Linux Practical

Saturday, August 10, 2024 05:32 PM

Batch: B-34

Day 1

Linux Labs

Labex.io

sudo -i	To switch to root
ls	Listing directories content
touch	To create empty files
mkdir	To create directories
pwd	To check present working directories
cd	To change directories
cd ..	To change directories 'One step backwardly'
ls dir1	To see contents inside a folder

Creating files and folders

```
touch godzilla
mkdir nikita
```

Specifying range

```
touch file{1..10}.txt
```

Day 2

```
Hostnamectl set-hostname mycomputer
(redhat command)
```

Copy multiple files

```
Cp file1.txt fil2.cpp file3.js
godzilla.txt dir1
```

Remove Multiple files

```
Rm file1.php file2.js file3.c
```

Remove the folder along with its files

```
Rm -r prachi
```

Wildcard

`Rm *.txt`	Remove all files ending with .txt extension

`Mv hello.txt rajpal.txt`	Here mv act as renamer

```
Cp dummy.txt /root/test1
```

Hostname	See or set hostname
Cp	Copy files
Mv	Move files
Rm	Remove files
Rmdir	Removes empty directories

Uname	Prints system information
History	Displays history
Tree	Displays tree
Free	Displays free memory (Ram)
Date	Shows and set date
Cal	Shows calender
Mv	Rename files and folders
Ls -a	Show all files including hidden

Copy command copies original file to destination location but not remove it from original location

Move command removes original file and move it to desired location

Examples

`Uname -r`	Display kernel version
`History -c`	Clears all history
`History -d 31`	Delete item from history
`Free -h`	Human readable format

Day 3

Nano Shortcuts

Ctrl + s	Saves the content
Ctrl + x	Exits the editor
Ctrl + k	Cuts the line
Ctrl + u	Pastes the line
Ctrl + \	Replace word
Ctrl + w	Search word
Alt + u	Undo
Alt + E	Redo
Ctrl + \	Goto
Alt + 6	Copy line
Ctrl + o	Save as

`Ls -l`	Display contents in long list format
`Cat`	Displays file content/Read file content
`Head`	Displays first 10 lines of file
`Tail`	Displays last 10 lines of file
`Nano`	File editor
`Vi`	File editor

Examples

`Ll`	Shortcut of ls -l
`Head -3 filename`	Display starting 3 lines
`Tail -3 filename`	Displays last 3 lines

Cat	/etc/ssh/ssh_config

Command --help See all options
Ls --help
History --help

Nano	Vim
Easy to use	Powerful and challenging to use
Modeless editor	Mode based
What you see what you get	-
Improved version of pico	Improved version of vi
Good for newcomers	Suitable for advanced users

Day 4

Understanding Vim

Vim Keys Navigation

k	Moves cursor up
j	Moves cursor down
h	Moves cursor left
l	Moves cursor right

Basic Commands

:q	Exit from the file
:wq	Save and exit
:q!	Exit without saving
:$	Move cursor to end of file
:w	Save changes
:set number	To show line numbers
:set no number	Hide line numbers

Vim Keybinds

yy	Yank the line (copy)
Dd	Delete the line
Dw	Delete the word
Yw	Copy the word
P	Paste the copied content
U	Undo
Ctrl + r	Redo
Gg	Move the cursor to start of file
G	Move the cursor to end of file

Vim Modes

i	Insert mode
Esc	Escape mode
:	Command mode

Vim insert modes

a	Insert cursor at right of current character
A	Insert at the end of line
o	Insert mode at new line below
O	Insert mode at new line above
r	To change the single character
R	Start Replacing everything

Word substitution

`:%s/wordabc/wordxyz`

Day 5

User Management

User Entry in various files

/etc/passwd	Users information
/etc/shadow	User password information
/etc/group	group information
/home/	List users (dir)

User Commands

Useradd	Create new user
Userdel	Remove user
Usermod	Modify user
Su - <username>	Switch user

Account Management

passwd <username> To **change the passwd** of user or root	passwd naveen
groupadd <groupname> adds a **new group**	groupadd newgroup1
userdel <username> **delete the user** from system but the user home directory doesn't deleted	userdel naveen
userdel -r <username> delete the user **along with home directory**	userdel -r naveen

User IDs

0	root
1 - 99	Pre-defined account
100 - 999	System users
1000+	Reserved for standard users

Types of Account

Super User	Root
System User	One with admin rights
Standard User	One without admin rights

Changing user information

usermod -u <new uid> <username> To **change the UID** of the user	Usermod -u 4001 naveen
usermod -c "comment" <username> To **add a comment** on the user	usermod -c "some comment" naveen
usermod -d <new_homedir> <username> To change the **default home directory** location	usermod -d /mnt/nvin naveen
usermod -s "<new_shell>" <username> To change the **default shell** of a user	usermod -s "/bin/sh" naveen
usermod -L <username> To **lock the user** from login (remote)	usermod -L naveen
usermod -U <username> To **unlock the user** from login	usermod -U naveen

Cat /etc/passwd

rajpal : x : 1000 : 1000 : rajpal : /home/rajpal : /bin/bash

User | encryption | user id | groupid | group | home directory | shell

Adding user into particular group

usermod -a -G <groupname> <username>	usermod -a -G devops naveen

Force to assign a group as a primary

usermod -g <groupname> <username>	Usermod -g newgroup1 naveen

Day 6

Setting Path

Set this env var in .bash_profile or .bashrc

```
$PATH=$PATH:$HOME/.local/bin:$HOME/bin:/tmp
```

Skeleton Files

`/etc/skel - skeleton directory`

Contains Basics Structure
Of Home directory
If any files created here shall get
copied to new users home

`Ls -la /etc/skel`

.Bashrc - execute at every launch of bash shell
.Bash_profile - execute at only time of login

```
ls -la /etc/skel/
total 24
drwxr-xr-x.  2 root root   62 Apr 11  2018 .
drwxr-xr-x. 77 root root 2880 Mar 28 03:38 ..
-rw-r--r--.  1 root root   18 May 30 17:07 .bash_logout
-rw-r--r--.  1 root root  193 May 30 17:07 .bash_profile
-rw-r--r--.  1 root root  231 May 30 17:07 .bashrc
```

File Permissions

Chmod 462 file1.txt

Users	Group	Others
Rwx= 7	Rwx=7	Rwx=7

R -> read -> 4
W -> write -> 2
X - > exec - > 1

Total = 7

Commands to change Permissions

`chmod`	Change file permission
`chown`	Change file ownership
`chgrp`	Change file group

Default Permission of
File: 644
Dir: 755

```
chmod u+r,g+x,o+r file1.txt
```

U	User	+	add
G	Group	-	remove
O	Other	=	replace

```
drwxr-xr-x. 1 root root 0 Sep  3 03:20 testdir
```

Note: Execute permission is automatically assigned to Directory after creating it. (To enable user to enter into directory its equal to execute action)

```
useradd -aG myuser mygroup
useradd -g myuser primarygroup

chmod 466 file1.txt
chown root:mygroup file1.txt
chgrp mygroup file1.txt
```

Day 7

File Types:

Regular File	-
Directory	D
Symbolic Link	L
Character Device	C
Block device	B
Named pipe fifo	p
Socket file	s

System Commands

Cat /proc/cpuinfo	Shows cpu info
Lscpu	Detailed cpu info
Lsmem	Shows memory info
Cat /proc/meminfo	Shows memory info
Ip addr show or ip a	Shows ip address info
Ifconfig	Shows detailed network info
Hostname -i	Shows ip address of host

To define group administrator

```
gpasswd -A <username> <group>
```

Sort	Sorts the content in a-z listing
who	Shows login user
more	Displays file content in scrolling buffer
less	Displays file content one screenful at time
head	Shows top lines of the file
tail	Shows last bottom lines of file
uniq	Filter out duplicates

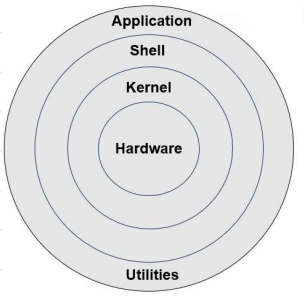
Swap - Virtual Memory used to relocate idle processes from physical RAM to Swap to provide more room in memory of more programs to run

Day 8

File System Hierarchy

Architecture

/bin	Binary Files	Cat, ls, cd, etc
/boot	Bootloader files	kernel files, initrd, initramfs, bootloader
/home	User home directories	User1, user2, etc
/dev	Device files	Stdin, stderr, sda1, sda2, tty
/etc	Configuration files	Httpd, dhcp, emacs, etc
/lib	System libraries	Dkms, dpkg, cups, init
/opt	Add on	Zoom, stremio
/media	Removable device files	Sdc1, sdc2
/sbin	System binaries	Openvpn, arp, alsa
/run	Runlevel files	Lock, lvm, log, mount, initctl
/srv	Service files	Ftp, ssh, httpd
/tmp	Temporary files	Tmpaddon, configerr
/usr	User files	/usr/local, /usr/share
/proc	Process files	Net, tty, kcore
/var	Variable files logs etc	log, cache, lib, opt, run, spool, mail
/sys	System files	Block, bus, class, dev



Hardware	Peripheral devices	Cpu, ram, mouse, keyboard
Kernel	Core of Operating System	Linux, dos, etc
Shell	command line user interface	Bash, zsh, ksh, etc
Applications	Programs run by user	Web browser, file manager, etc
Utilities	System level libraries/utilites	Network manager, disk defragmenter, etc

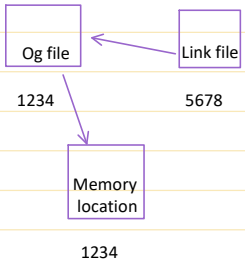
Day 9

Hard Link and Soft Link

Soft Link or Symbolic Link

```
ln -s <original file name> <linkfilename>
ln -s file1.txt filelink.txt

ln -s /root/testdir/file1.txt /root/linkfile.txt
```

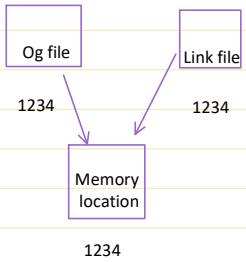
1. If link file edited original file also gets updated
2. If og file deleted link file will not work
3. Removing link file wont delete og file
4. Inode no. of link file is different from og file
5. Link file only point to og file and not its memory location
6. Absolute path must be provided for both argumets



Hard Link

```
ln <og file> <link file name>
ln file1.txt linkfile.txt

ln /root/testdir/ogfile /root/hardlinkfile
```

1. If link file edited original file also gets updated
2. If og file deleted link file still there
3. Inode of both link and og file will be same
4. Link file will direct to actual file in memory location



Umask

The umask acts as a set of permissions that applications cannot set on files. It's a file mode creation mask for processes and cannot be set for directories itself. Most applications would not create files with execute permissions set, so they would have a default of 666, which is then modified by the umask.

How to set umask value
Command:
`Umask 0133`

Default umask: 0022

Max perm: 666
\- Req perm: 644
umask 022

First 0 is for special perm

`Umask 0133`
Now user can create files with limited set of default permissions

Day 10

Find

Find Files more than 1kb size
```
find /var/log -type f -size +1k
```

Get detail info of every files
```
find /var/log -type f -size +1k -exec ls -lih {} \;
```

Without + shows exact files size
```
Find /var/log -type f -size 4k
```

Find files more than 1kb and less than 6kb
```
find /var/log -type f -size +1k -size -6k -exec ls -lh {} \;
```

To search only directories
```
find /var/log -type d -exec ls -lhd {} \;
```

To search files with exact name matching

-type f	To search files
-type d	To search directories

-exec: command output will transfer to exec to execute another command or operation on the output
Pipe |: it run command on the output of another command

Input output redirection

```
Transfer the output to report.txt file
Find /var/log -type f -size +4k > report.txt
```

To search only directories
```
find /var/log -type d -exec ls -lhd {} \;
```

To search files with exact name matching
```
find /var/log -type f -name file.txt
```

To search files with non-case sensitive name
```
Find /var/log -type f -iname FiLe1.txt"
```

Search files based on permission
```
find /var/log -perm 600 -exec ls -lh {} \;
```

Search empty files only
```
Find /var/log -type f -empty
```

To see empty directories
```
find /var/log -type d -empty -exec ls -lhd {} \;
```

To Search files starting from tmp word with wildcard *
```
find /tmp -type f -iname "tmp*"
```

Input output redirection

Transfer the output to report.txt file
```
Find /var/log -type f -size +4k > report.txt
```

Transfer the output to report.txt file and don't overwrite old data and append to it
```
Find /var/log -type f -size +4k >> report.txt
```

>	overwrite the old data
>>	Append the new data
|	Piping the output of cmd1 into cmd2

Using Pipe
To count lines of output
```
find /var/log -type f -size +1k | wc -l
```

Sort	Sorting output alphabetically
Uniq	Removing duplicates

Day 11

Grep

Search content from file
Pattern based matching

1. Search for "thistext" from file1.txt
   `grep thistext file1.txt`
2. For case sensitive (-i, --ignore-case)
   `grep -i someText file1.txt`
3. Count matches
   `Grep -c mango fruitlist.txt`
4. Ignore pattern
   `Grep -v notthistext alltext.txt`
5. Exact word match
   `Grep -w exemplar dictionary.txt`
6. Exact word match but ignore case
   `Grep -wi WeBinar seminars.txt`
7. To print line numbers of matches
   `Grep -n myword allwords.txt`
8. To search in multiple files
   `Grep sometext thisfile alsothisfile`

1. To search multiple keywords
   `Grep -e sample -e learn thisfile.txt`
2. To print only filename of match keyword
   `Grep -l apple cart.txt treelist.txt fruitlist.txt`
3. Search for word starting from pattern
   `Grep ^thebest wordlist.txt`
4. To print the matching line which end with given keyword
   `Grep $ing file.txt`
5. Bulk Search from directory
   `Grep -h sometext directoryname/*`
   `Or dir/{file1,file2,file3}`

Timezone

1. List timezones
   `timedatectl list-timezones`
2. Set timezone
   `timedatectl set-timezone Asia/Kolkata`

Input output redirection

```
echo "hello world" > file1.txt
echo "line 2" >> file1.txt
```

What is Daemon?

A **Daemon** is a computer program that runs in background without user interaction

Daemon Service
```
systemctl status crond.service
```

To restart cron service
```
systemctl restart crond.service
```

To see logs of crons
```
tail -f /var/log/cron
```

Cronetab

*/15 * * * *	Every 15 mins
0 * * * *	Every hour
0 */2 * * *	Every 2 hours
15 2 * * *	At 2:15AM of every day
15 2 * * ?	At 2:15AM of every day
10 9 * * 5	At 9:10AM of every Friday
0 0 * * 0	At midnight of every Sunday
15 2 * * 1L	At 2:15am on the last monday of every
15 0 * * 4#2	At 00:15am on the second thursday of every

crontab -e	Edit or create a crontab file if doesn't already exist.
crontab -l	Display the crontab file.
crontab -r	Remove the crontab file.
crontab -v	Display the last time you edited your crontab file. (non-standard)
Crontab -e -u <user>	Setting crontab for user

```
*  *  *  *  *  /bin/command
```

@reboot	Run once, at system startup (non-standard)
@yearly	Run once every year, "0 0 1 1 *" (non-standard)
@annuall	(same as @yearly) (non-standard)
@monthly	Run once every month, "0 0 1 * *" (non-standard)
@weekly	Run once every week, "0 0 * * 0" (non-standard)
@daily	Run once each day, "0 0 * * *" (non-standard)
@midnigh	(same as @daily) (non-standard)

0 0 * * 0	At midnight of every Sunday
15 2 * * 1L	At 2:15am on the last monday of every
15 0 * * 4#2	At 00:15am on the second thursday of every month
0 0 1 * *	Every 1st of month (monthly)
0 0 1 1 *	Every 1st of january (yearly)

Crontab -e -u <user>	Setting crontab for user

```
*  *  *  *  * /bin/command
T T T T T
|  |  |  |  |
|  |  |  |  └ Day of Week   (0=Sun .. 6=Sat)
|  |  |  └─── Month         (1..12)
|  |  └────── Day of Month  (1..31)
|  └───────── Hour          (0..23)
└──────────── Minute        (0..59)
```

	standard)
@daily	Run once each day, "0 0 * * *" (non-standard)
@midnigh	(same as @daily) (non-standard)
@hourly	Run once an hour, "0 * * * *" (non-standard)

Day 12

File Archiving Formats/File Compressions

Gzip

To create archive
`gzip messages.txt`

Archive file extension
`.gz`

To read archive
`zcat messages.gz`

Compressed 4mb file to 500kb

To decompress archive
`gunzip messages.gz`

Bzip2

To create archive
`bzip2 messages.txt`

Archive file extension
`.bz2`

To read archive file
`bzcat messages.bz2`

Compressed 4mb file to 325kb

To decompress archive
`bunzip messages.bz2`

XZ

To create archive
`xz messages.txt`

Archive file extension
`.xz`

To read file contents
`xzcat messages.xz`

Compressed 4mb file to 194kb

To decompress file
`xz -d messages.xz`

```
-rw-------. 1 root root 4.8M Sep 16 12:37 messages
-rw-------. 1 root root 325K Sep 16 12:37 messages.bz2
-rw-------. 1 root root 503K Sep 16 12:37 messages.gz
-rw-------. 1 root root 194K Sep 16 12:37 messages.xz
```

Gzip	Gun zip	Old gen	Slow compress and decompress
Bzip	Bu zip	Medium gen	Compress fast decom slow
XZ	-	latest	Compr decompr fast

By default these formats can deletes original file after compressing
Use -k to keep files

Tar - Tape Archive

Compression without algorithithm

To create archive
`tar -cvf filename.tar file1.txt file2.txt file3.txt`

Extract into specific directory
`tar -xvf filename.tar -C /root/mydir`

Compression with algorithithm

z	gzip	-cvzf	tar.gz
j	bzip2	-cvjf	tar.bz2
J	xz	-cvJf	tar.xz

```
tar -cvzf newfile.tar.gz file1 file2 file3
tar -cvjf newfile.tar.bz2 file1 file2 file3
tar -cvJf newfile.tar.xz file1 file2 file3
```

To decompress use single command
`tar -xvf filename.tar.gz`

```
tar -cvf filename.tar file1.txt file2.txt file3.txt file4.txt
```
where,
c stands for create
v stands for verbose
f stands for files

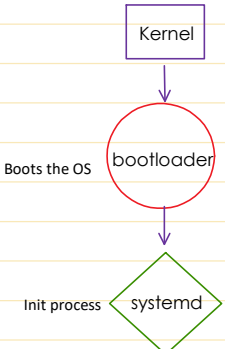
```
-rw-r--r--. 1 root root 325K Sep 16 13:41 tarfiles.tar.bz2
-rw-r--r--. 1 root root 325K Sep 16 13:41 tarfiles.tar.gz
-rw-r--r--. 1 root root 195K Sep 16 13:41 tarfiles.tar.xz
```

To add content into archived file

Day 13

Understanding Processes

First process in linux is always:
Init
Id: 1
Name: Systemd



What is Process?

-> Any Program that is running is process.

Every Process has PID (Process ID)

Parent	Process that created many

Understanding top

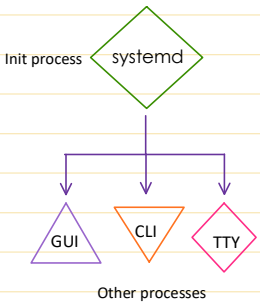
```
top - 15:33:26 up 5 days,  9:16,  3 users,  load average: 0.06, 0.10, 0.43
Tasks: 431 total,   1 running, 423 sleeping,   7 stopped,   0 zombie
%Cpu(s):  0.0 us,  2.8 sy,  0.0 ni, 97.2 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3618.7 total,    401.6 free,   2965.2 used,    563.7 buff/cache
MiB Swap:   4012.0 total,   2931.5 free,   1080.5 used.    653.5 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
      1 root      20   0  175064  13116   6352 S   0.0   0.4   9:04.44 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:01.94 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
```

%CPU(s) row

-> Any Program that is running is process.

Every Process has PID (Process ID)

Parent Process	Process that created many process
Child Process	Process spawn by any process before it



%CPU(s) row

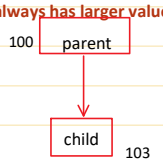
us, user	time running un-niced user processes
sy, system	time running kernel processes
ni, nice	time running niced user processes
id, idle	time spent in the kernel idle handler
wa, IO-wait	time waiting for I/O completion
hi	time spent servicing hardware interrupts
si	time spent servicing software interrupts
st	time stolen from this vm by the hypervisor

To check parent process id
`echo $PPID`

Current shell process id
`echo $$`

To see current running shell
`echo $SHELL`

Child process always has larger value than parent process



PID	process id
PPID	parent process id

Top Keybinds

key	Keybinds
Left	alt + h
Down	alt + j
Up	alt + k
Right	alt + l
Home	alt + ctrl + h
PgDn	alt + ctrl + j
PgUp	alt + ctrl + k
End	alt + ctrl + l

Process Status

D = uninterruptible sleep
I = idle
R = running
S = sleeping
T = stopped by job control signal
t = stopped by debugger during trace
Z = zombie

How to create process?

Creating process
`Sleep 6 &`

To see its id
`Ps -C sleep`

&	Sends process in background
Ctrl + z	Suspends the forground process
Ctrl + c	Terminates the forground process
Fg 3	Brings the background process into forground
Bg 3	To start the suspended process in background

How to find process?

`ps -ef | grep sleep`

Find all process IDs of sleep
`Pgrep sleep`

Find PID value
`Pidof sleep`

To see all running userspace processes
`jobs`

How to kill a process?

Kill the process
`kill 1569`

Kill all instance of process
`killall sleep`

Kill processes from its PIDs
`kill 1253 2212 5522`

Exit Codes:

0	Successful exit without errors
1	Exited with error
2	Permission denied error or missing keyword
126	Script not having executable permissiScript not having executable permission
127	Command not found error
130	Command suspended with ctrl + c
255	User didn't gave flags or option to command

Sigkills - Killing Signals

`kill -l` - List all signals

1. SIGHUP --> reread configuration file of specific program
`kill -1 <PID>`

2. SIGTERM --> stopping the process (terminating) but can be restarted later
`kill -15 <PID>`

3. SIGKILL --> killing the process for good
`kill -9 <PID>`

4. SIGSTOP stopping
`kill -19 <PID>`

5. SIGCONT starting
`kill -18 <PID>`

Bg	Starting process in background
Fg	Bringing background process in foreground
Ctrl+z	Suspending process (freezing it)

Day 14

Nice Value

Determines which userspace process will consume how much ram and cpu

`Nice` command
Create new process with given priority value

Niceness Range from
-20 to 19

Lesser the value	More priority
Greater the value	Less priority

More nice the process
Less resource it will consume and vice versa

Renice Lab
stress -c 3
It will spawn 3 stress process

Set nice value with renice command on one of the stress process by pid value

Renice -n 10 2321

Now recheck output of top

To see nice value
`Ps -efl`

To create new process to give priority

Only root user can set -20 to 19 niceness
Normal user can lower the priority from 0 to 19 niceness
But with sudo powers he can

Priority Value

0 highest priority

Lowest value	Highest priority

System Users

Without human interaction manages linux in background

eg. Apache, user, nginx, user, etc

```
mail:x:8:12:mail:/var/spool/mail:/usr/sbin/nologin
operator:x:11:0:operator:/root:/usr/sbin/nologin
games:x:12:100:games:/usr/games:/usr/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/usr/sbin/nologin
nobody:x:65534:65534:Kernel Overflow
User:/:/usr/sbin/nologin
dbus:x:81:81:System Message Bus:/:/usr/sbin/nologin
tss:x:59:59:Account used for TPM access:/:/usr/sbin/nologin
systemd-network:x:192:192:systemd Network
Management:/:/usr/sbin/nologin
```

Sudoers file

Enable Normal user to run admin command

Edit using visudo command

Give permission to specific command

To see nice value
`Ps -efl`

To create new process to give priority
`Nice -n -20 sleep 2m &`

Renice

Change the existing priority of process

To assign nice value to existing process
`Renice -n 10 2357`

0 highest priority

Lowest value	Highest priority
Highest value	Lowest priority

20 0 default values
Medium or normal cpu allocation

39 lowest priority

Edit using visudo command

Give permission to specific command

1. Allow shyam to run various commands:
   `shyam ALL=/sbin/halt, /bin/kill, /etc/init.d/httpd`
2. Allow user ram to run /sbin/halt without any password
   `Ram ALL= NOPASSWD: /sbin/halt`
3. Allow user janhvi to run any command from /usr/bin directory on the system dev02:
   `Jahnvi dev02 = /usr/bin/*`