

Linux

OS → Interface b/w user and hardware
(or)

Collection of programs that coordinates the operation of computer hardware & software.

→ It allocates system resources to users

→ Linux is a multiuser command line OS

Features of Linux:

1) Open source (along with source code)

2) Multiuser & multitasking

3) Enhanced security (Inbuild firewalls)

No virus

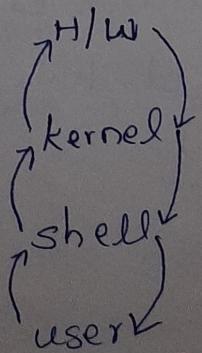
No Downtime

4) Reliability

5) GUI (graphical user interface)

↓
(A visual way of interacting with a computer using items such as windows, icons and menus, used by most modern O.S.)

Structure:



Shell → Interface b/w user & kernel

→ Executes commands

Shell types → 1) Bourn shell

2) Korn shell (AIX)

3) C shell

4) Bash shell (Linux)

5) Z shell

File structure:

/ (root)

→ bin directory (Normal user commands) Ex: ls, cat, rm, mv
(binary)

→ sbin (Admin commands) Ex: useradd, mount, shutdown
(system binary)

→ usr (manual pages) (Every command information)

→ etc (configuration files)

→ opt (3rd party SW) Ex: oracle, chef, java

→ var (log files, mails, messages)

→ home (Normal user home directory)

→ root (root user home)

→ dev (logical devices information) Ex: partition

→ media (removable devices) Ex: cd, dvd, pendrive

→ mnt (temporary mount point)

→ tmp (to download or share to other)
→ boot (kernel files)

Commands

pwd → present working directory
cd → change directory
/ → slash directory.
ls → to configure files Ex:- ls /opt

Linux Files

- 1) Regular files Ex: text files, directories
- 2) Special files Ex: devices, videos.

Commands to create file

- 1) cat
- 2) touch
- 3) vi

1) cat > (file)

Ex: cat > sample

this is my first file using cat cmd in linux basic

ctrl+d (save & quit)

cat filename → to view the content in the file

cat >> filename → to add content to the file
(to append a file)

using cat we can't modify a file

Through vi we can modify.

Touch file is to create empty file

→ touch demo10 demo11
cat demo10

→ vi sample10 (to create a file using vi)

→ Esc i (to insert text in vi)

→ Esc :wq! (to save and quit)

w means save

q quit

! force

→ Esc x → to delete a character

→ Escyw → to copy a word

→ Esc yl → to copy character

→ Esc yy → to copy a line

→ Esc dd → to delete a line

→ Esc dw → to delete a word

→ Esc p → to paste

→ Esc R → Replace line

→ Esc G → to move cursor last line

→ Esc \$ → to move cursor to ending char of line

→ Esc A → to move cursor to starting char of line

→ Esc:/word → to move cursor to particular word
in a container

`Esc : $s /oldword /newword/g` } to replace a word
\$ → starting of the file } in the whole file
\$ → ending of the file

`Esc : 10$ / old word {new word /g }` to replace a word in single line Ex: line 10

ls → to display the content in the file in alphabetical order

`ls -s` → to display the content in the file according to size

\perp Block \rightarrow 2MB

Is $-a$ → to display hidden files

`mkdir -sush` → to create hidden directory

Is .sash → hidden file

ls -l → long list of the content

- textfile

d directory

l linked

`ls -ld filename` → to view the particular file

`mv sample sample10` → to rename a file

`rm filename` → to remove a file

`rm -rf directory name` → to delete a directory.

`rm -f demo11 demo12` → to delete forcibly.

`mkdir -p devops/linux/basics/example`

to create a directory recursively (inside one directory another directory)

`cd` → change directory

`cd..` → to change one directory back

`cd.../..` → to change to two directories back

`cd` → to go to home directory.

`tree` → `tree directoryname`

to list the content in tree format

Filter commands

1) Simple filter commands

→ `head` (to view first lines of a file) default 10 line

→ `tail`

→ `wc`

→ `cut`

→ `diff`

→ `cmp`

→ `uniq`

→ `sort`

→ `less`

→ `more`

→ `tr`

2) Advance filter commands

→ `grep`

→ `sed`

→ find

→ awk

head

head filename

head -2 filename → to view first two line

tail

tail filename

* {tail /var/log/messages → to view the recently updated log files}

wc (word count)

wc filename

66 308 1584 filename
↓ ↓ ↓
line word char

wc -w filename → word count

wc -l filename → line count

wc -c filename → character count

cut

to cut and display the range of characters

{cut -c5 /etc/passwd → 5th char}

* {cut -c5-8 /etc/passwd → Range from 5th to 8th char}

{cat /etc/passwd → to view all linux user info}

root: x : 0 : 0 : root : / root : /bin/bash
 ↓ ↓ ↓ ↑
 user name password Group ID comment
 info home directory

{ user add testuser
 { user add username → to add user }

• cut -f1, -d ":" /etc/passwd → to display first field
 - separator

d delimiter

cut -f1,3 -d ":" /etc/passwd

→ cut -f1 -d ":" /etc/passwd | wc -l

User count

{ useradd -u <uid> -g <gid> -c <count> -s <shell>
 -d "<home dir> /home/user"

→ diff sample sample 2

→ sdiff (side by side)

→ cmp (compares two files char by char)

cmp sample sample 2

→ uniq filename (to eliminate duplicated lines and displays content)

* { uniq -d filename (to display duplicated lines)

 uniq -D filename (How many time duplicated)

→ tr (to convert small letters into capitals)

tr '[a-z]' '[A-Z]' < filename

→ sort (to get content in the alphabetical order)

sort filename

→ more & less (to scroll)

more filename

space → page by page

enter → line

f - next page

b - back page

q - quit

Advance filter commands

→ grep (to search particular word or line which was frequently used)

↪ global regular expression.

word/string or pattern wise search

→ grep word filename
(or)

grep -w 'user' filename

grep -w 'user' /etc/passwd

grep 'user' /etc/passwd

} to search a word

→ grep -i 'root' /etc/passwd } to ignore case sensitive
grep -n 'root' /etc/passwd } to display along with the word number.
line

grep -c 'root' /etc/passwd } → No. of lines

grep -v 'root' /etc/passwd } excluding the word
contains lines it will display

grep ^ 'root' /etc/passwd } → to display lines starting
with this words

grep \$ 'root' /etc/passwd } → to display

grep "root" \$ /etc/passwd } → to display lines ending
with this words

grep '^\$' /etc/passwd → to display empty lines

grep -n '^\$' sample → to display ^{empty} line numbers

How to search multiple words using grep

sed → stream editor

to display the selected line or to delete or to replace on o/p

sed -n '1p' sample → to print particular line

sed -n '1,20p' sample → to print range of lines

sed -n '1p enter 3p' filename → to print selected lines

{
 sed -n '1p'
 > 3p
 > 10p
 > 100p' filename

sed '1d' filename → to display excluding first line
d → delete

sed '1,5d' filename → to display ~~and~~ delete range of lines

sed '1,5d' filename > filename

`sed 's/Last/first/g'` filename → last word ~~sabs~~
replaces with first

How to replace URLs using sed commands

find → searching purpose

Ex: `find /root -name sample`

{ `find /root -name *.xml`

`find /root -type` (l)

l → linked file

f → file

d → directory

`find /root -size 100M` → to search with size

`find /root -size +100M` → above 100

-100M → below

owner
group
others

← { `find /root -perm 654`

ch mod 654

read - 4 r
write - 2 w
execute - 1 x

`find /root -ctime 2` → creation

`find /root -mtime 2` → modified

`find /root -atime 2` → access

days ago

`find /root -cmin 2` } minutes ago

mmin 2

a min 2

• find /tmp -name core -type f -print | xargs /bin/rm-f
to find particular file and takes action

awk → to print or to print cat

awk -F ":" '{print \$1}' </etc/passwd
↓
first field

awk -F ":" '{print \$1,\$3}' </etc/passwd

{print \$0} → to print all the fields

RPM (Redhat package manager)

to install or to uninstall packages

chefdk (-1.3.4) -1.el6 -x86_64.rpm
↓ ↓ ↓
package name version release
 -rc

rpm -q chef

starting name of the package

→ To know whether this package is already installed or not

rpm -e fullname of the package
↓
erase

`rpm -ivh fullpackage name`

i → install

v → verbose (view on screen)

h → hash progress bar

`rpm -qi package name`

i → information (who created, what time, purpose, etc.)

q → query

`rpm -ql package full name`

l → list of files inside the package

`rpm -qR & package name`

R → Resolver

to know the package dependencies

`rpm -Uvh package name`

to upgrade the package to new version

wget → command to download anything from internet to virtual machine

`wget url`

RPM Redhat, centos, fedro

• rpm → package extension

Ubuntu debium

• deb → package ext.

dpkg -ivh

dpkg -l packagename l → list

dpkg -r " " r → remove

yum (through internet if we want to install packages
we can use yum) (Redhat, centos, fedro)

→ There is no need to download. we can directly install

yum install packagename

Ex: yum install tree

yum remove packagename → to remove

ls /etc/yum.repos.d /

to list the repositories

Base → default repository

yum list installed httpd

package name

→ To know whether the package is installed or not

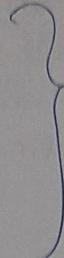
yum update packagename

→ To get latest version

{apt or apt-get} → in ubuntu

service httpd status

↓
Service name



to operate a service

service httpd stop

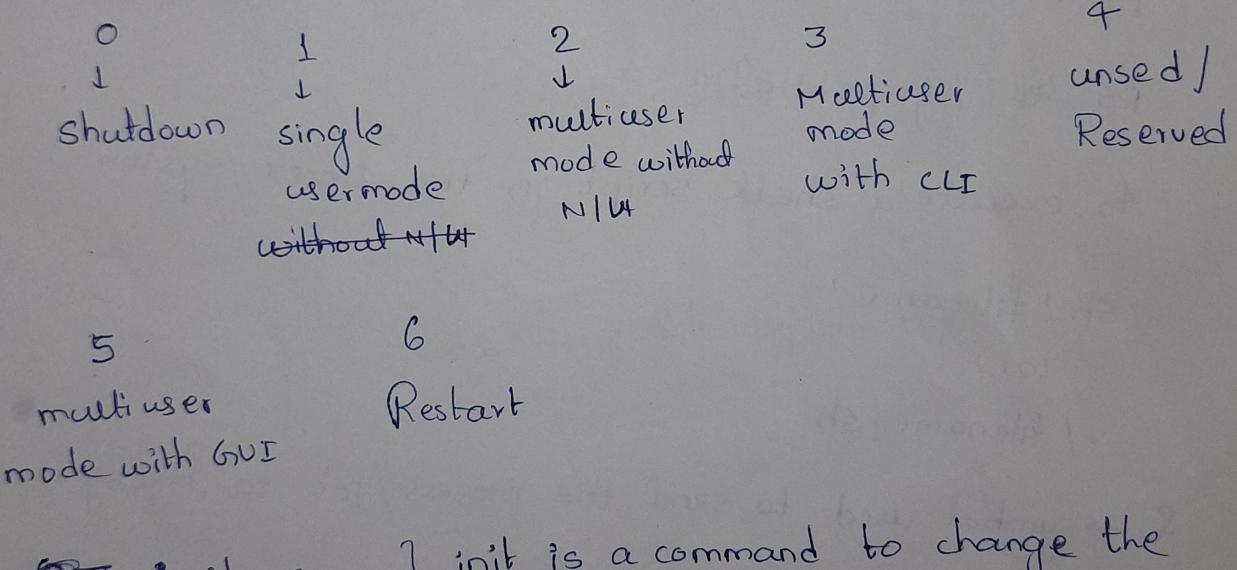
service httpd start

service httpd restart

Systemctl status httpd -

chkconfig --list → list the existing services
to enable all services

Linux have 7 runlevels



init 0 } init is a command to change the
init 2 } runlevel

chkconfig -level 35 httpd on → to on Runlevels

chkconfig --list httpd

ps -ef

↓
command to know the process running on our system.

→ to list all process information

UID	PID	PPID	C	STIME	TTY	TIME	CRID
+ which user	+ process ID	+ parent process ID	+ priority	+ Start time	+ terminal	time takes for execution	command

ps -ef | grep httpd

→ to list ~~pro~~ service related process

kill -9 3213 → ID, to kill the process
 ↓
 signal

pkill -9 httpd

ps -ef | grep httpd

to execute a shell script in the linux system

sh filename

(or)

./filename.sh

& → symbol used to send the process/job to background

• ./sample.sh &

bg → to list the background jobs

fg [number] → to bring it foreground
 ↓

every job has a particular number

cat /etc/redhat-release } to know which operating
 cat /etc/issue } system we are using

rsh → remote system command to stop and start services
in another system

lsb release -a → ubuntu (to know the o.s.)

uname -m → to know [machine type 32-bit
64-bit]

uname -r → kernel release (version)

uname -a → Architecture (what time it is installed,
version, etc.)

ifconfig → to know in IP address of a system.

(eth0 (or) eth1) N/W interface card (NIC)

⑩ loop back (local, self pinging purpose)

linking

links → to attach one file to another file

1) Hard link

2) Soft link

Hard link :-

→ To create hard link ln <source> <link file>

→ if we update any one of the file both will update

→ permissions, size and inode numbers are same for
both files

→ if we remove source file we can access link file

→ this for files this is only for the files

rsh → remote system command to stop and start services
in another system

lsb release -a → ubuntu (to know the o.s.)

uname -m → to know [machine type 32-bit
64-bit]

uname -r → kernel release (version)

uname -a → Architecture (at what time it is installed,
version, etc.)

ifconfig → to know IP address of a system.

(eth0 (or) eth1) N/W interface card (NIC)

⑩ loop back (local, self pinging purpose)

linking

links → to attach one file to another file

1) Hard link

2) Soft link

Hard link :-

→ To create hard link ln <source> <link file>

→ if we update any one of the file both will update

→ permissions, size and inode numbers are same for
both files

→ if we remove source file we can access link file

→ ~~this for files~~ this is only for the files

To create link

In  sample2 sample2_link
↓ ↓
Source link fine name
file

ls -i → to list inode number
↓
kernel understandable format

Ex: ls -li sample2 sample2.lnk (list of files with
inode numbers)

im -of sample2 } If

soft link

→ to create soft link ln -s <source><link file>

→ if we update any one of the file both will update

→ permissions, size and inode numbers are different

→ If we remove source file we can't access

softlinks are used to link the ^{link files}
^{↓ version files}
^{any}.

netsat -r → which route is connected to system

`netstat -natpl | grep httpd` → to know the service related running port numbers