

To connect with internet Through Terminal:

mmcli.com up emp053 ←
 ↑ zero

YUM PACKAGE:

yum stands for
 y → yellow log
 u → update
 M → Modifier

yum command is used to update, install, remove Packages.

- 1) To update: yum update Package Name
- 2) To install: yum install Package Name
- 3) To remove: yum remove Package Name

[NOTE: To install a Package,
 firstly download in PC,
 using firefox. Then copy "complete file name".

Then in Terminal change directory. To where
 The downloaded file is stored.

yum install.]

Then we used

To go as root user:

5V ← (or SUDO)

Password: ←

Password is typed, but
 doesn't show

To exit from root mode :

ctrl + d ↪

To create Directories in LINUX :

mkdir Name of Directory ↪

To show The Directories :

ls ↪

Or

ls -l ↪

To change Directory :

cd directory name ↪

cd .. ↪ # it helps to come out
space of a Directory.

To create a file :

touch filename ↪

eg: touch lab.tut ↪

To Enter data in a file :

nano filename ↪

To Save The file :

ctrl + x ↪

To see The content of file:

cat file-name ↪

To clear The whole Terminal:

clear ↪

To come out of some directory:

cd .. ↪ Or cd \ ↪
space space

To remove directory/File:

rm directory-Name ↪

To get details about harddrive:

fdisk ↪

Or

fdisk -l ↪

To know size of a specific Partitions

Step 1: fdisk -l ↪

Step 2: fdisk /dev/sda

↓
file
name ↓ file
name

Step 3: Then press m for help(commands list)

Step 4: -h

Step 5: Partition number 3

To copy content of one file into Another:

`cp file1 file2 ↵`

To copy The data of file1 into file2.

To cut Paste from one file into another:

`MV file1 file2 ↵`

To display every user logged in computer systems

`who ↵`

To display active user:

`whoami ↵`

To show calendar of current month:

`Cal ↵`

for calendar of a specific Month.

`Cal 5 2020 ↵`

month year

for whom we want to see.

To display date :

`Date ↵`

To create directories within directories:

mkdir -p dir1/dir2/dir3 ↵

it would create three directories dir1 & in it,
if we create dir2.
And in dir2 it would
create dir3.

if we want to find something about a command.

what is command-name ↵

e.g. what is sudo ↵

Head And Tail Command:

if we have a big length of Text file and
we need top 10 lines,

Then we use head command.

And Tail command returns last 10 lines.

Syntax:

head filename ↵

Tail filename ↵

To find difference we have three commands:

- diff → diff file1 file2 ↵
- cmp → cmp file1 file2 ↵
- comm → comm file1 file2 ↵

To sort the elements in a file:

sort filename ↪

Vi editor:

To create new file →

vi filename ↪

To enter into insert mode →

i ↪

To save this we need to come out of insert mode.

Esc ↪

↓
(key)

Then To save this file:

Shift + :

After this we'll reach at bottom.

Then we type

wq ↪

To save & exit.

But if we want to save & not exit, Then we press

w ↪

(NOTE: we are outside of Insert Mode.)

To undo change in vi editor:

u ↲

To undo all changes in all line:

U ↲

To open new line:

o ↲

To delete lines:

dd ↲

To delete Three lines:

3dd ↲

replace character:

r ↲

Shift: ZZ → Save The file & quit

Shift: w → Save

Shift: wq → save & quit

Shift: q → quit without saving.

if you want to enter data after cursor:
a ↵

if we want to enter data at the end
l ↵

History Command → Shows history of command

Man Command → it is for system manual
man ↵

Shortcut to enter root →

cd / ↵

(forward slash is used represent root.)

To list various extension of manual command:

MAN -ls ↵

Appropos → it is used to change Time, everything etc.

man appropos ↵

ACL (Access Control List) → controlling access extended permission

- To see the details of User → cat /etc/passwd ↵
- To see Password of user → cat /etc/shadow ↵
- If you want to list out the permission → ls -l
here r, w, rw ,
read write read & write

Permissions:

To see the permission of particular file →

get fact filename ↪

FORMAT:

File : user.txt

File name

owner: Linux

User Name

group: Linux

By default the file is automatically added in group as by default group name is Username

User :: rw

User: Linux: r-

group :: rw-

MASK :: rw-

Other :: r-

#

To set the permission of file →

Set fact -m U : userName : rwx file name ↪
modification ↪ user
read & write ↪ execute

#

To Compress file:

To Compress (zip)

To Uncompress (unzip)

① zip → zip Newfile Oldfile ↪ unzip → filename ↪

② gzip → gzip filename ↪ gunzip → gunzip filename ↪

③ bzip2 → bzip2 filename ↪ bunzip2 → bunzip filename ↪

④ lz → lz filename ↪ lz unlz → lzunlz filename ↪
(BEST METHOD)

These algorithm are based on different algorithm and taking to zip file ↪

Tar Archive (To create Archive of file):

`tar -cuf` file name file1 file2 file3 ...
 ↴ ↴ ↴
 create list of Name of file file
 verbose file Archive
 (Progress). file
 example- abc.tar ↴
 ↴
 ↴ list of files

To extract:

i) `tar -xvf` → `tar -xvf file name` ↴

eg: `tar -xvf abc.tar` ↴

ii) `tar -xvf` → `tar -xvf file name` ↴

eg: `tar -xvf file.name` ↴

To change Permission → `setfc` ↴

To know Permission → `getfc` ↴

To change access Permission of files & directories:

Syntax: `chmod [reference] [operator] [mod] File name` ↴
`ls -l` ↴ # To list out all commands

eg: `rwx-rw-r-- , 1 linuu linuu 7 Nov 30 15:00`
 ↓ ↓
 owner group
 name

To see permission of file or for same:

`getfac file name` ↴

Date: _____
YOUVA

chmod's various attribute:

Reference Description:

v → owner

g → group

o → others

a → all

Operator Description

+ To add

- To remove

Mod Description

r read

w write

e execute

To change permission of group:

Ex - chmod g + rw filename ↳

↓ ↓ ↳
group if we want read
to add write
permissions

To match permission of file:

chmod g - rw filename ↳

eg: chmod v + rwn vm.tut ↳

To check getfacl vm.tut ↳

F.T. E...

chmod (in numeric) →

0 → no permission

1 → execute

2 → write

4 → read

if 3,
That is $1+2$ or $2+1$

This means write & execute

if 5,
That is $4+1$ or $1+4$

This means read + execute

if 6,
That is $4+2$

This means read + write

if 7,
That is $4+2+1$

This means read + write + execute

Syntax: chmod NNN FilenName ↪

owner ↓
group ↓
other ↓

∴ In one command we can give permission of all other.

e.g. ex - chmod 273 abc.txt ↪

Permission ↓
for owner
Permission ↓
for group
Permission ↓
for write & execute

P.T. [] .00

To change owner of file or directory.

Chown →

- 1) Change by name → SUDO chown New User File Name
- 2) Change by id → SUDO chown USER ID File Name
- 3) Add new user → useradd main12
Logout
- 4) If we want new owner for file vm.tut

Syntax → SUDO chown main12 vm.tut
Password:

[Note: To check new owner getfact fileName]

LINUX Filters

To achieve desired output from the files we apply filters.

filters means which is going to be filter.
There are many filters in unix/Linux.

- 1) head → used to display the part of file (By default 10).
we can use '-n' to desired no. of line.

Syntax - \$ head filename

If we want three lines.

\$ head -n filename

[n = 1, 2, 3... and so on for multiple line]

for Multiple files:

\$ head file1 file2 file3 ↵

NOTE: retrieving lines from multiple file is not applicable on tail command.]

tail →

it is used to display last part of file.
(by default 10).

We can use '-n' to display desired no. of lines.

Syntax: \$ tail filename ↵

\$ tail -n filename ↵

pipe (!): it is used to combine two or more commands & in this the output of one file acts as input to the other file.

Syntax →

head -n filename ! tail -n

The output of this file would be treated as input of other command.

e.g: head -8 abc.txt ! tail -3

To add number as prefix in any file

#! nl filename ↵

And if we want to save output in new file

\$ nl b > b2
old filename new filename

5) `sort()`: it is used to sort a file's content arranging records in Particular orders.

Sorting Priority when sorting:

- 1) Numeric order
- 2) Alphabetical order
- 3) Lower case - To upper Case

Syntax

`$ sort filename ↲`

* To sort Multiple files:

`$ sort file1 file2 file3 ↲`

e.g: `$ sort ab` (say with values as

av
ngl
2
3
4
9
v
8
7)

Output: 0

2

3

4

6

8

9

av

ngl

as

av

ds

ngl

6) To reverse content → `$ sort -r filename ↲`

7) To sort file numerically → Ascending order → `$ sort -n filename`
 Descending order → `sort -rn filename`
`$ sort -nr filename`

SE LINUX (Security Enhanced Linux):

it is a Linux Kernel - it provides high security developed by NSA - (National security)

SE Linux kernel security module, That provides mechanism for access control, security Policy & MAC (Mandatory Access control).

* it can run into Three Modes:

- ① **Disabled** → here SE LINUX rules are not applied and your system is at higher risk.
- ② **Permissive** → The file system is labelled & "access denied" entries are logged & no access is actually denied.
- ③ **Enforced** → here security Policies of SE LINUX is in full Mode.

here we have given permission to every entry. But we are keeping log of each entry.

* To see working condition/status of SE LINUX:

getenforce ↪

* To change mode of system:

- ① To do in Permissive mode → 0
- ② To do in Enforced mode → 1

* To do Permissive mode from enforced:

setenforce 0 ↪

To again change into enforced:

setenforce 1 ↪

* To Disable SELinux Mode:

SUDO nano /etc/SELINUX/config ↵
space

it would lead to editor, Then we have to write

SELINUX=disabled

To save → ctrl + x ↵

Scheduling Commands:

To Perform some Task in future. Then we can do
that with help of scheduling commands.

i.e. To Schedule
a task to run at later time.

* Two Commands:

① at

② cron - cron tab

I) at: at command is used to run single time
scheduling for a task.

This Task would close after
single process.

Do me at Commands:

① Command to list out users pending task/job:

at -l ↵

or

at q ↵

(with these two command we can find out all predefined scheduled task)

To Schedule a task:

Syntax: \$ at now ↳ # if we want to execute right now
\$ at now + 5 min ↳ # if we want to execute after 5 min
now enter task say... touch job12.tut ↳ to create file

Then ctrl+D

ctrl+D (To save changes)

Output Job 11 at Mon Jan 2 15:23:00 2023
↓
no. of task

Various formats:

\$ at 1:40 01 02 23 ↳
↑ ↑ ↓
Month date year

go to at mode give the task,

ctrl+D ↳

ctrl+D

③ To delete That Task/ Job:

\$ at 20m JobNo ↳

II. Crome - Crome tab:

on
(Crome Tab!!)

it is a software utility. This automate a scheduled task at a Pre-defined Time. This is daemon process which runs as a background process and perform specified operation at a predefined time.

* Chrome tab :

it is a list of command for executing the scheduled operations at a particular time.

- so it permits →
 - ① editing → \$ crontab -e ↵
 - ② adding → \$ crontab -e ↵
 - ③ removing → \$ crontab -e ↵
 - ④ To list scheduled job → \$ crontab -l ↵
 - ⑤ To delete scheduled job → - \$ crontab -r ↵

To create cron job (or create a task using cron)

*	*	*	*	*	Commandname ↵
mins ↓	hours ↓	date ↓	month ↓	days ↓	
(0-59)	(0-23)	(1-31)	(1-12)	(0-6)	

minutes → (0-59)

hours → (0-23)

date → (1-31)

Month → (1-12)

day → (0-6) (Sunday to Saturday)

e.g: After opening editor

36 15 2 11 Shut down ↵

Press Esc To exit editor

:wq To save it.

Rules for Time/date format →

→ if in min we leave colon symbol, Then it would perform task every other minute & bly 'for hour, date, month, day etc'.

→ for range like (2-5) month, it would perform task in b/w range 2 to 5 in month

→ for particular, like (3,5,6) ← i.e we want to assign some specific months. (or mins, hours, days etc.)

SHELL SCRIPTING:

A SHELL SCRIPT is a computer program designed to be run by UNIX/LINUX cell.

i.e it is a command line interpreter. (it executes line by line).

SOME TYPICAL OPERATION performed by shell script:

- ① file manipulation
- ② program execution
- ③ printing text