

dashes are longer (—), so they are used between phrases or clauses (groups of words). Hyphens are shorter (-)

ctrl + l --> clear terminal in a second

ls -> list down all the files in your directory

nano <file name> -> open a particular file.

used to add content to the blank file
find why vim or gedit can not do that?

vim <file name> -> open a particular file.

gedit <file name> -> open a particular file in graphical user interface.

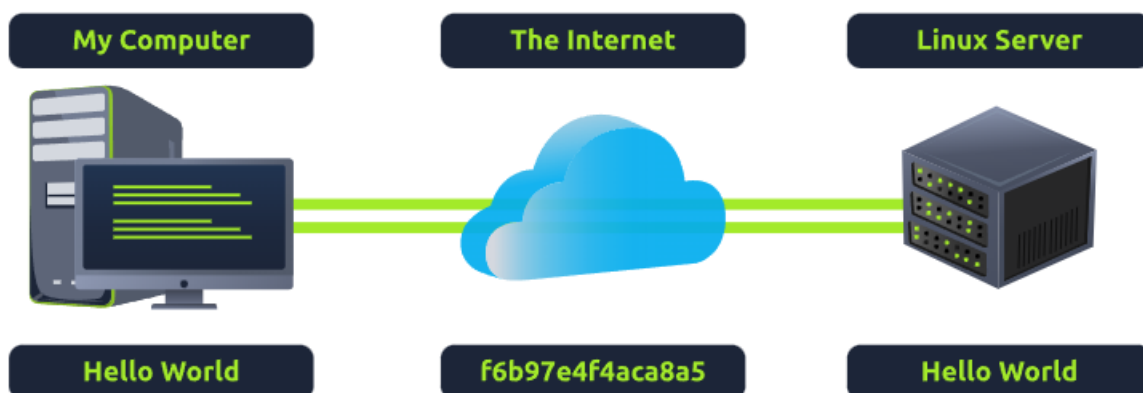
Ctrl + O(letter) -> save the changes we had done to file.

Ctrl + X(letter) -> exit from a particular file.

cat <file name> -> print the content of the file without opening the file.

Secure Shell or SSH

- the common means of connecting to and interacting with the command line of a remote Linux machine.
- SSH simply is a protocol between devices in an encrypted form.



- SSH allows us to remotely execute commands on another device remotely.

The way to access another device using SSH

```
root@ip-10-10-222-189:~# ssh tryhackme@10.10.62.180
The authenticity of host '10.10.62.180 (10.10.62.180)' can't be established.
ECDSA key fingerprint is SHA256:y1UF0F4H/u9NCB0adUTfSGng9lhU0VZKTjgSpCWWe3I.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.62.180' (ECDSA) to the list of known hosts.
tryhackme@10.10.62.180's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1047-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Oct 13 11:08:08 UTC 2023

System load:  0.0               Processes:            107
Usage of /:   26.8% of 7.69GB   Users logged in:     0
Memory usage: 45%              IPv4 address for eth0: 10.10.62.180
Swap usage:   0%
```

ssh <username>@<target_IP_address>

↑
space

↑
IP addresses of the remote machine

we also need to know,
correct credentials to log valid username and password of an account on that machine.

Importance of flags and switches,

ls lists the contents of the working directory. However, hidden files are not shown. We can use flags and switches to extend the behavior of commands.

Example of **-a**,

However, after using the **-a** argument (short for **--all**), we now suddenly have an output with a few more files and folders such as **".hiddenfolder"**. Files and folders with **"."** are hidden files.

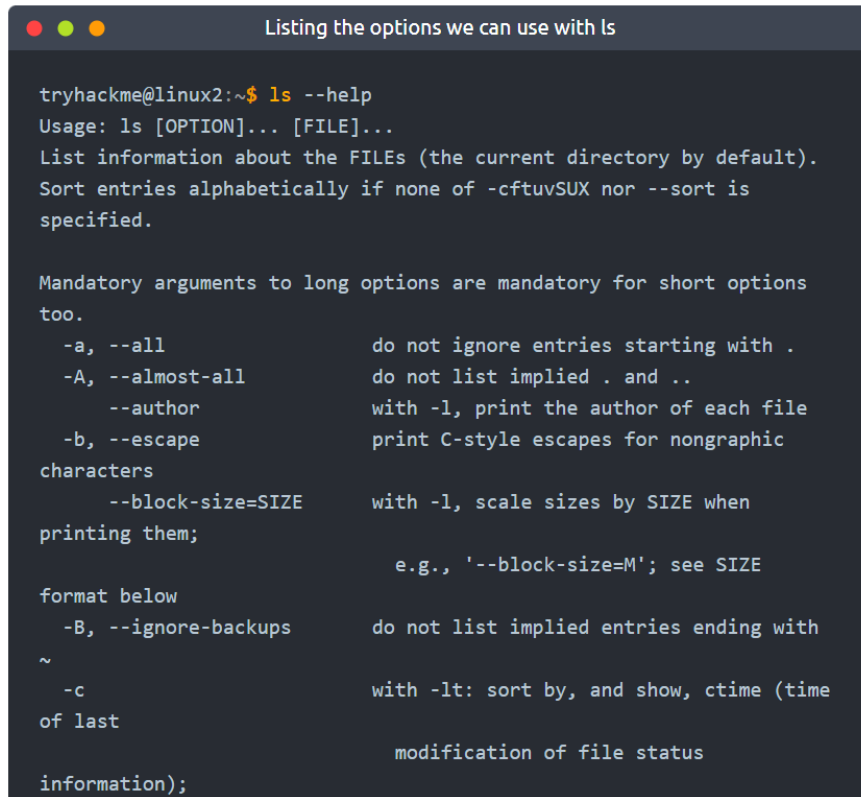
```
Using ls to view hidden folders

tryhackme@linux2:~$ ls -a
.hiddenfolder folder1
tryhackme@linux2:~$
```

To see all the flags and switches behavior of particular command

<Command> --help

If you want to know what the **capabilities have particular command**, we can simply type “--help” after the command.



```
tryhackme@linux2:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is
specified.

Mandatory arguments to long options are mandatory for short options
too.
  -a, --all                do not ignore entries starting with .
  -A, --almost-all        do not list implied . and ..
      --author              with -l, print the author of each file
  -b, --escape             print C-style escapes for nongraphic
characters
      --block-size=SIZE    with -l, scale sizes by SIZE when
printing them;              e.g., '--block-size=M'; see SIZE
format below
  -B, --ignore-backups     do not list implied entries ending with
~
  -c                       with -lt: sort by, and show, ctime (time
of last                    modification of file status
information);
```

Way to access manual page.

Manual page contains about flags and describe the functionality of each of the flags.

Type “man ls” command

Listing the options we can use with ls

```
tryhackme@linux2:~$ man ls
LS(1)
User Commands
LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the
    current directory by default). Sort entries
    alphabetically if none of
    -cftuvSUX nor --sort is specified.
```

Ex:

- What flag use to display the output in a “human-readable” way?
-h

File system related commands

Command	Full Name	Purpose
touch	touch	Create file
mkdir	make directory	Create a folder
cp	copy	Copy a file or folder
mv	move	Move a file or folder
rm	remove	Remove a file or folder
file	file	Determine the type of a file

- create files and folders

touch command

touch <file name>

this creates blank file according to the name of the file.

mkdir command

mkdir <directory name>

this creates blank directory.

- Remove files and folders

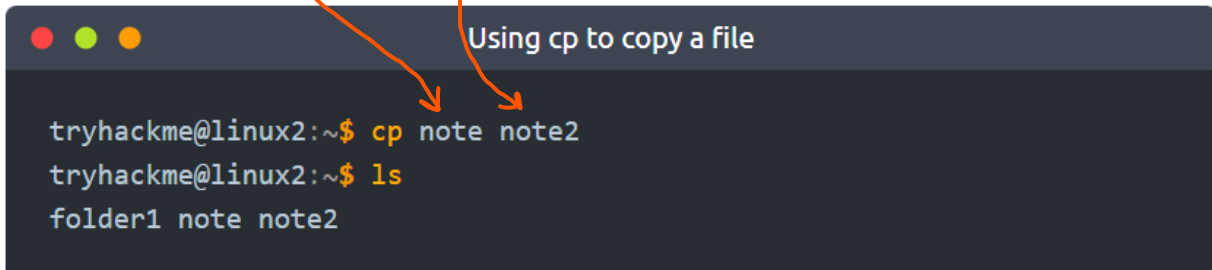
Remove a file

rm <file name>

remove a directory

rm -R <directory name>

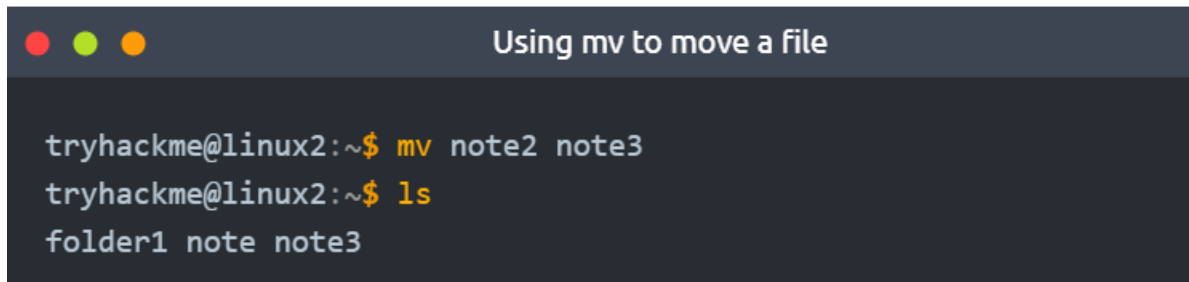
- cp command
this takes two arguments.
 1. The name of the existing file.
 2. The name we wish to assign.



```
tryhackme@linux2:~$ cp note note2
tryhackme@linux2:~$ ls
folder1 note note2
```

“note” is already existing file. But “note2” is newly created file with this command.

- mv command



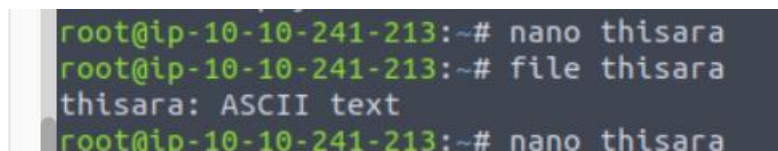
```
tryhackme@linux2:~$ mv note2 note3
tryhackme@linux2:~$ ls
folder1 note note3
```

you use **mv** to move a file to a new folder. But that folder must already create before run this command.

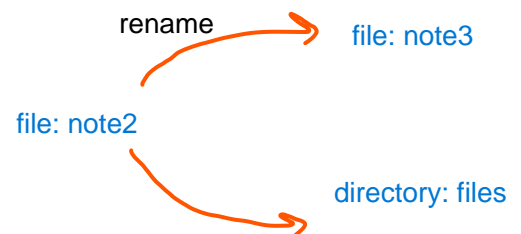
you can also use **mv** to rename a file or folder.

- To view the Data format of a file.

file [file name]



```
root@ip-10-10-241-213:~# nano thisara
root@ip-10-10-241-213:~# file thisara
thisara: ASCII text
root@ip-10-10-241-213:~# nano thisara
```



Permissions

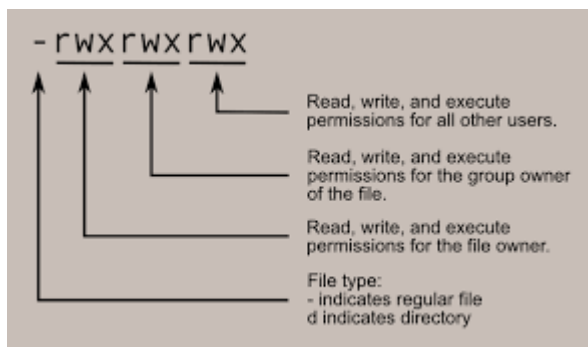
```
Using ls -lh to list the permissions of all files in the directory

tryhackme@linux2:~$ ls -lh
-rw-r--r-- 1 cmnatic cmnatic 0 Feb 19 10:37 file1
-rw-r--r-- 8 cmnatic cmnatic 0 Feb 19 10:37 file2
```

"-" indicator highlighting that it is a file

then "rw", This means that only the owner of the file can read and write to this "cmnatic.pem" file but cannot execute it.

If we can see "r", "w" and "x" then all the privileges are granted. Read, write and execute.



ls -l gahuwama meke enne



One awesome feature of Linux is that you can control who can do what with a file very precisely. Even if one person owns a file, you can let a group of people have their own specific permissions for that file, without changing the file's owner.

Switching Between Users

On a Linux system, you can switch between users using the **su** command.

To switch users:

- The username we wish to switch to and
- The user's password

Ex:

currently logged in as the user "user1," and you want to switch to "user2." Type,

~~su - user2~~
su user2

The difference between users & groups

In Linux, users and groups are different. A regular user and a system user have differences, but Linux allows for precise permission control. Even if a file is owned by a user, it can be shared with a group of users, each having different permissions. For example, a web server system user needs to read and write files for a web app, but web hosting companies want customers to upload files without compromising security.

Common directories

/etc

The root directory is super important, and the "etc" folder is where **your system keeps important files**, which is like a storage place for special files that your computer needs to work.

For example, there's a file in there called "**sudoers**": **which decides who can do special things on the computer**.

There are also files called "passwd" and "**shadow**": **that hold your passwords in a secret way called sha512**, which is unique to Linux.

Go to inside of the "etc" folder and type "ls",

```
tryhackme@linux2:/etc$ ls
shadow passwd sudoers sudoers.d
```

/var

"/var" directory, with "var" being short for **"variable data"**.

This folder **stores data that is frequently accessed or written by services or applications running on the system**.

Ex: **log files** from running services and applications are written here (/var/log)

```
tryhackme@linux2:/var$ ls
backups log opt tmp
```

/root

home directory of a "root" user is /root

"root" user, whose home directory is "/root" and not "/home/root."

The /root folder is like the home for the "root" system user. It's just where the "root" user's stuff is.


```
root@linux2:~# ls
myfile myfolder passwords.xlsx
```

/tmp

The "/tmp" directory in Linux is like a temporary storage space. It's used for data that you only need briefly, and it gets cleared when you restart your computer.

For penetration testing, it's handy because anyone can put files in there. So, when we're testing a system, we can store our tools and scripts there.

```
root@linux2:/tmp# ls
todelete trash.txt rubbish.bin
```