is provided by the vsftpd RPM  
• /etc/vsftpd/vsftpd.conf is the main configuration file
```

Service Profile :FPT

Type: SystemV-managed service
Package:vsftpd
Daemon: /usr/sbin/vsftpd

Script: /etc/init.d/vsftpd

Port: 21(ftp: connect), 20(ftp-data: transfer), 23 telnet, 22 SSH (Telnet and SSH are used for remote administration)

Difference between Telnet and SSH		
S.No	Telnet	SSH
1.	Terminal NW	Secure Shell
2.	Insecure username and Pw, data sent In plain text	Username, Pw and data are Sent through encrypted channel
3.	Port 22	Port 23
4.		SSH works on SSL=secure socket layer

Telnet and SSH are utilities for Remote Administration.

Configuration: /etc/vsftpd/vsftpd.conf /etc/vsftpd.ftpusers /etc/pam.d/vsftpd

Log: /var/log/xferlog

Related: tcp wrappers, ip_conntrack_ftp, ip_nat_ftp

When you login to server as a user we connect to varftp/pub directory as a unnnonymous user.

The common conf files are in /etc/

Server side configuration for ftp connection:

1. Download the package vsftpd
 - a. Yum install vsftpd
2. Go to /etc/vsftpd/vsftpd.conf and set configurations --Unnonymous user enabled (Yes) and banner- ftpd_banner
 - a. #vi /etc/vsftpd/vsftpd.conf
 - i. Unonymous user enabled= YES
 - ii. Banner: ftpd_banner=<Welcomemsg>
3. Add firewall
 - a. firewall-cmd --add-service=ftp --permanent
 - i. Alternative:
 - ii. firewall-cmd --add-port=21/tcp --permanent --for ftp
 - iii. firewall-cmd --add-port=20/tcp --permanent --for data
 - b. Systemctl reload firewalld / firewall-cmd --reload
 - c. Firewall-cmd --list-services
4. Configure SELinux: enforcing-->disabled (Additional Security -Mandatory Acccess control)
 - a. vi /etc/sysconfig/selinux
 - i. Selinux= 3 types- (Enforcing, Permissive, Disabled)
 - ii. Reboot to take effect
 - iii. Check it at #selinux shows Enforcing,
 - 1) Getenforce
 - 2) Setenforce 0 (0 means permissive)
 - 3) Getenforce
 - iv. Restart the service to take effect
 - 1) Systemctl restart vsftpd
 - 2) Systemctl enable vsftpd

Client side configuration for ftp connection:

1. Go to server and send the package from server to client
 - a. cd /var/ftp/pub/AppStream/Packages
 - b. scp ftp-0.17.-89.e19.x86_64.rpm root@172.22.61.32:/ (press tab after ftp- and write client IP after root@)
 - c. Scp= Secure Copy over SSH from one system to other
2. Go to client machine and download the ftp by:
 - a. Cd /

- b. Ls
 - c. Rpm -ivh ftp-(tab) - It downloads the ftp
3. Now run ftp <ServerIP>
- a. Username ftp'
 - b. Password ftp
4. **get** to download a package in FTP, **mget** to download multiple package
5. **put** to upload a package in FTP, **mput** to upload multiple package
- a. Inside ftp in client:
 - i. #cd pub/BaseOS/Packages
 - ii. Mget zsh* and enter yes
 - iii. Quit to exit
 - b. The files come inside the directory you are in

-->YUM:

Client-Side-

```
#cd /etc/yum.repos.d = for moving to yum packages
#pwd
#ls
#mv centos*/ =centos repos are the files which connects directly to internet
So it moves internet to root
#ls
#pwd
#vi client.repo
[BaseOS] //reponame
name=BaseOS rpms //description
baseurl= ftp://<yourServerIP>/pub/BaseOS //path to install
gpgcheck=0 //gnu check
enabled=1

[AppStream]
name=AppStream rpms
baseurl= ftp://<yourServerIP>/pub/AppStream
gpgcheck=0
enabled=1
//GPG= GNU private guard (0=disabled, 1=enabled) it checks against the key if it is a valid package
#cat client.repo =to check
#Yum clean all =clean cache files
#Yum list all =list all the packages
#Yum repolist
#Yum install httpd

#hostnamectl =for seeing hostname details
#hostnamectl set-hostname <name> =to set host name
#usr= UNIX System Resources
#var/log= Contains log files, used for monitoring and troubleshooting
```

11/9/2025 9:00 AM

File sharing services are:	FTP :done 9/9
	NFS
	SAMBA

Look at nft-utils:

PACKAGE
PORT
DIRECTORY

CONFIG.
(RPC: remote procedure call)

NFS:

Service Profile: NFS-

Type: System V-Managed Service
Package: nft-utils
Daemons: rpc.nfsd, rpc.lockd, rpciod(remote procedure call), rpc.mountd, rpc.quotad, rpc.statd
Ports: 2049(nfsd), Others assigned by portmap(111)
Config: /etc/exports
Related: portmap(mandatory), tcp_wrappers

NFS Server:

Exported directories are defined in /etc/exports
Each Entry specifies the host to which the filesystem is exported plus associated permissions and options
Options should be specified
Default options: (ro-read only, sync, root_squash- you can not get root access for the folder shared by client (use not root_squash to get it)

NFS Utilities:

exports -v
showmount -e hostname
rpcinfo -p hostname

Client-Side NFS:

Implemented as a kernel module
/etc/fstab can be used to specify network mounts
NFS shares are mounted at boot time by /etc/init.d/netfs
autofs mount NFS shares on demand and unmount them when idle

Server side steps:

```
#yum install nft-utils
#mkdir /nfsshare
#cd /nfsshare
#cat > fiel1
    This file is created for sharing with NFS
#touch f{1..10}
#cd
#vi /etc/exports
    /nfsshare *(rw,sync) //What we are sharing to whom
#ls -ld /nfsshare/ - permissions
    -ld for only directory(folder) permissions also and not files
#chmod -R 777 /nfsshare
    -R for recursive, 777 (rwx permissions for -root,-u, -others)
#firewall-cmd --add-service=rpc-bind --permanent //use tab after rpc
    RPC- helps in communication= map the program for the port number
    RPC is a program number database, a library routines for RPC
#firewall-cmd --add-service=nfs --permanent
    NFS- file sharing service type
#firewall-cmd --add-service=mountd --permanent
    For mount sync
#firewall-cmd --reload or Systemctl reload firewalld
#firewall-cmd --list-services
#systemctl stop rpcbind
#systemctl start rpcbind
    Both equal to restart but for rpc use start stop only, it may have some ambiguity
#systemctl restart nfs-mountd
#systemctl restart nfs-server
```

```
#systemctl enable rpc-bind  
#systemctl enable nfs-server  
#exportfs= to show what is exported  
#exportfs -r= to re export files and folders
```

Client side steps:

```
#yum install nfs-utils  
#showmount -e <your server ip> //to show what is shared on server -e=exported  
#mkdir /share //folder to mount shared dir from server to client  
#mount -t nfs <your server ip>:/nfsshare /share  
#cd /share  
#ls //you will see server files here
```

SAMBA: file and print sharing between linux and windows

Four main services are provided:

- Authentication and authorization of users
- File and printer sharing b/w linux and windows systems
- Name resolution
- Browsing (service announcements)

Related:

Smbclient command line access
Linux can mount a Samba share using the cifs or smbs file system

Service Profile: SAMBA-

Type: System V-managed service
Packages: samba, samba-common, samba-client
Daemons: /usr/sbin/nmbd, /usr/sbin/smbd
Script: /etc/init.d/smb
Ports: [NetBIOS] 137 (-ns= NameServer), 137(-dgm= datagram), 138(-ssn= Session), [SMB over TCP] 445(-ds)
Config: /etc/samba/*
Related: system-config-samba, testparm

{Dd to delete a line in VI}

Server side steps:

```
#yum install samba*  
#yum install cifs-utils //Common Internet file system (for windows and linux both)  
#rpm -qa | grep samba  
    To check all what is installed  
#cd /etc/samba  
#ls  
#cp smb.conf.example smb.conf //example is like a template, so we just edited that template itself  
#vi smb.conf  
    //Add hosts and than  
    // at end copy 7 lines and paste them (remove comment and change them as follows)  
  
hosts allow = 172.22.61.  
  
comment= Public Stuff  
path = /sambashare  
valid users = root student  
public = no  
writable = yes  
printable = no  
  
#firewall-cmd --add-service=samba --permanent  
#-firewall-cmd --reload  
    Allow firewall and reload  
#systemctl restart nmb (NetBIOS- Windows service)
```

```

#systemctl restart smb (Samba Service)
    Restart the services
#sbmpasswd -a root
    New pass= redhat
#useradd student
#sbmpasswd -a student
    New pass= redhat

#systemctl restart nmb (NetBIOS- Windows service)
#systemctl restart smb
#systemctl enable nmb (NetBIOS- Windows service)
#systemctl enable smb
    Restart the service, set samba password for both users(root, student) than restart and enable the services
#mkdir /sambashare (folder to share)
#cd /sambashare
#touch f{1..10}
#cat > file11
    Hi, This file is for SambaShare

```

Client side steps:

```

#yum install samba* cifs-utils
#smbclient -L <IP Address of Server> -U root (-u= user) write password redhat
#mount -t cifs //172.22.61.31/public /opt -o username=root
    Mounting on /opt because it is free we can mount at any directory
    -t= mount client server
    -o= override
#cd /opt
#ls //ALL SET

```

-----#testparm /etc/samba/smb.conf = This command help us to check the syntax-----

Seven Fundamental File types in Linux: (From ppt)

S.No.	File Type	ls -l display (1st char of first column)
1.	Regular File	-
2.	Directory	d
3.	Symbolic Link Or Soft Link	l (small L)
4.	Block Special File (Ex: hdd /dev/sda)	b
5.	Character Special File (Ex: /dev/ttyS0,/dev/lp0)	c
6.	Named Pipe (Use to pass data between processes)	p Ex: ls -l /dev grep "p"
7.	Socket (End Point for Communication) Used for NW Communication	s Ex: ls -l /dev grep "s"
	file <filename>	Shows file type with more information

```

#ls -l
#cat >file1
#ls -l file1
#ln -s file1 sf1
#ls -l file1 sf1

```

```
#ls -l /dev/sda
#ls -l /dev/lp0 /dev/ttys0
Character special file- transfer bit by bit
#ls -l will show file type, but for more information: file <filename>
#for socket:
file /dev/log
And than #file /run/systemd/journal/dev-log <-- this is a socket file
```

#User Administration:

It is the most important functionality
Useradd user is stored in /etc/passwd file

#useradd(or adduser) [options] <username>

Options:

- g=primary group
- G=secondary group
- s=Shell
- c=Comment or full name
- d=Home Directory

Primary	Private	Secondary or additional or supplementary
Mandatory	Optional	For additional access
User can be a member of one pri group only	If username and primary group name are same	User can be a member of more than one sec group

#useradd -g prod -G sales user1 // create user1 with primary group prod and sec grp sales

#groups user1 [or] id user1 = shows group of a user user1

#cat /etc/passwd

Uid,gid for root is 0, uid/gid for system users is 1-999, uid/gid for regular users is 1000-6000

/etc/passwd = Contains user accounts, It has 7 fields, can be viewed by:

#grep user1 /etc/passwd

7 fields as follows- username:password:uid:gid:comment:homedir:shell

Ex: user1:x:1001:1001::/home/user1:bin/bash (:: because it has no comments)

#cat /etc/group

fields as follows- groupname:grp passwd: grp id: secondary member of that group

#usermod [Options] <username> =to modify user details

useradd -s /bin/ksh sagar //Create a shell user Sagar **-s=shear**

(use #yum install ksh= to download shell package)

grep sagar /bin/passwd =check initial details

usermod -c "CR" -s /bin/bash sagar = user modify sagar add comment "cr" and change shell to /bin/bash

/etc/shadow contains encrypted user password

Fields are as follows:

username : encrypted password: 3-8(output of chage command) : 9 (reserved for future use)

Use- passwd <username> = to change the password

Than observe you will see encrypted pass and same passwd encrypted for 2 user is different

#userdel -r user1 = to delete a user

#find /root -user student -o -group prod

Search for files in /root with user student (-o=or, -a=and) group prod

#uptime= how long the server is up

#last= last time I rebooted or logged in the system or service

#chage- to change the age information for username

Min password age= min no of days to change password

Max passwd age= max no of days to change password= EXPIRY time
 Last Password Change:
 Pass expiration warning: date to change pass
 Pass Inactive: [-1=change anyday after expiration, +1=change within 1 day of expiration]
 Acc Expiration data: date to expire the acc
 Chage -l sagar= to see the details

```
#groupadd -g 2001 it (-g for grp details <gid> grpname)
#grep it /etc/group= to search it group and check group id(2001) and name
#groupmod -n mis it = grpmod to change group details , -n to new grp name
#grep mis /etc/grp = to recheck that the grp name of id 2001 changes
```

GREP for searching that string in a file and show (ex user details in grep /etc/passwd)

#ifconfig for ip address
#route -n or netstat -rn to check gateway,router address
nmcli connection show =Viewing Active Network Connections
 ✓ Check running services systemctl list-units --type=service
 ✓ Check the status of a specific service systemctl status sshd
 ✓ Find the services path cat /etc/services | grep ssh

Where is your network card stored?	/etc/sysconfig/network-scripts/ifcfg-ens160
To know your LAN Card	Ifconfig
Router and Gateway Address	Netstat -rn or route -n
Deactivate LAN Card -(loose your network)	Ifdown /etc/sysconfig/network-scripts/ifcfg-ens160
Activate LAN Card -(Add your network)	Ifup /etc/sysconfig/network-scripts/ifcfg-ens160
Lines and words and bytes	wc file1 -l lines -w words -c bytes -m chars

#SSH: (VV Imp MCQ+COD) :

Objective- Login to the server without password i.e using ssh= RSA key

Server Commands:

```
#rpm -qa | grep ssh (to see what all packages for ssh)
#rpm -qa | grep ssl (what all packages are installed)
#vi /etc/ssh/ssh_config
      Write : Banner /banner
#vi /banner
      Your msg
#firewall-cmd --list-services
```

Client Commands:

```
#ssh-keygen //Created a RSA=Rivest Sharier Adremain key for passwordless login to server
#ssh-copy-id root@172.22.61.31 //Copy the SSH key created to the server to login
#ssh root@172.22.61.31 //success Able to login w/o passwd now
```

#Different Levels of Help:

- whatis <command>
- <command> --help

- man <command>

- Search string /<string>
 - Next n
 - Previous N

- Info <command>

- n next page
 - p previous
 - u half page up

- Cd /usr/share/doc/zip

- **cat README**

12/9/2025 9:10 AM

Apache Web-Server-

Currently the most popular web server

Cross-Platform Support

Your type of OS doesn't really matter since Apache is versatile

Configurable and flexible

Open-source and community driven

Security Failures

Shared and virtual Hosting

No GUI

Difference between Apache and other web server- see in LMS

Service Profile:

Type: SystemV-managed service

Packages: httpd, httpd-devel, httpd-manual

Daemon: /usr/sbin/httpd

Script: /etc/init.d/httpd

Ports: 80 (http), 443 (https)

Configuration: /etc/httpd/* , /var/www/*

Related: system-config-httpd, mod_ssl

Main config file- /etc/httpd/conf/httpd.conf (configure web server to host web pages)

Default Doc Root= /var/www/html (i.e the first page we lands on like root page)

Server side steps:

```
#yum install httpd
```

```
#firewall-cmd --add-service=http --permanent
```

```
#firewall-cmd --reload
```

```
#firewall-cmd --list-services
```

```
#systemctl restart httpd
```

```
#systemctl enable httpd //to make it persistent after reboot
```

```
#curl 172.22.61.31 //curl don't work
```

```
#yum install lynx //lynx
```

Lynx is a text-based web browser for Linux and other Unix-like operating systems. It allows users to browse the World Wide Web directly from a terminal window, without the need for a graphical user interface.

Curl does the same

```
#rpm -qi lynx
```

```
#lynx 172.22.61.31 (your ip address)
```

Go to the base machine of your VM and open edge browser and type your server IP address
You will see centos welcome page

Now to have your HTML page create a HTML file=index.html inside /var/www/html and all set
#cd /var/www/html
vi index.html
Welcome to Ltimindtre

Name Resolution= DNS:

```
#vi /etc/hosts
<server IP address> <hostname>
172.22.61.31 server.lti.com
172.22.61.31 virtual.lti.com
#source /etc/hosts
#ping -c3 server.lti.com
#ping -c3 virtual.lti.com
```

Creating Actual and Virtual Host for different websites:

```
#cd /etc/httpd/conf
#vi httpd.conf
At end after EnableNmap add following:
<VirtualHost server.lti.com>
ServerAdmin root@server.lti.com
DocumentRoot /var/www/html
</VirtualHost>

<VirtualHost virtual.lti.com>
ServerAdmin root@virtual.lti.com
DocumentRoot /var/www/html/virtual
</VirtualHost>
```

Now also create a welcome page for Virtual Host

```
#cd /var/www/html
#mkdir virtual
#cd virtual
#vi index.html
<h><b> Hi Virtual Host </b></h>
#systemctl -restart httpd
It may be the case that it config may be wrong so use:
-----#apachectl configtest -----
#lynx server.lti.com, lynx virtual.lti.com now points to two different web pages
```

SQUID- Web Proxy Server- (Advanced security at Application Level,)

A wall between you system and internet restricting who can browse and what can he browse.

Rpm required: SQUID

Main config file: /etc/squid/squid.conf

Port : 3128 (configurable: can be changed to any available port)

Service: squid

READ LMS NOTES for more important commands

Steps on server:

```
#yum install squid
#rpm -qi squid
#vi /etc/squid/squid.conf
```

ACL= Access Control List, SRC source, Dest Destination

```

#acl Iti src 172.22.61.0/16 ClassA = 8 , ClassB=16
Allow browsing: http_access allow Iti
Config Http port: http_port=8080-Allow that port also
#firewall-cmd -add-service=squid --permanent
#firewall-cmd --reload
#systemctl restart squid
#systemctl enable squid

#systemctl status <Package>
#systemctl stop <Package>

#scp: scp copy your file from your system to remote system
#scp file1 root@172.22.61.32:/ (copy file1 to root@<ClientIP> at / directory)
#scp -r /newDir root@172.22.61.32:/ (copy folder and its subfolder and files recursively to client)

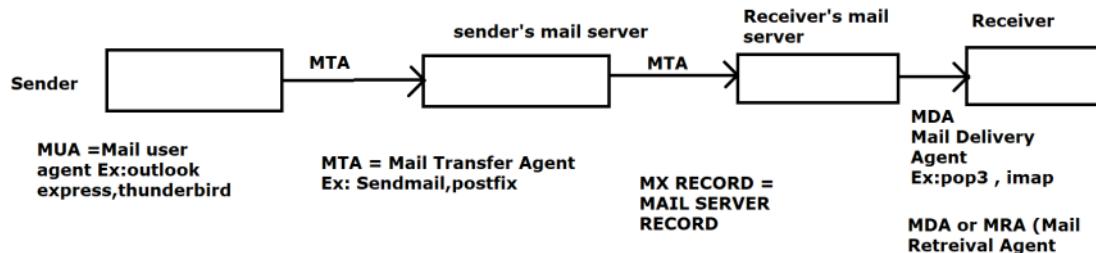
#rsync: used to sync a file or directory to the client // less bandwidth usage
#rsync -r /newDir root@172.22.61.32

```

#find command:

Linux Mail Server-

Browser/MailClient = MUA= Mail User Agent (Sender)
 Mail goes to the senders Mail Server
 Mail goes to receiver's Server
 Than the mail goes to the receiver



Service Profile:

Type: System V-managed service
 Packages: sendmail, sendmail-cf, sendmail-doc
 Daemon: /usr/sbin/sendmail
Port: SMTP=25, POP3=110, POP3S=995, IMAP=143, IMAPS=993
 Configuration: /etc/mail/sendmail.mc, /etc/aliases, and others
 Related: **procmail**(MDA), **spammassassin**, **tcp_wrappers**, sendmail-doc

Main Configuration file /etc/mail/sendmail.cf we have to edit /etc/mail/sendmail.mc and update /etc/mail/sendmail.cf using the **Macro m4**, .cf is system understandable file type, so we write in text and convert
`#cd /etc/mail/
#m4 sendmail.c > sendmail.cf`

Server(sender) steps:

```

#yum install sendmail sendmail-cf m4
#vi /etc/mail/sendmail.mc
Dnl=do not read this line
-->dnl Daemon_options

```

```
#m4 sendmail.mc > sendmail.cf
#firewall-cmd --add-service=smtp --permanent
#firewall-cmd --reload
#firewall-cmd --list-services
#systemctl restart sendmail
#systemctl enable sendmail
#systemctl restart sendmail
```

Dovecot(MDA) steps: Dovecot is the MDA i.e Mail Delivery agent

Service Profile-

Type: System V-managed service
Packages: dovecot
Daemon: /usr/sbin/dovecot
Port: POP3=110, POP3S=995, IMAP=143, IMAPS=993
Configuration: /etc/dovecot.conf
Related: **procmail**(MDA),fetchmail, openssl

Steps-

```
#yum install dovecot
#vi /etc/dovecot/dovecot.conf
    Protocols are commented, add protocols
    -->protocols = imap pop3 pop3s imaps
#firewall-cmd --add-port=110/tcp --permanent
#firewall-cmd --add-port=995/tcp --permanent
#firewall-cmd --add-port=143/tcp --permanent
#firewall-cmd --add-port=993/tcp --permanent
#firewall-cmd --reload
#firewall-cmd --list-services
#systemctl restart dovecot
#systemctl enable dovecot
#systemctl restart sendmail
```

Send mail:

```
#mail root@172.22.61.32
    It will Install s-nail to send
#mail root@172.22.61.32
    Enter Subject
    Enter Description and ctrl+d
#mail student
    Sends mail to the user student
    Check it by logging in as student, and write mail, than enter mail no to view and quit to exit
```

```
#mailq =to see the unsent mails
#sendmail -q =to (Force Push)retry to send mails to the dest
```

Client side steps:

```
#yum install s-nail sendmail
#mail (to check the mail box)
```

13/9/2025 9:00 AM

LFTP is light weight file transfer protocol.

Server:

```
#whatis lftp
#yum install *ftp*
```

Client:

```
#yum install lftp
#whatis lftp
```

```

#lftp 172.22.61.31 //<your Server IP>
#ls
#cd /pub/BaseOS/
GFTP is Graphical FTP.

#get <filename> = to download a file or package
#mget <filename>= to download multiple files or package
#put to upload a package or file
#mput to upload multiple packages or file

```

Covert ext2 to ext 3:

```

#fdisk /dev/sda
    New partititon
#partprobe
#mkfs -t ext2 /dev/sda18
#mkdir /data
#mount /dev/sda18 /data
#df -Th = ext2
#tune2fs -j /dev/sda18
#umount /data
#mount /dev/sda18 /data
#df -Th  = for partitions which are mounted, T for file system Type
#du = for disk usage = what all disk is used for

```

POSTFIX- Mail Transfer Agent

Another way to configure the server, Parallel Execution, -->faster

Service Profile-

Type: SystemV-managed service
 Package: postfix
 Daemons: /usr/libexec/postfix/master and others
 Script: /etc/init.d

```

#yum install postfix
#rpm -qi postfix
#alternatives --display mta
#alternatives --config mta
    Set postfix the as default by entering 2 instead of sendmail
#alternatives --display mta
#postfix -e "inet_interfaces=all" //Command to add this "" to the file initialized
#vi /etc/postfix/main.cf
#firewall-cmd --list-services //smtp need to be allowed
#firewall-cmd --list-ports
#systemctl restart postfix
#systemctl enable postfix
#systemctl restart dovecot
#mail student
#su - student

```

NTP (chrony) Server: Network Time Protocol

Server Side: It is same for client side also
 npt id is used to synchronize the time with the accurate time server on internet.
#yum install chrony
#rpm -qi chrony
#vi /etc/chrony.conf
Add server- {server 172.22.61.171 iburst} iburst= send a burst of packets
#vi /etc/chrony.conf
#systemctl restart chronyd

```
#systemctl enable chronyd  
#chronyc = Utility where we can see what are the servers by typing {sources} in it
```

#Tar: to backup or restore dir

The tar command in Linux is a powerful utility used for creating, viewing, and extracting files from archives. It is short for "tape archive" and is widely used for bundling multiple files and directories into a single archive file, often for backup, distribution, or storage purposes.

```
#mkdir /newdir  
#cd /newdir  
#touch f{1..10}  
#cat > f11  
#ls  
#cd  
# tar cvf /opt/newdir.bak /newdir  
Syntax: tar [Options] <destination> <source> (- is optional to add options)  
-c= create backup  
-v= verbose  
-f=filename  
#cd /opt  
#ls  
#tar tvf newdir.bak (-t=list backup files)  
#tar xvf newdir.bak (-x=extract or restore from backup)  
#ls  
#cd
```

Directory or file both can be backed up but only file can be compressed

Compress a file using gzip:

```
#gzip -v anaconda-ks.cfg  
#ls -lh anaconda-ks.cfg.gz --to see the compressed file  
#cat anaconda-ks.cfg.gz //can not view compressed file in cat  
#zcat anaconda-ks.cfg.gz //to view a compressed zcat  
Unzip-  
    #gunzip -v anaconda-ks.cfg.gz  
#bzip2 compress more than gzip  
    -Use #less <filename> instead of zcat to view bzip2 compressed files  
    --use bunzip2 instead of gunzip to uncompress
```

Combining gzip and tar:

```
tar czvf /opt/newdir.bak /root =to back up and compress  
tar tzvf root.tgz =to list backed up compressed file  
tar xzvf root.tgz =to extract backed up compressed files
```

Combining bzip2 and tar:

```
tar cjvf /opt/mydir.tbz2 /mydir = to back up and compress  
tar tjvf mydir.tbz2 =to list backed up compressed file  
tar xjvf mydir.tbz2 =to extract backed up compressed files
```

VI= Visual Editor:

Similar vim=visual improved editor
See vi editor file definitely

#find command:

```
Find[path]<criteria>  
#find /etc -name passwd  
    -name=search by file name  
    -size for size of file  
        1024 k search for exact 1 MB file  
        +1024k search for greater than 1MB file
```

-1024k search for less than 1MB file

BACKUP Partitions:

```
#xfsdump -f /backup /dev/sda5  
Dump(backup), -f=filename , partition  
Media Label ="new"  
#xfsrestore <dest> -f <source>
```

-->

```
#mkdir /backup  
#cd /backup  
#xfsdump -f /backup /dev/sda5  
#ls  
See bak file in it  
#xfsrestore /root -f bak  
#cd  
#ls
```

All files of folder mounted on /dev/sda5 are now in /root

15/9/2025 9:00 AM

DNS- Domain Name System

BIND- Berkeley Internet Name Daemon

DNS-

Converts HOSTNAME to IP address (Forward Lookup) (a record)
Converts IPaddress to HOSTNAME (Reverse Lookup) (prt record)

DNS Records:

A record Address record maps hostname to IP
PTR record Pointer record maps ip to hostname
MX Record Mail Server Record
NS= Name server Record(DNS Record)

Servers:

Master: contains all the records
Slaves
Cache

Service Profile:

DAEMON: BIND
MAIN config files: /etc/named.conf, /etc/named.rfc1912zones
Forward and Reverse Lookup files: Under /var/named
Ports: **53-DNS, 953-RNDC** (Remote Name Daemon Control)
Service or Daemon to start: named

Practical-

```
#yum install bind bind9.18-utils  
#vi /etc/named.conf  
    -any for ports and listen on  
#vi /etc/named.rfc1912zones  
    Comment out first zone,  
    5yy localhost master lines and paste  
        zone "ltim.com" IN {  
            type master;  
            file "ltim.for";  
            allow-update {none; };  
        };
```

```

5yy down also:
zone "61.22.172.in-addr.arpa" IN {
    type master;
    file "ltim.rev";
    allow-update {none;};
};

#cd /var/named
#ls
#cp named.localhost ltim.for
#cp named.empty ltim.rev
#vi ltim.for
#SOA=Start of Authority
server.ltim.com. root.ltim.com.
Delete last line and
@ IN SOA server.ltim.com root.server.ltim.com( same)
NS server.ltim.com.
server.ltim.com. A 172.22.61.131

#vi ltim.rev
IN SOA server.ltim.com. root.server.ltim.com(same)
NS server.ltim.com.
131 PTR server.ltim.com.

#ls -l
#chown named.named *
#cd
#firewall-cmd --add-services=dns --permanent
#firewall-cmd --reload
#systemctl restart named
#systemctl enable named
Now To check DNS lookup utilities (resolve and check):
#nslookup server.ltim.com
#nslookup 172.22.61.131
#dig server.ltim.com
#dig -x 172.22.61.132

```

To check the conf files:

```

#named-checkconf
#named-checkzone (zone name)

```

Server Steps:

```

#vi /etc/resolve.conf
search ltim.com
nameserver 172.22.61.31
#vi /var/named/ltim.for
Addline:
client.ltim.com. A 172.22.61.32
#vi /var/named/ltim.rev
Addline
32 PTR client.ltim.com.
Now you can dig client also, Forward as well as Backward
#nslookup client.ltim.com
#nslookup 172.22.61.132
#dig client.ltim.com
#dig -x 172.22.61.132

```

Client Steps:

```

#vi /etc/resolve.conf
search ltim.com
nameserver 172.22.61.31

```

Now you can dig from client also, Forward as well as Backward

```
#nslookup client.ltim.com  
#nslookup 172.22.61.132  
#dig client.ltim.com  
#dig -x 172.22.61.132
```

IMPORTANT: vi /etc/nsswitch.conf:
ADD --> hosts: dns files

REDIRECTORS:

- < to redirect input
- > to redirect output
- 2> to redirect error
- | pipe to redirect output of first command to second command
- &> to redirect error and output both

Filter: will filter the output. It won't affect the source file

Ex-

```
#tr 'a-z' 'A-Z' =(convert small case to upper case) get input from keyboard and enter to get changed output  
# tr 'a-z' 'A-Z' <file1 = print file1 in upper case and don't change file  
#ls > listing (direct ls output to listing file)  
# ls Chirag 2> errfile (redirect error to some errfile and save it)  
#ls file1 Chirag &> errfile (since file1 is there so output is err+some output)  
Hence if I used &> it directed both,  
If I used 2> it only directed err to that file
```

Some Important Filters-

- cat
 - Filter the file by rows= head from top and tail from down
 - Ex- To display 11th to 15th line of /etc/passwd
 - Use -n to display with line no
 - **# cat -n /etc/passwd | head -15 | tail -5**
- more
- head
- tail
 - Filter the file by rows= head from top and tail from down
 - Ex- To display 11th to 15th line of /etc/passwd
 - Use -n to display with line no
 - **# cat -n /etc/passwd | head -15 | tail -5**
- cut
 - Filter by columns i.e. fields and characters
 - **#cut -f1 -d: /etc/passwd**
 - -f= fileld number
 - -c= character number, no delimiter
 - -d= delimiter
 - **#cut -f1,3 -d /etc/passwd** prints 1,3 fields
 - **#cut -f1-3 -d /etc/passwd** prints first 3 fields
- tr
 - Translates or deletes corrector
 - Echo hello | tr 'a-z' 'A-Z' = case convert
 - Echo hello | tr -d 'h' =delete all occurrences of h in it
 - Echo hello | tr -s 'l' = delete duplicate occurrence of a char in continuous
- wc = prints newline, word and byte counts for each file
 - **wc file1**
 - l lines
 - w words
 - c bytes
 - m chars
- grep search for string in given data output

- grep "word" <filename>
- -w <word> for w=exact match of word
- ^r for all begining with r
- r\$ for ending with r
- -v for excluding
- -i for ignore case
- Hence to ignore dhoni in a file
 - grep -vi dhoni file1
- Sort
- sed= search and replace
 - -i will replace the source file
 - -n line number
 - -p print
 - **# cat -n /etc/passwd | sed -n '11,15p'**
 - **#sed '1s/hi/hello' file1** (changes hi in line 1 of file1 and print)

#FOR LOOP:

```
#vi file1
#print even no from one to 20
echo "print even no from one to 20:"
for((i=2;i<=20;i+2))
do
    echo $i
done
#cat file1 to check script
#sh to run the script
```

#umask: give default permission for files and directories

Read from ppt shared
#whoami
#id
#umask
 0022 for root
#mkdir /d1
#ls -ld /d1
 777-022-755

#TROUBLESHOOTING: from file shared

Filesystem troubleshooting:

In case there is a problem with filesystem or /etc/fstab

In Maintenance mode,

give root password

#mount -a = to get the faulty line in /etc/fstab

Check the faulty line in etc/fstab as follows:

#mount -o remount,rw /

#vi /etc/fstab

Comment or delete or correct the faulty line in /etc/fstab

Ctrl +d to reboot

For advanced troubleshooting: Go to **RESCUE** mode

Ex GRUB:

#grub -install /dev/sda

//EXTRA for learning

Download Linux cheat sheet and learn all basic commands

Key Characteristics

- **Open-Source:**

Linux's open-source model encourages widespread collaboration, rapid development, and extensive peer review, which helps to minimize bugs and improve security.

- **Kernel-Based:**

The Linux kernel serves as the central component of the operating system, managing hardware and resources, and providing a platform for software.

- **Distributions (Distros):**

Various distributions package the Linux kernel with other software to create a complete, user-friendly operating system tailored for different purposes and users.

//go through

Shell - command interpreter- convert High level to machine level

Read shell bash powershell and all

Kernal uses a core and interact with hardware /boot files

Inverted Tree Structure

CDE drive root / windows

Linux on root \ ADMINISTRATOR

Easy to trouble shoot,- modular

Configuration data in **/ETC directory** and can be modified there

H/W- 64bit, 8gb ram, 100gb Harddisk bare minimum-Linux

HCL- Hardware Compatibility List available on redhat website

It is the list of all h/w tested and certified to be working with Linux OS for that version

Install in both GUI, Text

Booting process- steps- **PPT**

Linux basic commands, different types of shells

FileSystems-

RAID- fault tolerance or redundancy Redundant array of independent/inexpensive disks

For organisation to recover data

Suppose main Disk fails, the childs are used to recover

Some for performance,

Others for redundancy- fault tolerance

Some for both

LVM- Resizing Data Disk for losing or expanding it

Package MGMT- Installation s/w

Install the packages for any applications or servers

NAC- Networks Access Control- IP table and now Firewalls

FTP, NFS- network file service for application specific, **Samba**- File and print sharing between Linux and Windows OS
Troubleshooting

BIND-Barcley internet name daemon used in **DNS(IP to host name)**

Service that runs continuously in the background for particular service or server

Linux Apache Webserver in Linux

Squid- Web Proxy server

VVIMP-

LVM, NFS server, Partitioning, Mounting, SSH, SCP,

Install Packages

~RAID~

[//includehelp.com/mcq/linux-mcq.aspx](http://includehelp.com/mcq/linux-mcq.aspx)