

\* CMD challenges  
\* Bandit challenges.  
imB \* Linux Journey → Both theory & practicals.

rm → To remove  
-rf → recursive & forcefully

\* Find data in linux:

# find/-name (filename) -delete.

\* User management & permissions

0/sep  
10/20/20

\* Notes: DAY 1: Linux Directory structure

/ → Mount Point / slash  
Superuser → Root User → Administrator

/sbin → System Binaries.

~tild: H

o Filenames (start with '.') are hidden

files → We can access them by using -a

(a for all)

∴ \$ ls -a

↓ → for all hidden files.

(list of directory)

o -l for long:-

Shows a detailed list of files in a long format.

∴ \$ ls -l

combination [ \$ls -la ]

It will also work.

5) Touch

Used to create new empty file.  
\$ touch filename

6) file:

- To find out what kind of file a file is.  
- It will show you the description of file content.

\$ file banana.jpg

7) cd .. → Previous directory.

8) history

9) cp (copy): To copy



-P → Parent flag

-r, recursive → means d files inside directories within directories.

∴ If we use this with cp (copy) then this will recursively copy the files & directories within directory.

\$ cp filename /home/sourabh/Beeg

↓  
path

ohildcard, '\*'

Used to represent all single characters or any string.

\* mv (move):

for moving / Renaming.

\$ mv oldfile newfile → Renaming

\$ mv files hohhe/pete/Documents

→ file moving

2) mkdir → make directories.

\$ mkdir books/paintings

<http://tryhackme.com/room/linuxfundamentalsports>  
linux hack tryhackme.  
b4shGBMNd72jv2

16) Man: used to display the user manual of any command that we can run on the terminal.  
IMP →

17) Whatis:

gives brief description of command line program.

\$ whatis cat

18) alias: - To make alias

\$ alias foobar='ls -la'

19) cat command: print the content of file.

onto the standard output stream.  
(Also used to create the file.)

\* -P → Parent to child flag

20) wc → used to count total number of lines, words & characters contained in the file.

∴ → To go in previous / specific file.



o ignoredups: To ignore duplicates

\* Which → To find location in binary.

\* Pipe Sign: "|"

To connect two command

o/p → i/p → output of one

# wc -l rbs.com.txt command used as

→ 104 rbs.com.txt i/p for next command

\* wc: word count

o/c -l filename

o -l → Total number of lines

\* grep: To search/find the string.

grep "string we need" filename

Eg: grep "THO" Sourabh

String filename

\* Ifconfig: To get confi. of machine.

\* We can use pipe with grep

o -v [Inverse grep]

We do not want particular that file.

∴ grep -v "inet 6"

↳ It will remove inet 6

grep "inet" | grep -v "inet 6" | grep "127.0.0.1"

It will remove both inet 6 & 127.0.0.1

\* Metachore:-

1 → look for something start with

\$ → look for something end with

Eg cat /etc/group | grep "nsgs"

cat /etc/group | grep "hunt\$"

cat file | grep "A Dheeraj\$"

Both start with used (A Dheeraj\$)

It will search for Dheeraj whole word.

o → > → redirecting

maybakur/s cdac.in > urls.l

↑ redirect o/p to

Imp → Also we use 'tee' for redirecting

In Grep it will not consider as pattern it will use as string. So to avoid we have to use '\'. It will search for pattern.

Eg: `grep "\.doc"`

↑  
search for pattern .do at end of line

• Grep will search in case sensitive manner to avoid use '-i'

↑  
Case insensitive.

∴ `grep -i "\.ZIP"`

↑  
search Both upper & lower case.

`# cat ZIP.txt | grep "\.ZIP$"`

↑  
To check all digit betn 0-9

0-9

\* Predefined classes for Grep:

`[:alpha:]` → Alphabetic char.

`[:digit:]` → Digits

`[:lower:]` lower case

`[:upper:]` upper case.

`[:blank:]` → space / tab

\* awk is programming lang & we use awk as command in linux.

\* awk: separate data into columns based on space & tab.

Eg `awk '{print $2}'`

`awk '{print $2 "\t" $4}'`

`# ifconfig | grep "inet space"`



②  
from Regular to root user → sudo sh  
\$ → Regular  
# → Root use  
from root user to regular user  
su 'username'

- adduser 'username' → To add user.

- To see existing user:

- passwd → command to password for user

- groupadd → command to set password for grp.

/etc/group → If rel. to grp

/etc/passwd → If rel. to user

- for useradd 'name' → Directly added without asking any passwd

- adduser 'name' → Ask for setting the password.

- cat /etc/passwd → show user/list of users

- To delete the user →  
userdel "username"

- To change the password of user:  
sudo passwd "username"

To check the permission:

- ls -ld filename/username  
ls -l

---

\* Group:

- 1) addgroup → groupadd "grpname"  
or addgroup "groupname"

- ②
- `cat /etc/group` → list of groups
  - `less /etc/group` → //
  - To delete group:  
`groupdel "groupname"`

• To add user to group :-

`usermod -a -G "grp" "user name"`  
 ↑ append      ↘ secondary grp

to which we want to add user.

• To add user to multiple groups :-

`usermod -a -G "group1", "group2", "group3" "user"`

Eg: `usermod -a -G g1, g2, u1`

\* Permissions:

4 → Read, 2 → Write, 1 → Execute  
 full → 7, 0 → no permission

chmod: To change/update file access/permissions

Eg: `chmod 777 file.txt/path`

rwX rwX rwX      filename  
 ↓      ↓      ↓  
 user group other

chgrp → To change group ownership

`Sudo chgrp -c Nick file`

↳ We changed the grp ownership to Nick.

③ chown: To change user ownership of file.

`Sudo chown -c nick testfile.`

↑ To show details.      ↓ new user owner      ↓ file

(so from this user "nick" we can make changes to file)

↳ To check ownership

`ls -l filename`

↳ VIM dummy file

↳ Write anything + W + save

↳ `cat dummyfile.`

\* ACL commands :-

`setfacl -m u/g:username:rwX`

- filename

↓ Permission

u → user

g → group

• To check the permission:

`getfacl filename` or

`ls -l filename`

`ls -l /etc/shadow` →

To check permission for user

`ls -l /etc/group`

To check permission for grp

command:

- 1) finger 'username' →  
→ print the information about the user.
- 2) Passwd → To change the Password
- 3) Ps → List of running Processes

\* ls -l "filename/username"

↳ To check the permissions

↳ -rw-rw-r-- root root size  
file user group other user grp  
owner owner size

file → Start With '-'

directory → start With 'd'

/etc/shadow vs /etc/passwd - ?

Process - ? , Examples → ?  
Assignment - ?

tee - ?

12) last log.

su - user

whoami - current user

\* login as user 1 →

login su - username  
as → su - username

15) → ~~to~~ home/user 2

/home#ls

→ show all users

5) run level → who - r



- ① get subjects. ✓  
② add subjects. ✓

③

- 1)  $Su - user1$  to login as user1  
id / whoami → current user
- 2)  $Su -l -u username$  →  
login as user.
- 3)  $Cat > file.txt$  → to  
write anything in file.
- 4) tee — redirect. Always use  
tee to redirect with pipe.  
Eg →  
 $echo "Name" | tee file.txt$   
→  $cat file.txt$  → "Name"
- 5)  $head -5 path/filename$   
↳ To print top 5 line
- 6)  $Tail -5 path/filename$   
↳ To print bottom 5 lines  
⊕ To join both
- 7) Grep syntax  
 $grep "string" filename$   
^ → start with  
\$ → End With
- 8) AWK:-  
 $awk -F ":" '{print $1}'$   
\$NF → count column &  
print last col.