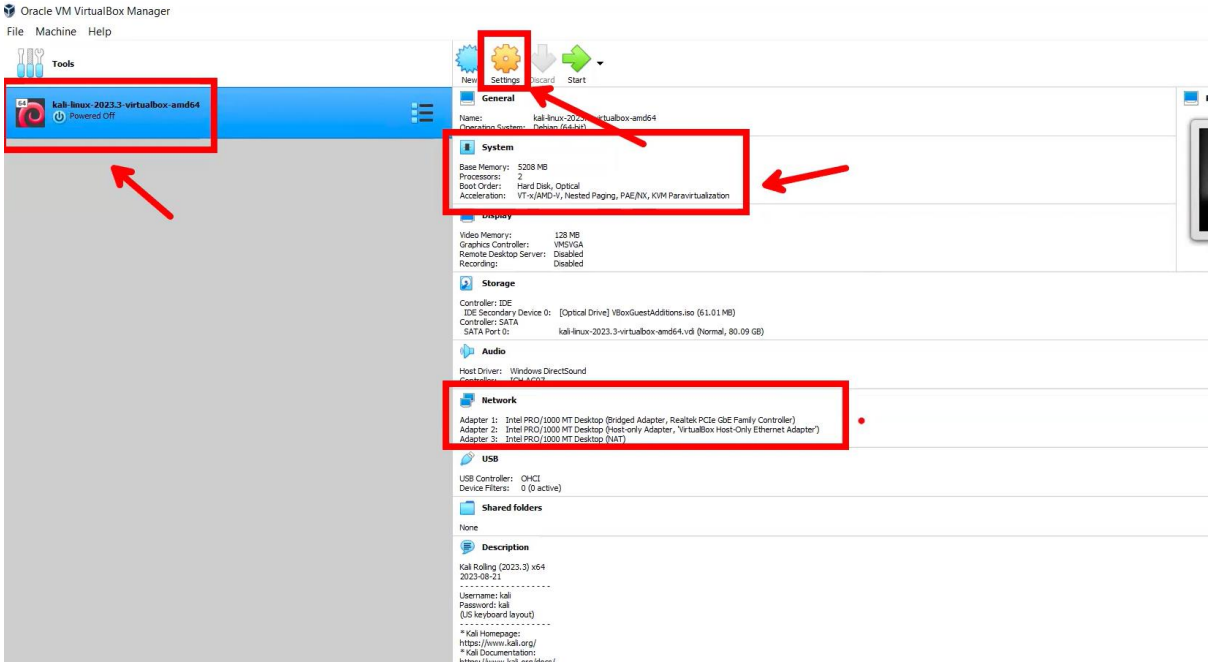
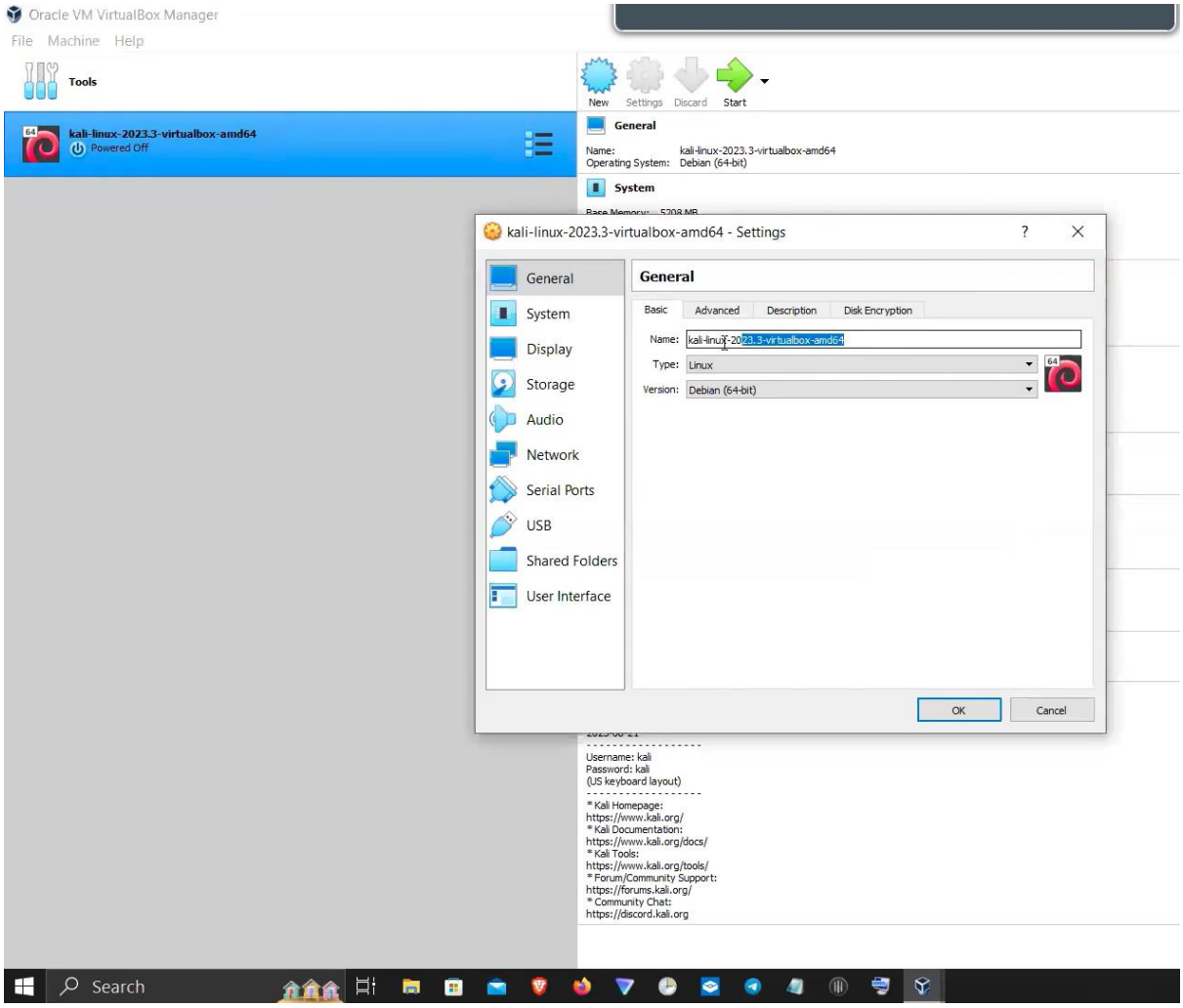
Day 5 :

How you can change virtual machine system configuration



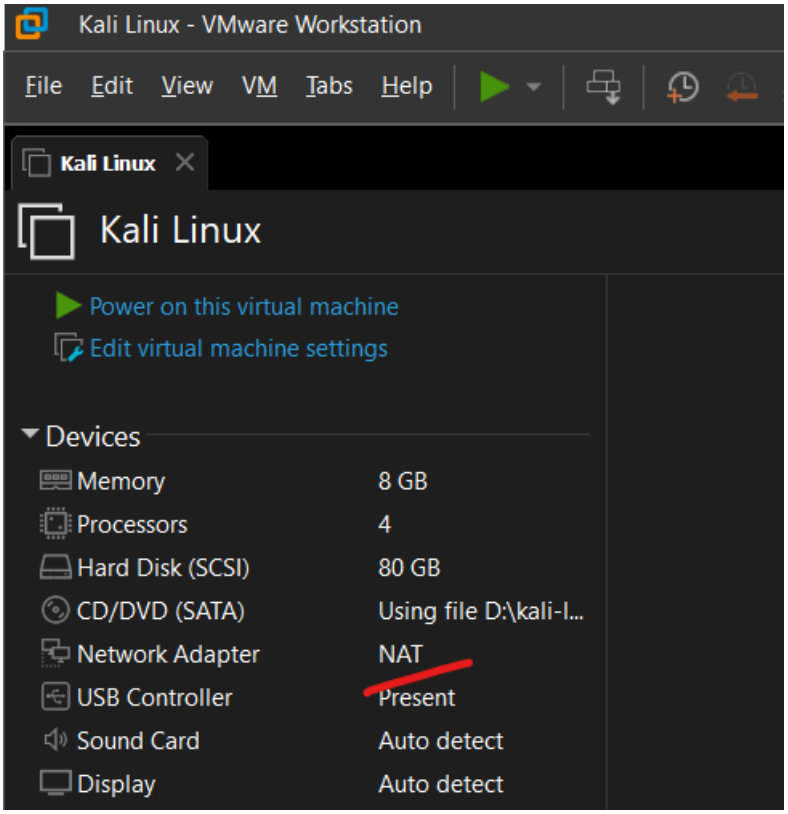


NAT :

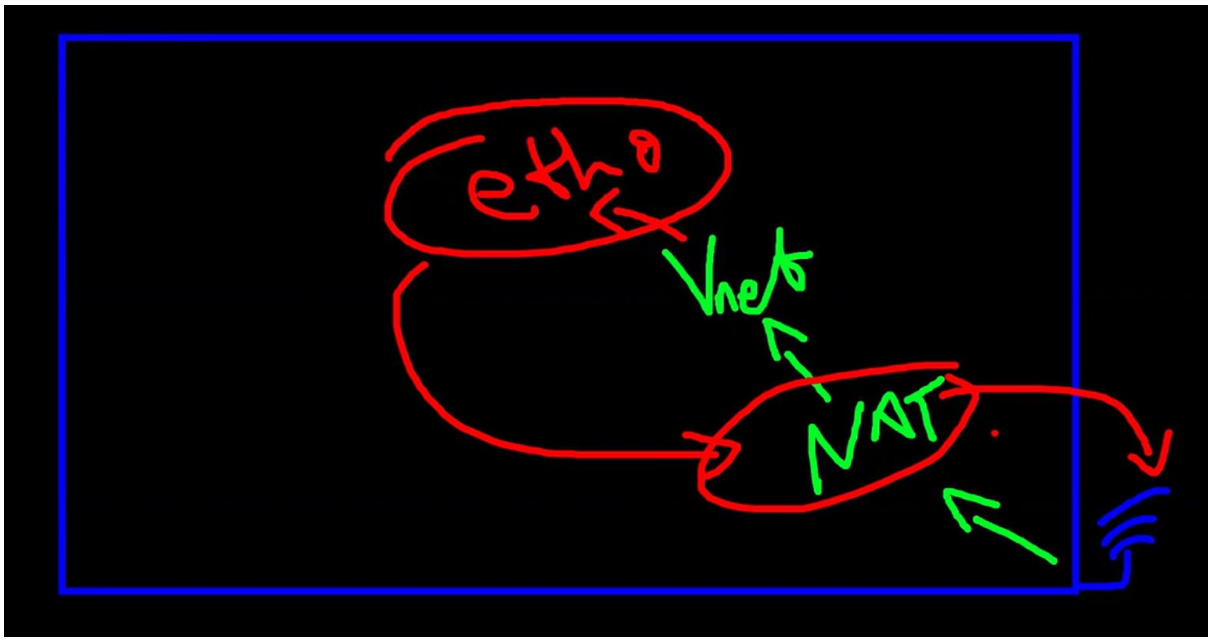
NAT stands for network address translation. It's a way to map multiple private addresses inside a local network to a public IP address before transferring the information onto the internet. Organizations that want multiple devices to employ a single IP address use NAT, as do most home routers. If you're connecting from your home right now, chances are your cable modem or DSL router is already providing NAT to your home.

By default NAT adapter is present in machine you set up in virtual machine, here we deploy kali machine in virtual box kali may have default network adapter is NAT.

Our Kali NAT is used to translate host machine network to virtual machine network, hence we can use network and other devices in network via virtual machine (kali linux).



Lets see In simple language how it works,



As shown in picture the rectangle is our machine, 3 blue line our Wi-Fi so internet comes to wifi then it goes NAT and NAT translate public internet to accessible private internet and future it creates virtual network Vnet for virtual machine operating system kali Linux and VM application crates eth0 adapter for kali linux which is virtual or we can say software based not physical ethernet.

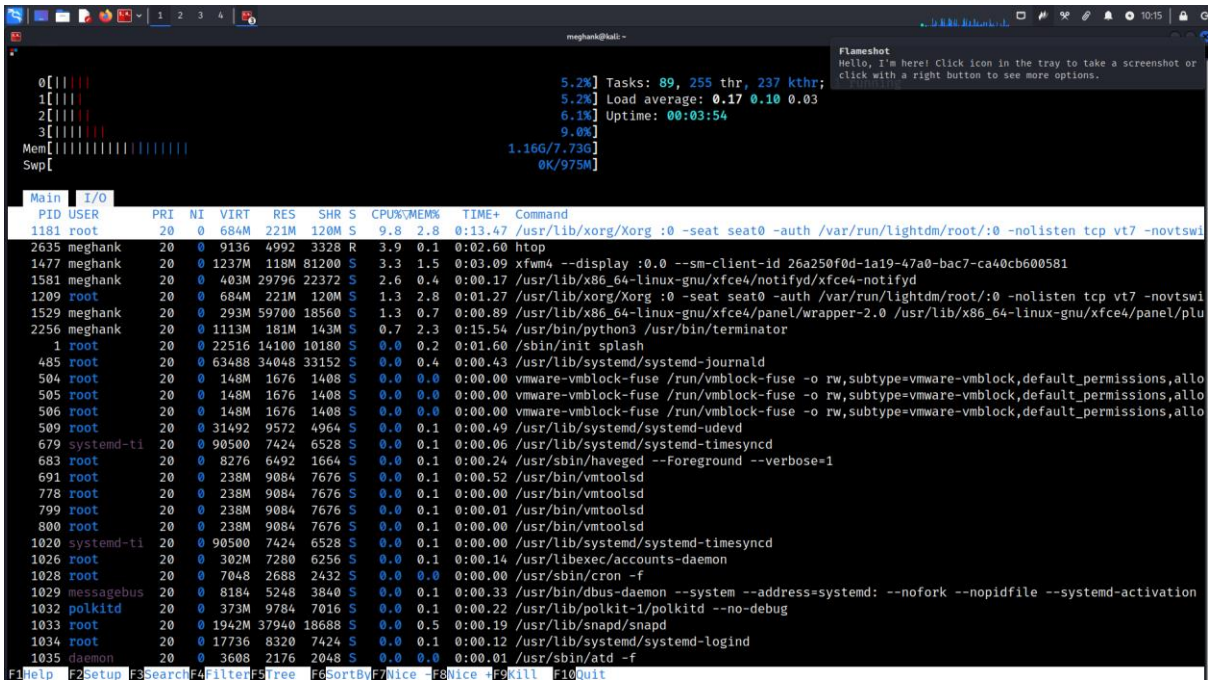
Host-Only adapter in Virtual Box :

A Host-Only Ethernet Adapter allows communication between the VM and the computer it's running on. For example, you can setup shared folders or run a web services on the VM and access it from the host. However it doesn't allow the VM to access anything else in your network, not even the internet.

htop :

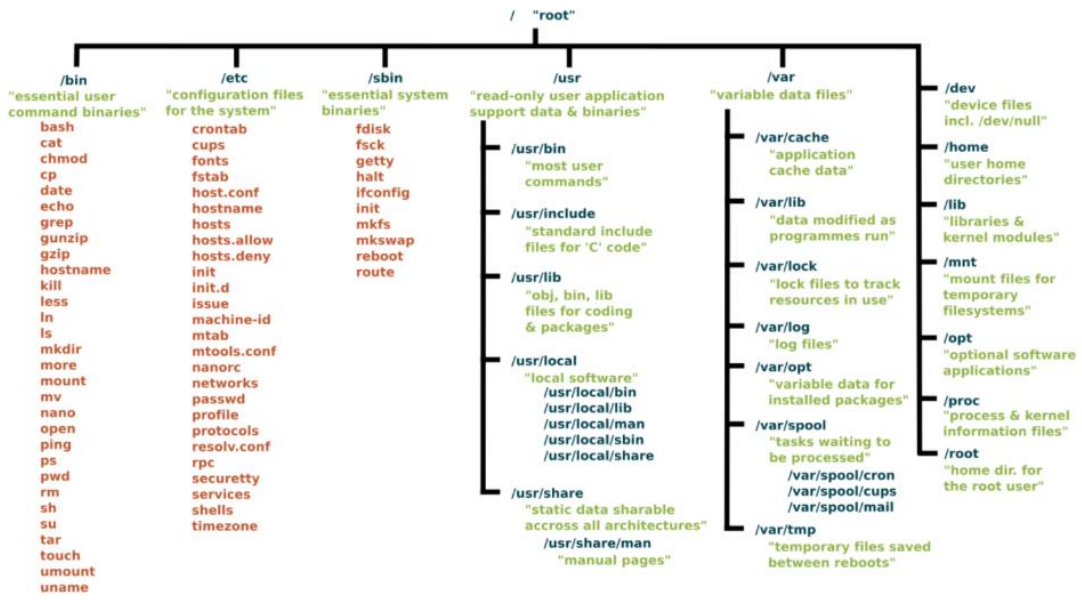
The htop is a command-line utility that allows you to interactively monitor your system's vital resources or server processes in real-time

In a nutshell, htop is a useful command-line tool in the Linux environment to determine the cause of load by each process. It is similar to Task Manager in the Windows OS environment. It can be used to troubleshoot and kill a process that is utilizing excessive server resources



Linux File Structure :

Linux file system/ Directory



1. **/ (Root)**: The starting point for the file system hierarchy. All other directories are subdirectories of the root directory.
2. **`/bin`**: Contains essential command binaries required for booting and repairing the system.
3. **`/etc`**: Holds system-wide configuration files and shell scripts used to initialize system settings for applications.
4. **`/home`**: Home directories for all users. Each user has a subdirectory named after their username.
5. **`/var`**: Contains variable data files such as logs, databases, and temporary files.
6. **`/usr`**: Contains user binaries, libraries, documentation, etc. It's a secondary hierarchy for read-only user data.
7. **`/lib`**: Contains essential shared libraries and kernel modules.
8. **`/dev`**: Contains device files which represent hardware components.
9. **`/tmp`**: Temporary storage for files. It's cleared on system reboot.
10. **`/opt`**: Optional application software packages.
11. **`/sbin`**: Contains system binaries essential for booting, restoring, and recovering the system.
12. **`/srv`**: Contains data for services provided by the system.
13. **`/proc`**: A virtual filesystem that provides detailed information about kernel and processes.
14. **`/sys`**: A virtual filesystem that provides an interface to kernel data structures.
15. **`/run`**: Contains runtime data for processes started since the last boot.
16. **`/boot`**: Contains files needed to start the boot process.
17. **`/mnt`**: Temporary mount points for mounting filesystems.
18. **`/media`**: Mount points for removable media like USB drives and CDs.

File permissions :

File permissions are an essential part of managing and maintaining a Linux system because they control the access to files and directories and ensure the security of the Linux system and the data stored in the files and directories.

```
/  10:27
❯ ls -l
total 72
drwxr-xr-x   5 root root  4096 Aug 17 10:47 ReconDog
lrwxrwxrwx   1 root root     7 Jun  2 05:15 bin -> usr/bin
drwxr-xr-x   3 root root  4096 Aug 29 09:57 boot
drwxr-xr-x  17 root root  3420 Aug 30 10:12 dev
drwxr-xr-x 196 root root 12288 Aug 30 10:12 etc
drwxr-xr-x   5 root root  4096 Jul 22 18:16 home
lrwxrwxrwx   1 root root    28 Jun 12 05:20 initrd.img -> boot/initrd.img-6.8.11-amd64
lrwxrwxrwx   1 root root    28 Jun 12 05:20 initrd.img.old -> boot/initrd.img-6.6.15-amd64
lrwxrwxrwx   1 root root     7 Jun  2 05:15 lib -> usr/lib
lrwxrwxrwx   1 root root     9 Jun  2 05:35 lib32 -> usr/lib32
lrwxrwxrwx   1 root root     9 Jun  2 05:15 lib64 -> usr/lib64
drwx------   2 root root 16384 Jun  2 05:15 lost+found
drwxr-xr-x   3 root root  4096 Jun  2 05:15 media
drwxr-xr-x   2 root root  4096 Jun  2 05:15 mnt
drwxr-xr-x   4 root root  4096 Aug 13 06:33 opt
dr-xr-xr-x 339 root root     0 Aug 30 10:12 proc
drwx------  28 root root  4096 Aug 29 10:35 root
drwxr-xr-x  38 root root   980 Aug 30 10:12 run
lrwxrwxrwx   1 root root     8 Jun  2 05:15 sbin -> usr/sbin
drwxr-xr-x   9 root root  4096 Aug 10 06:26 snap
drwxr-xr-x   3 root root  4096 Jun  2 05:42 srv
dr-xr-xr-x  13 root root     0 Aug 30 10:12 sys
drwxrwxrwt  17 root root   440 Aug 30 10:17 tmp
drwxr-xr-x  16 root root  4096 Jun  3 05:14 usr
drwxr-xr-x  13 root root  4096 Jul 29 19:00 var
lrwxrwxrwx   1 root root    25 Jun 12 05:20 vmlinuz -> boot/vmlinuz-6.8.11-amd64
lrwxrwxrwx   1 root root    25 Jun 12 05:20 vmlinuz.old -> boot/vmlinuz-6.6.15-amd64
```

Above image I marked at 2 places with red line one is at top at ‘l’ so l represent the file is link file or we can say it is shortcut bin is shortcut which is inside usr/bin.

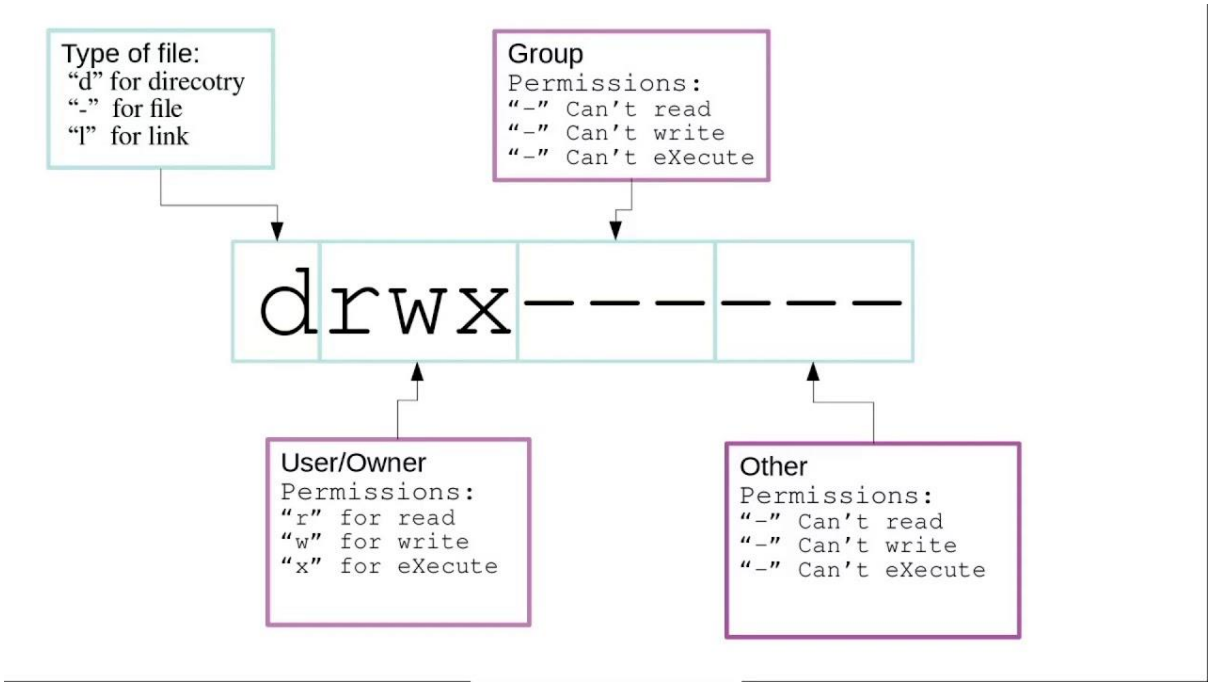
And second mark is ad ‘d’ d represent it is directory.

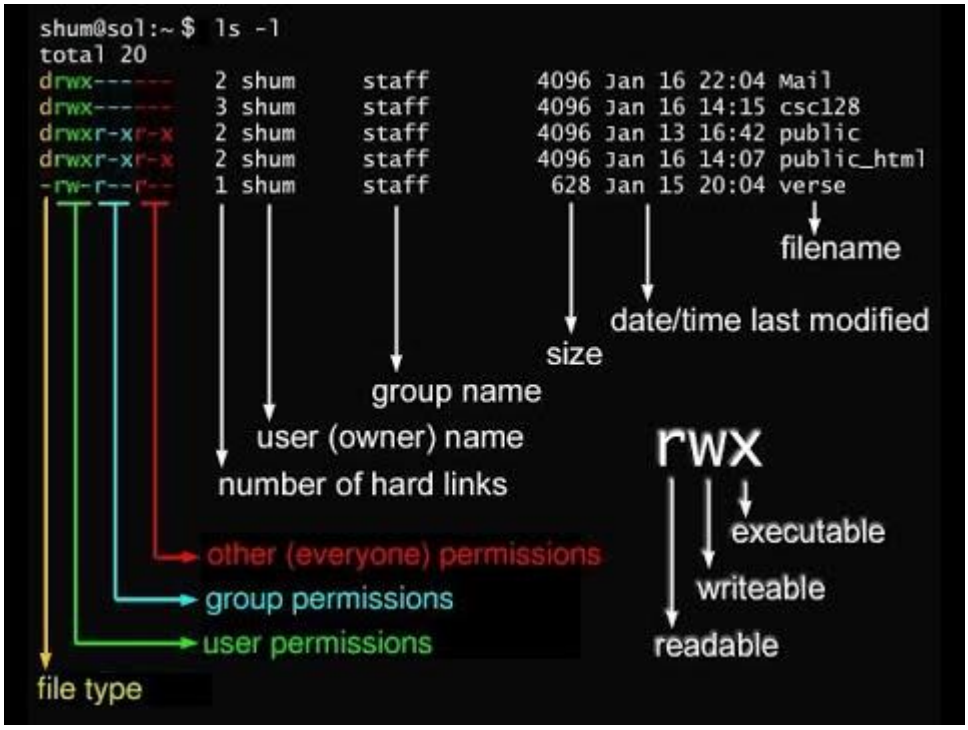
```
drwxr-xr-x  2 root     root      4096 Jun 26 05:32 tmpfiles.d
-rw-r--r--  1 root     root       938 Sep  9  2019 ts.conf
-rw-r--r--  1 root     root      1260 Jan 27  2023 ucf.conf
drwxr-xr-x  4 root     root      4096 Jul 17 05:34 udev
drwxr-xr-x  2 root     root      4096 Aug 27 05:56 udisks2
drwxr-xr-x  3 root     root      4096 Jun  2 05:23 ufw
```

If there is not d or l it is file other than directory or link file which can be anyfile.

Considering above picture there is also present “root root” which show username with root can only access that files or those are shows username and groupname.

Everything in one picture :






```
-rw-r--r--  1 root    root      299 Jun 21 11:35 crack.txt
```

Above crack.txt file owner is root we can change ownership by using command chown

Chown stands for "change owner"

```
~  10:57
❯ sudo chown meghank crack.txt
[sudo] password for meghank:

~  10:57
❯ ls -lh crack.txt
-rw-r--r-- 1 meghank root 299 Jun 21 11:35 crack.txt
```

Here we are change user ownership only from root to Meghank user

grep command :

grep searches for PATTERNS in each FILE. PATTERNS is one or more patterns separated by newline characters, and grep prints each line that matches a pattern. Typically PATTERNS should be quoted when grep is used in a shell command.

```
~  10:43
❯ ls | grep d
Downloads
Videos
calendar.exe
data
demo.txt
encoder.exe
headers
hydra.restore
modified_vim.txt
nerd.txt
passwordlist.txt
practice_sed.txt
subd_for_ss.txt
```

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

Above picture ca easily clear picture for you about file permissions in linux files