INPT

LINUX FOR ETHICAL HACKING

CIT

MADE BY:

FATIMA EZZAHRA ACHAIT

MOHAMMED KHALDOUNE

INPT

LINUX FOR ETHICAL HACKING

CIT

MADE BY:

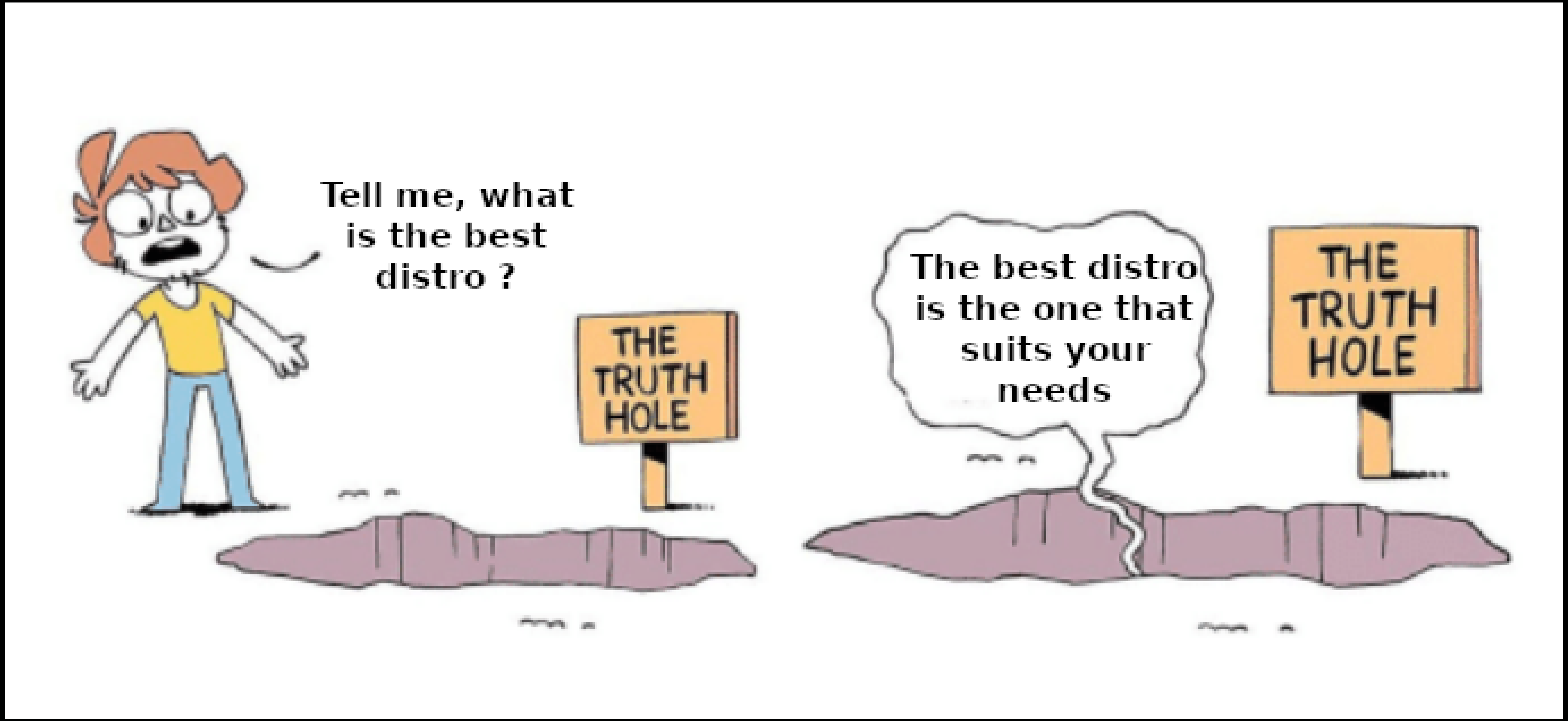
FATIMA EZZAHRA ACHAIT

MOHAMMED KHALDOUNE

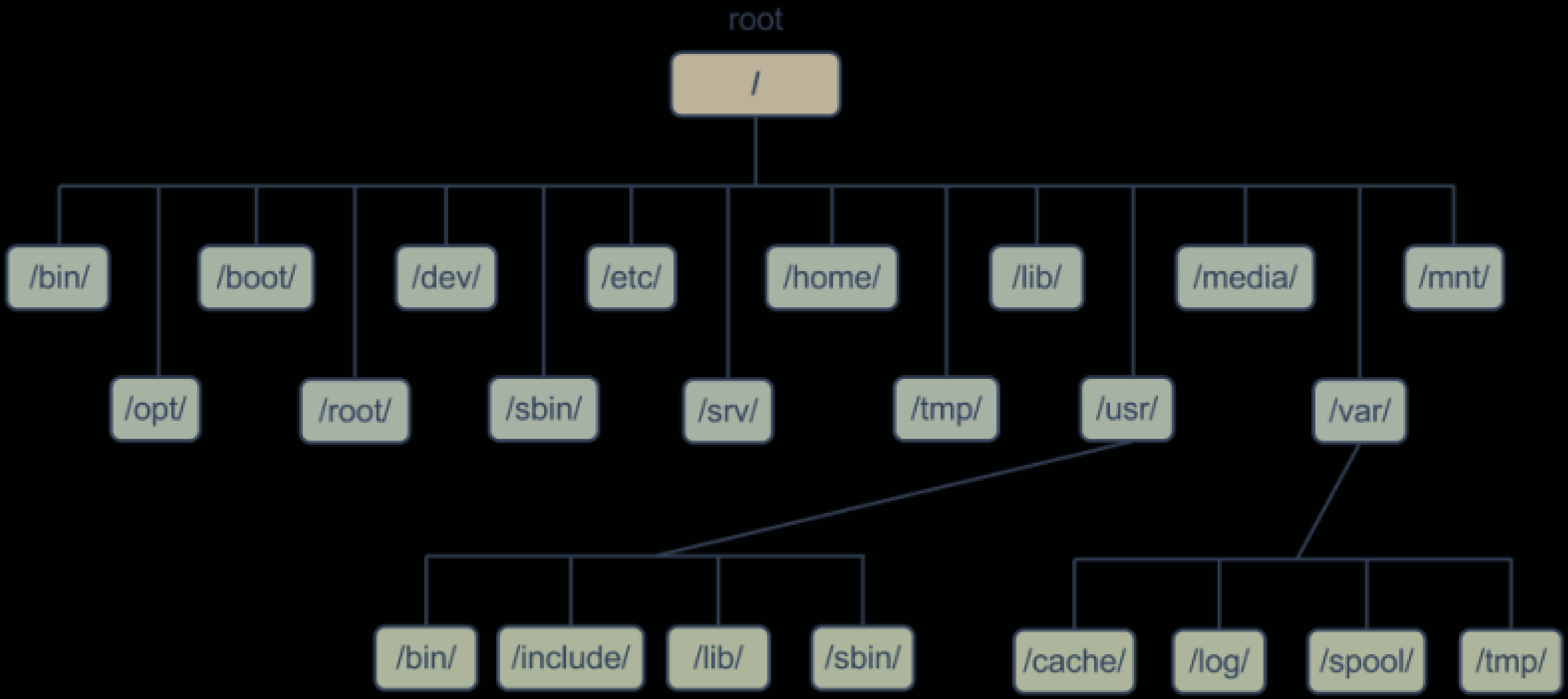
PLAN

1. INTRO TO THE SYSTEM
2. THE IMPORTANCE OF MAN COMMAND
3. NAVIGATING THE SYSTEM
4. FILE VIEWING CREATING & EDITING
5. FILE PERMISSIONS
6. USERS & THEIR PRIVILEGES
7. INSTALLING & UPDATING TOOLS
8. SPECIAL CHARACTERS
9. USEFUL COMMANDS
10. MORE RESOURCES

INTRO TO THE SYSTEM


Tell me, what is the best distro ?
THE TRUTH HOLE
The best distro is the one that suits your needs
THE TRUTH HOLE

INTRO TO THE SYSTEM



THE IMPORTANCE OF MAN COMMAND

man command in Linux is used to display the user manual of any command that we can run on the terminal.

```
--$ man [command]

--$ man man

--$ man ls

--$ man cd
```

Note : you can also use

```
--$ command --help
--$ command -h
```

NAVIGATING THE SYSTEM

pwd stands for Print Working Directory. It prints the path of the working directory, starting from the / .

ls lists the files in the current working directory.

--$ ls -l	list files with more details
--$ ls -a	list all files even hidden ones
--$ ls -al	list all files and hidden ones with more details

cd used to change the current directory.

--$ cd .	dot means the current directory
--$ cd ..	two dots means go back one directory
--$ cd ../..	take me two directories back

CHALLENGE-1

Go Deep & Find The Flag ....

🔗

NAVIGATING THE SYSTEM

Note: absolute path vs relative path

an absolute path specifies the location from the root directory '/ ' whereas relative path is related to the current directory.

locate will print the absolute path of all files and directories that matches the search pattern and for which the user has read permission.

```
--$ locate file.txt              locates file.txt in the system
--$ locate -i readme.txt         the -i option tells locate to run a case-insensitive search
```

FILE VIEWING CREATING & EDITING

mkdir creates a directory

--$ mkdir thisIsMyDirectory	creates a new directory called thisIsMyDirectory
--$ mkdir -p dir1/dir2/dir3	the -p option create non-existent directories in a path

touch creates an empty file

--$ touch thisIsMyFile	creates a new file (empty) called thisIsMyFile
--$ touch file1 file2 file3	can create more than a file in a time

FILE VIEWING CREATING & EDITING

cat It reads data from the file and gives their content as output.

```
--$ cat file.txt
--$ cat /etc/passwd
```

echo outputs the strings that has been passed as arguments

```
--$ echo "Hello World"
--$ echo "Hello World" > file1
--$ echo "Hello World2" > file1
--$ echo "Hello World3" >> file1
```

Note : > overwrites and >> appends.

CHALLENGE-2

Make A File Empty Using echo Without Deleting It ....

FILE VIEWING CREATING & EDITING

cp used to copy files or group of files or directory

```
--$ cp file1 file2
--$ cp -r dir1/ directory1/
```

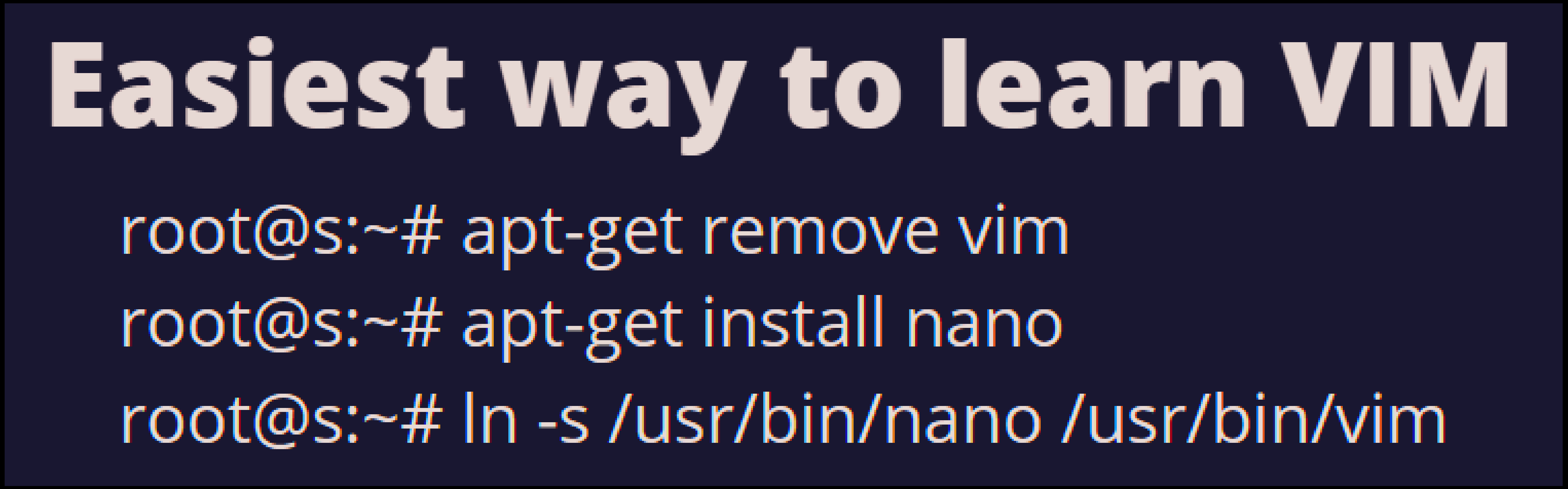
mv moves files or directories from one place to another

```
--$ mv file1 file1.txt
--$ mv file1.txt  /opt
--$ mv dir1 /opt
```

FILE VIEWING CREATING & EDITING

nano is an easy to use command line text editor

```
--$ nano new_filename
```



Easiest way to learn VIM

```
root@s:~# apt-get remove vim
root@s:~# apt-get install nano
root@s:~# ln -s /usr/bin/nano /usr/bin/vim
```

FILE VIEWING CREATING & EDITING

vim a universal text editor that can be incredibly powerful when used properly. From basic text editing to editing of binary files

```
--$ vim new_filename
```

gedit text editor for the GNOME Desktop

```
--$ gedit new_filename
```

Note : to install gedit use the command

```
 --$ sudo apt-get install gedit
```

FILE VIEWING CREATING & EDITING

rmdir removes the directory

```
--$ rmdir emptydir/
```

Note : it removes empty directories only

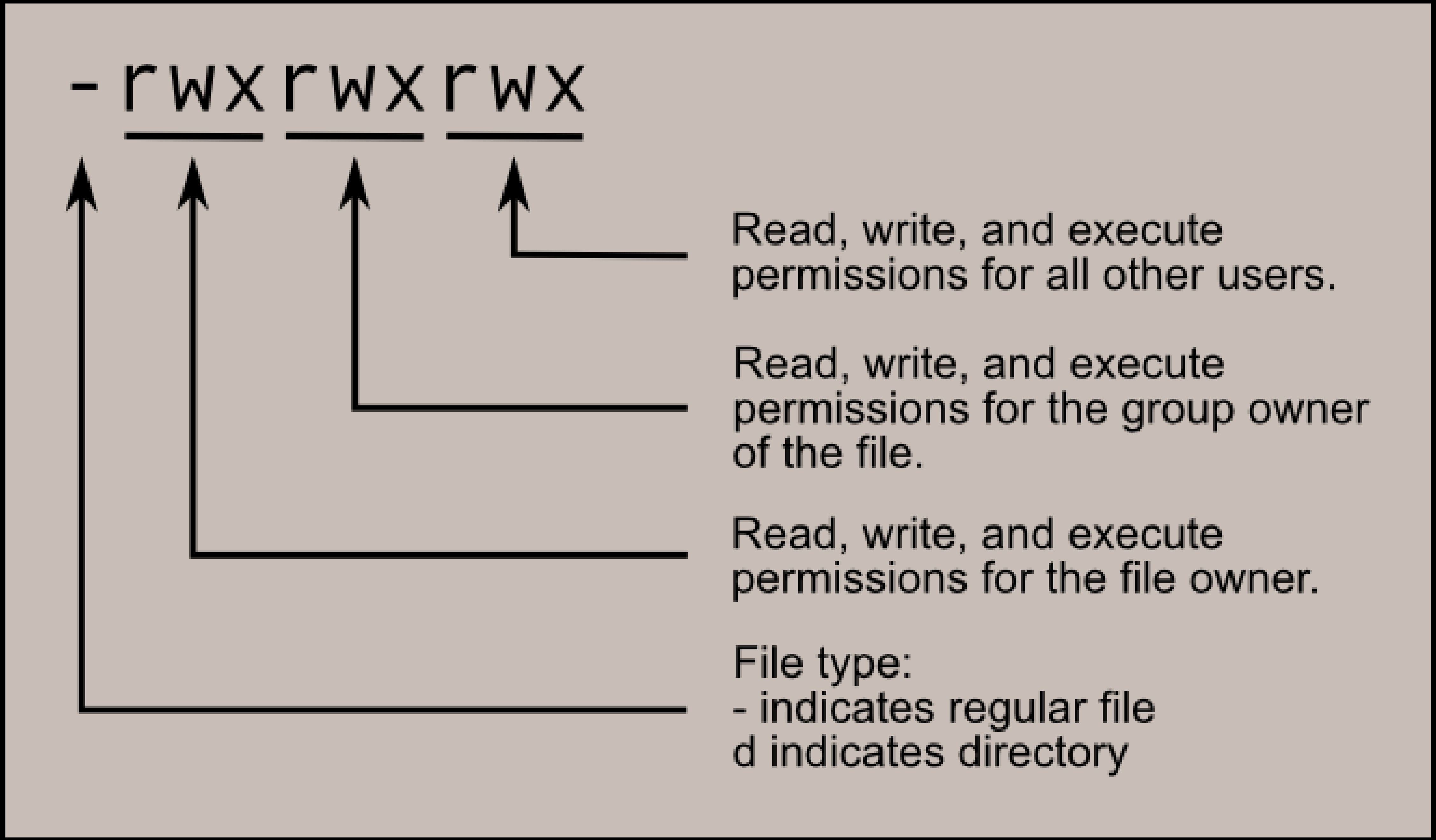
rm delete one or more files or directories

```
--$ rm file1 file2 file3
--$ rm -rf dir1/ dir2/
```

Note : never try this command

FILE PERMISSIONS

File permissions:


-rwxrwxrwx
Read, write, and execute permissions for all other users.
Read, write, and execute permissions for the group owner of the file.
Read, write, and execute permissions for the file owner.
File type:
- indicates regular file
d indicates directory

```
--$ ls -l     list files with their permissions
```

FILE PERMISSIONS

chmod command:

```
--$ chmod u+r g+w o+x somefile
```

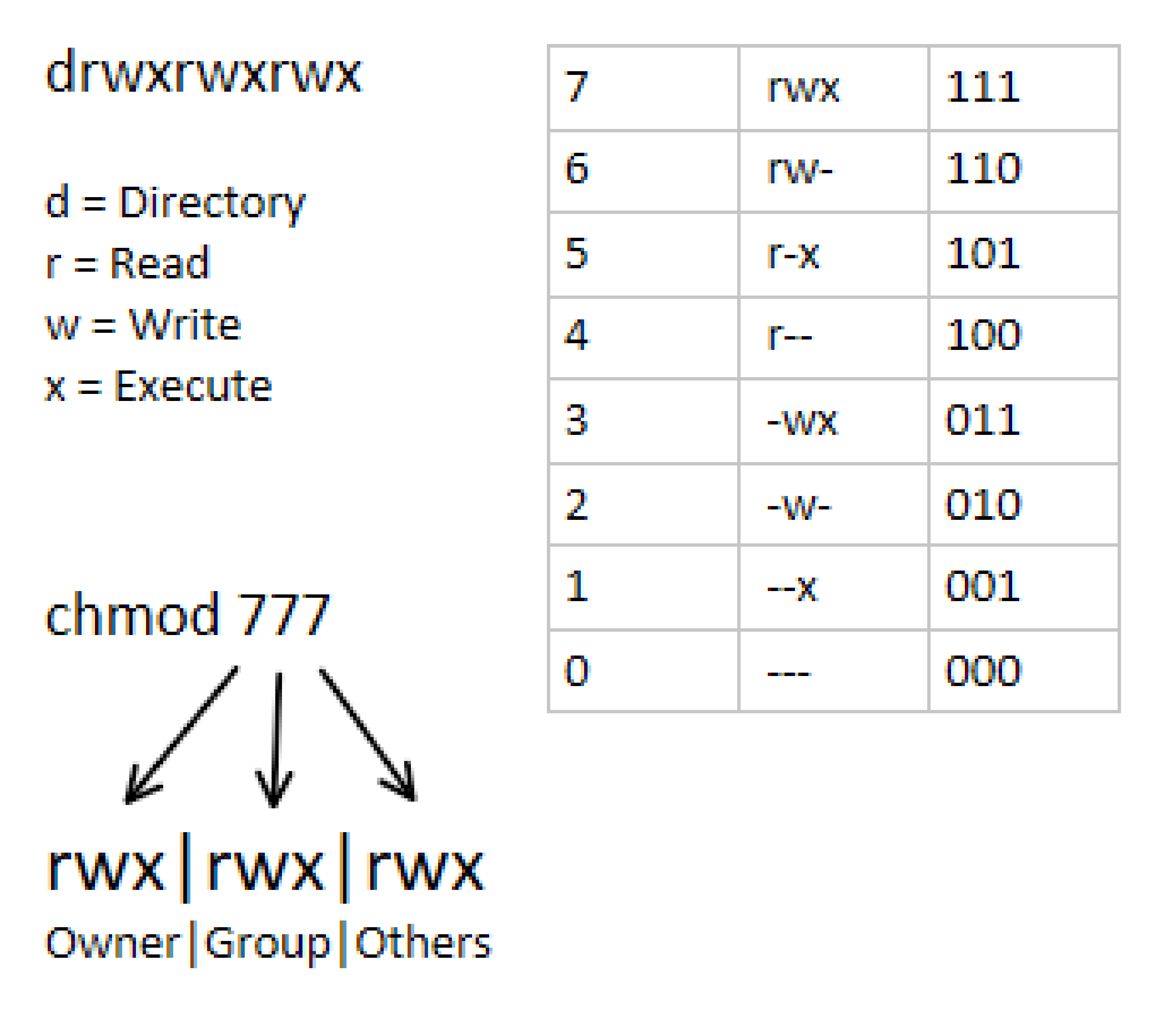
- --- --- --- → - r-- -w- --x

```
--$ chmod u-r g-w o-x somefile
```

- rwx rwx rwx → - -wx r-x rw-

FILE PERMISSIONS

Using chmod with numerical format: --$ chmod 777 somefile



drwxrwxrwx

d = Directory
r = Read
w = Write
x = Execute

chmod 777

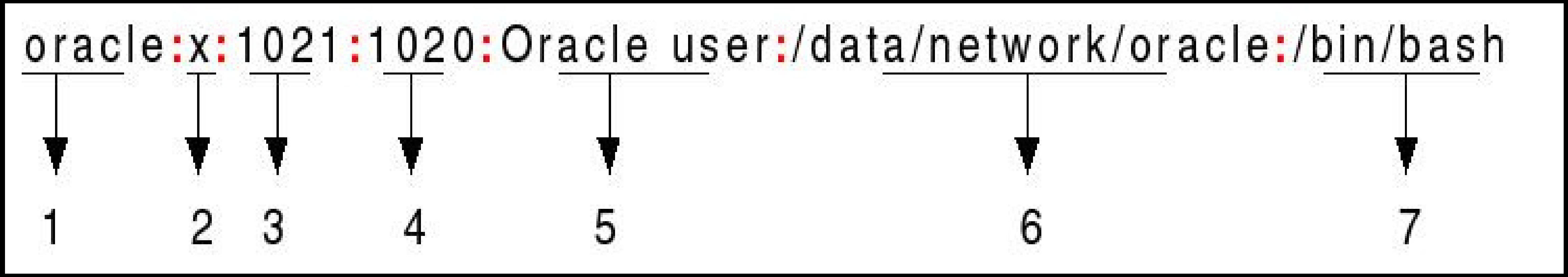
rwx|rwx|rwx
Owner|Group|Others

7	rwx	111
6	rw-	110
5	r-x	101
4	r--	100
3	-wx	011
2	-w-	010
1	--x	001
0	---	000

USERS & THEIR PRIVILEGES

USERS & THEIR PRIVILEGES

The file /etc/passwd contains all users in the Linux machine:

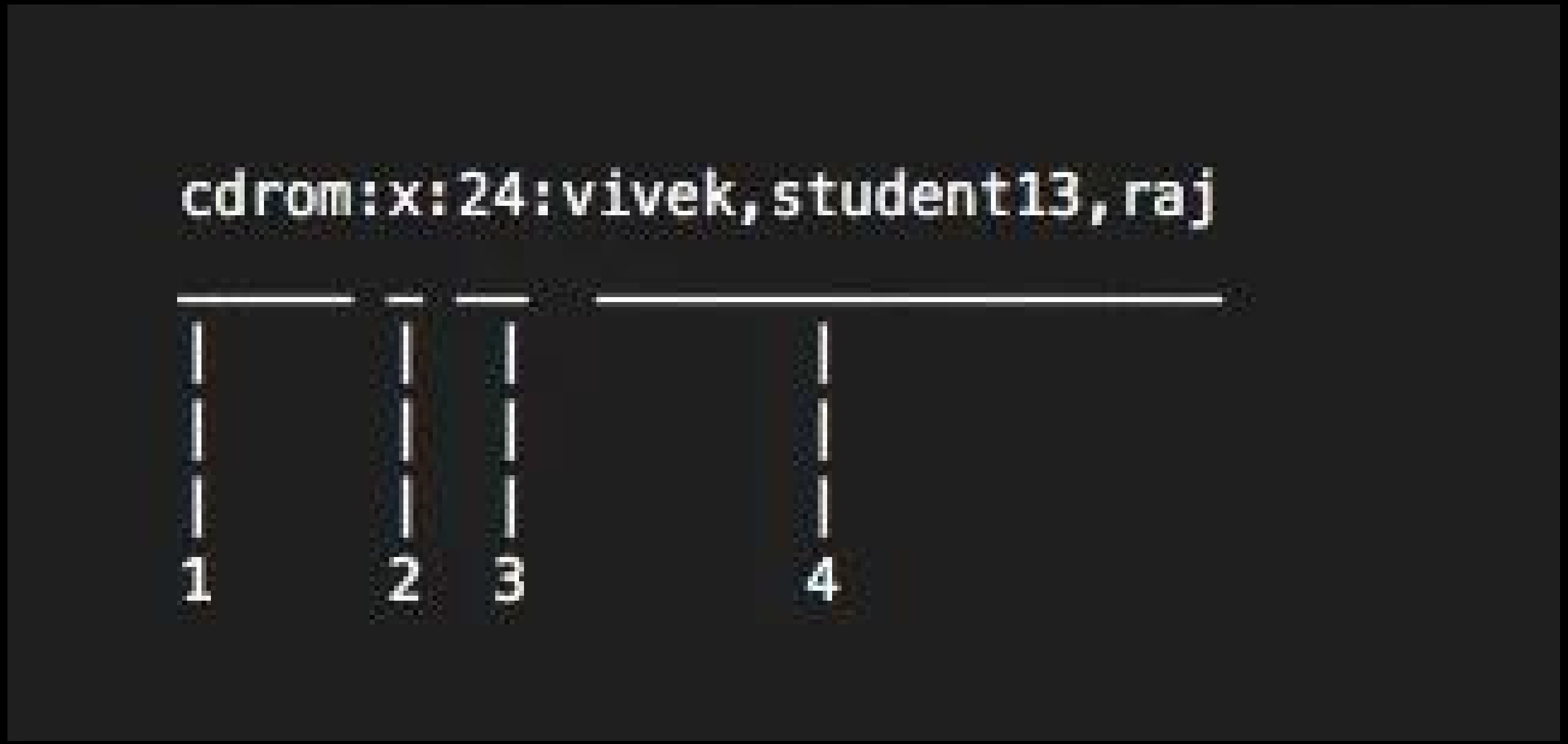


```
oracle:x:1021:1020:Oracle user:/data/network/oracle:/bin/bash
```

1 - Username
2 - Password
3 - UID (User ID)
4 - GID (Group ID)
5 - Comment
6 - Home Directory
7 - Command/Shell

USERS & THEIR PRIVILEGES

The file /etc/group contains all users in the Linux machine:

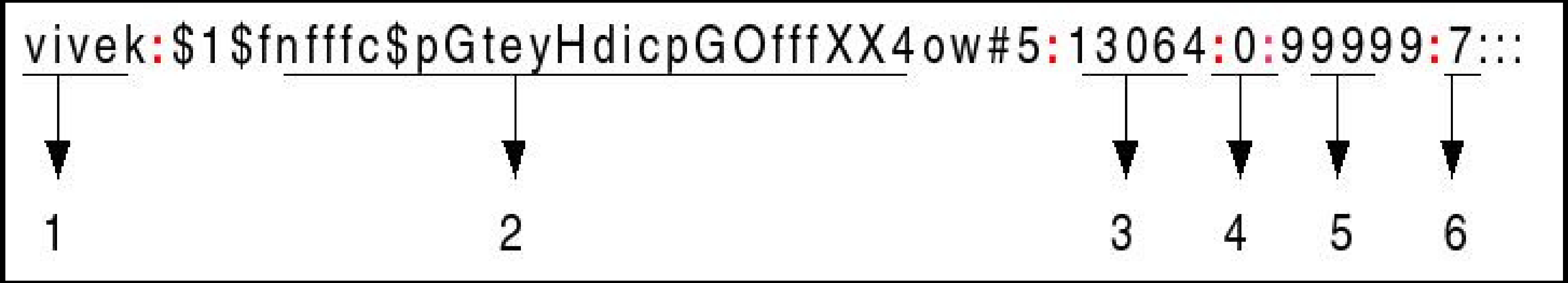


```
cdrom:x:24:vivek,student13,raj
_____ _ __  __________________
|     |  |        |
|     |  |        |
|     |  |        |
1     2  3        4
```

1 - Group Name
2 - Password
3 - GID (Group ID)
4 - members of the group (users)

USERS & THEIR PRIVILEGES

The file /etc/shadow contains all users in the Linux machine:



```
vivek:$1$fnfffc$pGteyHdicpGOfffXX4ow#5:13064:0:99999:7:::
```

1 - Username

2 - Hashed Password

3 - Last password change

4 - Minimum

5 - Maximum

6 - Number of days before the expiration

7 - inactive

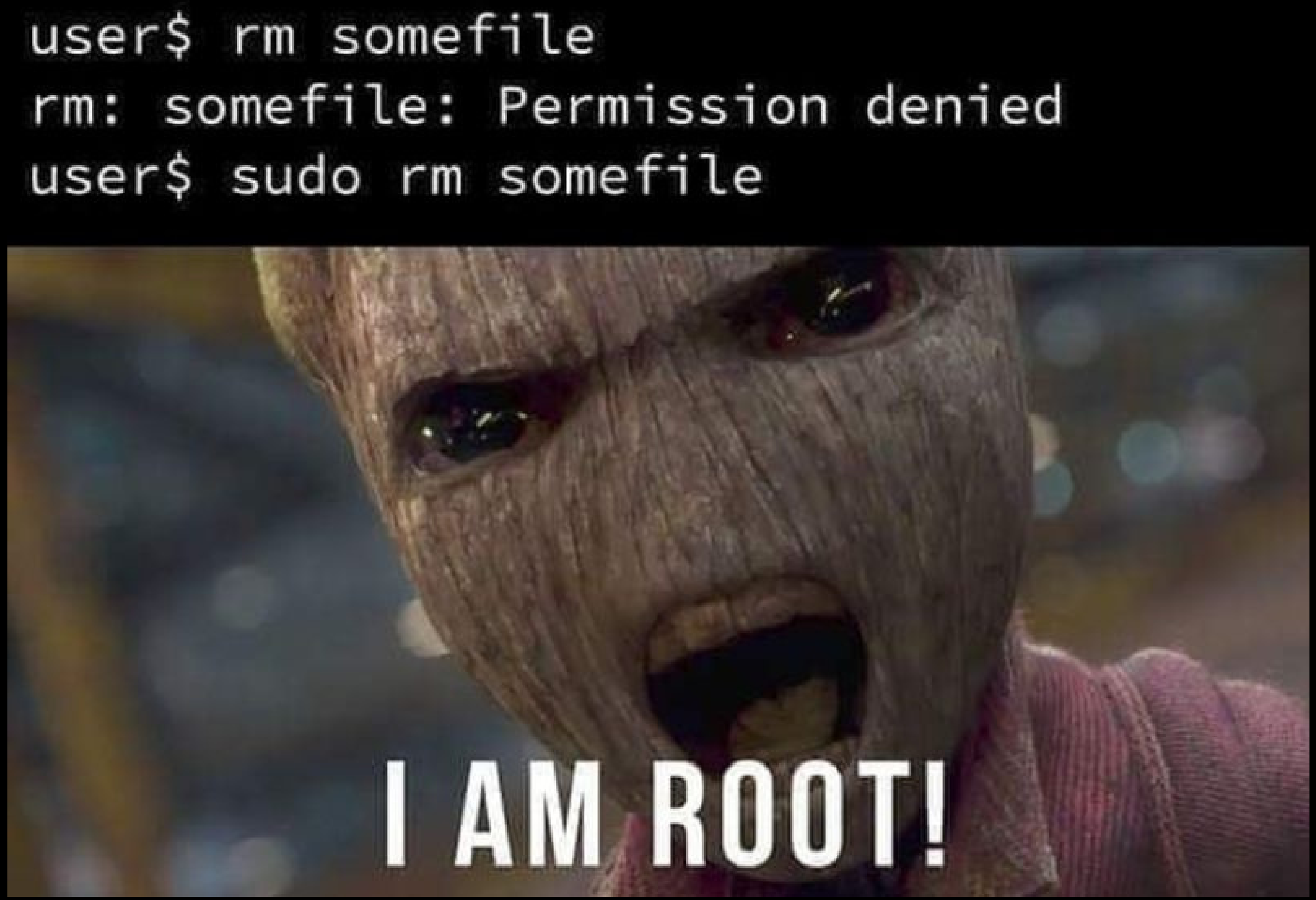
DEMO-1

Users Manipulation Demo ....

USERS & THEIR PRIVILEGES

--$ sudo command

Execute "command" as root



```
user$ rm somefile
rm: somefile: Permission denied
user$ sudo rm somefile
```

--$ su username

Switch to another user

DEMO-2

Cracking /etc/shadow Password Demo....

INSTALLING & UPDATING TOOLS

Update and upgrade the system:

```
--$ sudo apt-get update
--$ sudo apt-get upgrade
```

Install new tools:

```
--$ sudo apt-get install ToolNameHere
```

Install new tools from an online software repository pointed to by your sources

```
--$ git clone https://github.com/.......git
```

Download scripts/binaries from GitHub

SPECIAL CHARACTERS

- The tilde '~'

```
--$ cd ~
--$ cd ~/Desktop
--$ ls ~/Document
--$ nano ~/Desktop/myfile.txt
```

- the pipe '|'

the syntax : Command 1 | command 2 | command 3 | ……

```
--$ cat file1.txt | sort
--$ cat file2.txt | sort | uniq
--$ cat file2.txt | sort | uniq > list4.txt
--$ ls | wc -l
```

SPECIAL CHARACTERS

- the star '*'

```
--$ ls *.png
--$ cat *
--$ locate secret.*
--$ mv *.txt textFiles/
```

- the semi-colon ';'

  the syntax : Command1 ; command2 ; command3 ; ...

```
--$ ls -al ; mkdir newdirectory ; cd ~ ; ls -al
```

SPECIAL CHARACTERS

- The AND '&&'

the syntax : command1 && command2 && ....

```
--$ mkdir newDir && cd newDir
--$ touch script.sh && chmod 700 script.sh
```

- The OR '||'

the syntax : command1 || command2 || ....

```
--$ mkdir newDir || cd newDir
--$ touch script.sh || chmod 700 script.sh
```

- the Ampersand '&'

the syntax : Command [options] &

```
--$ gedit file.txt &
```

SPECIAL CHARACTERS

- The '$'

```
--$ var_1=2020
--$ echo  $var_1
--$ string="CLUB CIT"
--$ echo  welcome to $string
```

Note: dont let spaces arround = when indecating a variable

- the backtick '`'

```
 --$ echo `ls -al` > file.txt
--$ echo $(ls -al) > file.txt
--$ echo "There are `ls | wc -l` files in this directory"
--$ file_count=`ls | wc -l` ; echo "There are $file_count files in this directory"
```

Note: we can replace `command` with $(command)

CHALLENGE-3

Read File Starts With Dash -

CHALLENGE-4

Output The Help Menu Of The echo Command ....

SPECIAL CHARACTERS

- The NOT '!'

```
--$ touch a.doc b.doc a.pdf b.pdf a.xml b.xml a.html b.html
--$ ls
--$ rm -r !(*.html)
--$ ls
```

- the '#'

```
--$ # this will be ignored by bash because it is comment
```

- the '<'

```
--$ sort < mylist.txt
--$ sort < mylist.txt > alphabetical-file.txt
```

USEFUL COMMANDS

- Grep:

```
--$ cat text-file | grep "password"
```

Find the word "password" in text-file

```
--$ cat text-file | grep  -i "password"
```

Find the words "password", "Password", "PASSWORD", "PaSsWoRd" ... in text-file

```
--$ cat text-file | grep  -oE "pa..word"
```

Find words in text-file using regular expressions

CHALLENGE-5

Output File Content Without Using : cat more less head tail ...
Just Use grep command

USEFUL COMMANDS

- cut:

```
--$ cat /etc/passwd | cut -d ":" -f 2
```

cut text with delimiter ":" and choose just the 2nd field to display

USEFUL COMMANDS

- tr:

`--$ echo "hello" |  tr l s`	change any character to another character
`--$ echo "hello" | tr a-z A-Z`	change lower to upper characters
`--$ echo "hello" |  tr -d l`	delete character

USEFUL COMMANDS

- find:

```
--$ find / -name "passwd" -type f          find files with the name "passwd"

--$ find / -name "*secret*" -type f        find all files that have "secret" in their name

--$ find / -user "root" -type d            find all directories that have "root" as owner
```

DEMO-3

Stdin, Stdout, Stderr Demo

USEFUL COMMANDS

- file:

```
--$ file File1
--$ file compressed.7z
--$ file audio.wav
```

- unzip:

```
--$ unzip File.zip
--$ unzip filename.zip -d /path/to/directory
--$ unzip -P PasswOrd filename.zip
```

USEFUL COMMANDS

- ifconfig:

```
--$ ifconfig
```

- ip:

```
--$ ip addr
```

- ping:

```
--$ ping google.com
--$ ping 8.8.8.8 -c 1
--$ ping 10.0.2.8
```

MORE RESOURCES

- Rooms In TryHackMe :

Linux Fundamentals Part 1

Embark on the journey of learning the fundamentals of Linux. Learn to run some of the first essential commands on an interactive terminal.

TryHackMe

Linux Fundamentals Part 2

Continue your learning Linux journey with part two. You will be learning how to log in to a Linux machine using SSH, how to advance your commands, file system interaction.

TryHackMe

Linux Fundamentals Part 3

Power-up your Linux skills and get hands-on with some common utilities that you are likely to use day-to-day!

TryHackMe

MORE RESOURCES

- Rooms In TryHackMe :

The find command

A learn-by-doing approach to the find command

TryHackMe

Toolbox: Vim

Learn vim, a universal text editor that can be incredibly powerful when used properly. From basic text editing to editing of binary files, Vim can be an important arsenal in a security toolkit.

TryHackMe

Bash Scripting

A Walkthrough room to teach you the basics of bash scripting

TryHackMe

MORE RESOURCES

- Rooms In TryHackMe :

Linux Strength Training

Guided room for beginners to learn/reinforce linux command line skills

TryHackMe

Linux Backdoors

Learn all the different techniques used to backdoor a linux machine!

TryHackMe

Linux Modules

Learn linux modules in a fun way

TryHackMe

MORE RESOURCES

- PRACTICING:

- overthewire.org-bandit

- picoCTF