**Introduction to Linux**

Linux is a general purpose computer operating system. Originally build by Linus Torvalds and released

in 1991.

Linux is defined by its kernel, Which is the core component of the system, which interacts with the

computer hardware to allow the software to work.

It was inspired by the Unix and based on a philoshopy of software and operating system should be free.

Both free of cost and freely modifyable under a license GNU Public License.

Linux become popular in many different application and ther are hundreds of different variation of

Linux are availabel. Linux are mostly used in Servel Computing System, Super Computers and Mobile Phones Running Android.

Notes:

Major Distributions:

Arch, Debian, Red Hat, and Slackware

Debian and Red Hat derivations are popular

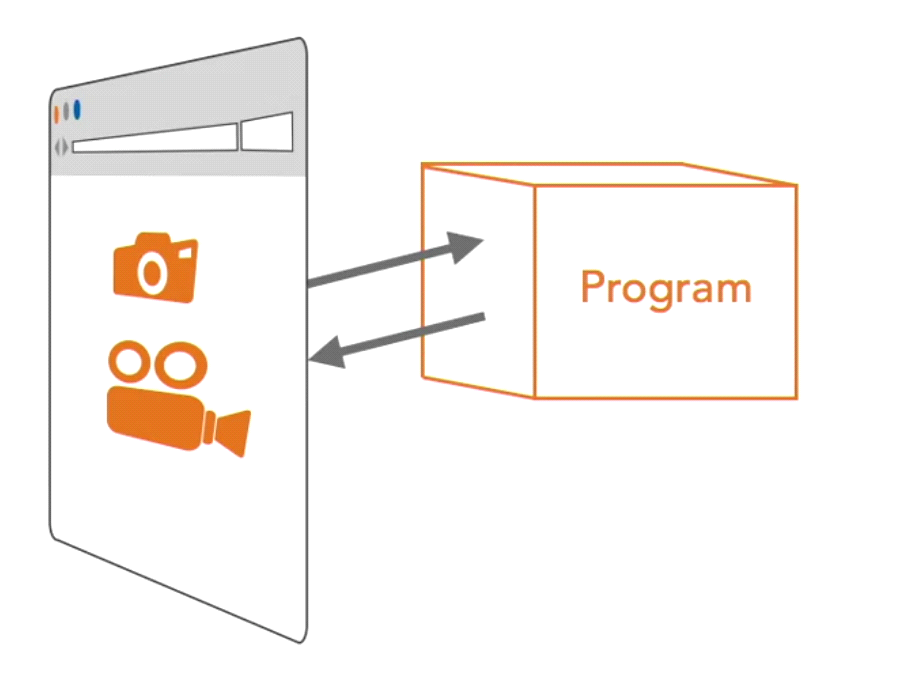
Debian Example:

Linux Mint, Ubuntu, Elementary OS, and Kali Linux

Red Hat Example:

CentOS, Fedora, and Red Hat Enterprise Linux

**What is Command Line**



GUI are actually used to run the program which are running behind to show us information and allow the data to be in a proper human readable format and also to interact with them.

In the world of software development and Administration we don’t use GUI programs to do out tasks. We usually need some text based information to start working. These kind of programs run on some server and server usually doesn’t have monitor.

To interact with the program which doesn’t have a GUI we go for Command Line.

> The Command Line is a Text Based

> In the command line interface we type text command as Input and get Text as output.

> This Command Line Environment is called as Shells or Command Line Interpreter

General Commands Syntax

<command><options><arguments>

There always be a command and ‘options’ and ‘arguments’ are optional

The options tells the command how to operate. Most option starts with a – (dash). And most command come up with multiple options and for multiple options we use multiple -(dash).

Ex:

**ls -l /var/log //one option**

**ls -l -a -h /var/log // multiple options**

**ls -lah /var/log //combining options**

Some options have loger syntax so they use two dashed (--)

The last portion of the command is called as argument or arguments. This is the portion where you tell the command what thing to operate on.

Writing a Command in the Shell

Open a Terminal in your System



~ (tilt) means you’re in home directory

Username

Computer Name

~ (tilt) means you’re in home directory

Command:

**ls**

**ls** command is used to list the content in the Directory

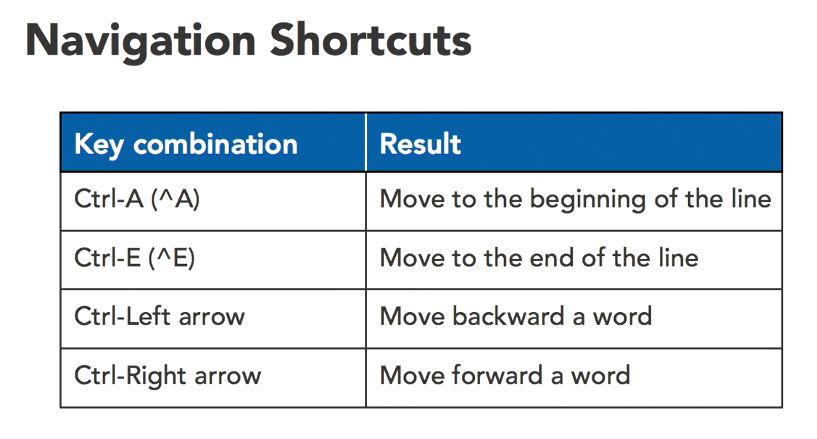
Command:

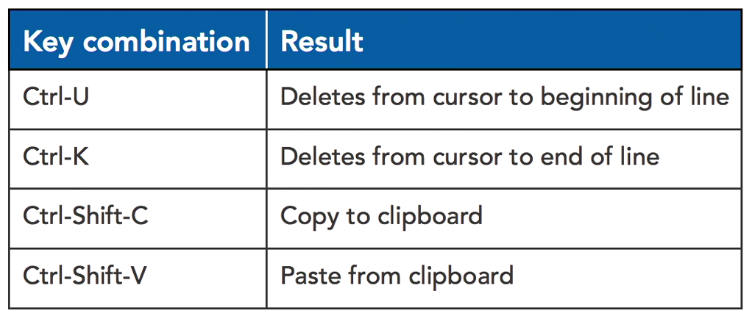
**ls -l**

**-l** option use a long listing format

**Tab Completion**

**>**Automatically completes a file or folder name

> Makes guesses based on the contents of the folder



***You don’t need to memorize everything.***

Find Help for Command:

Command:

**man**

This command is use to view the reference manual or manual page for a specified command

To exit the manual page hit the ‘q’ key

***Explore and Navigate the File Sytem***

**cd**

cd stand for change directory. Which is use to navigate toa file system.

**pwd**

pwd command is used to know the present working directory. This command shows us the full path or absolute path of a folder where we are currently working

**cd ..**

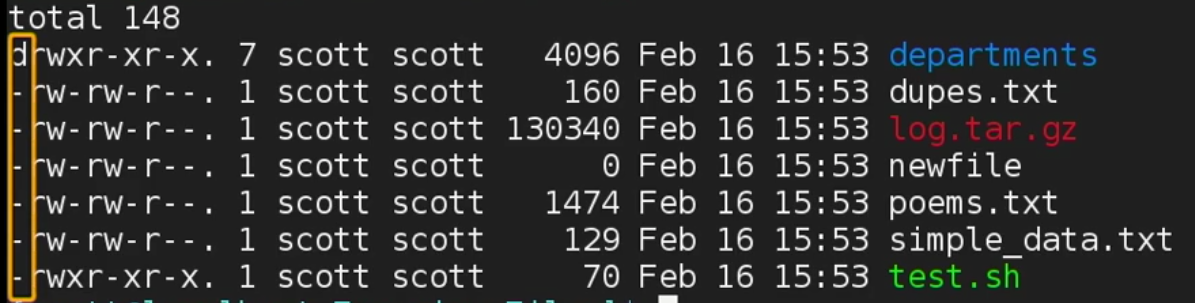
cd .. command is use to move back to the parent directory of the present directory

to directly go to home folder from anywhere in your filesystem type the cd command wtihout any argument.

**cd -**

it is used to switch back and forth between your current folder and previous folder

**More on ls command**

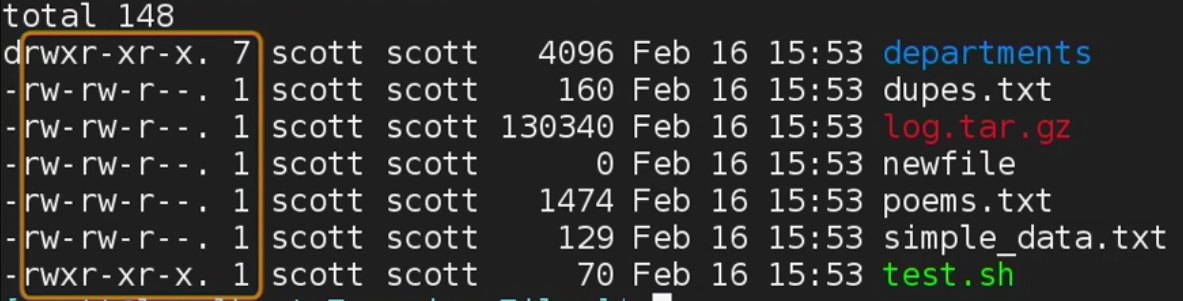


The first column here shows wether the item in the list is a folder/directory or a file of a link.

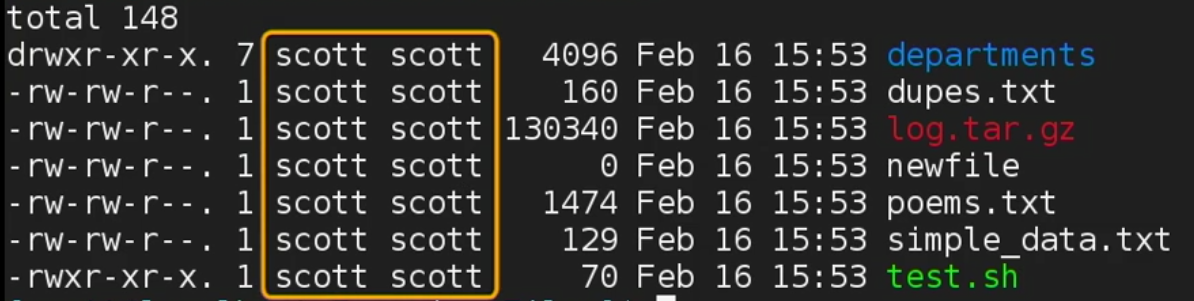
‘d’ stands for directory

‘-’ stands for a file

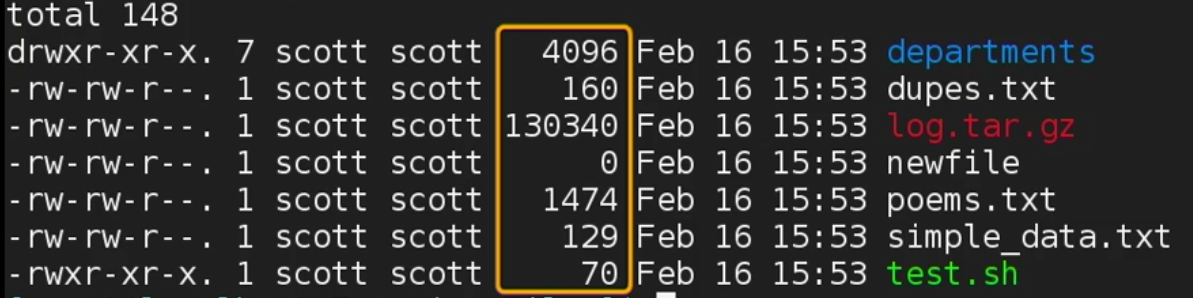
‘-l’ stands for a link



The next set of column show a representation of permission on the file.

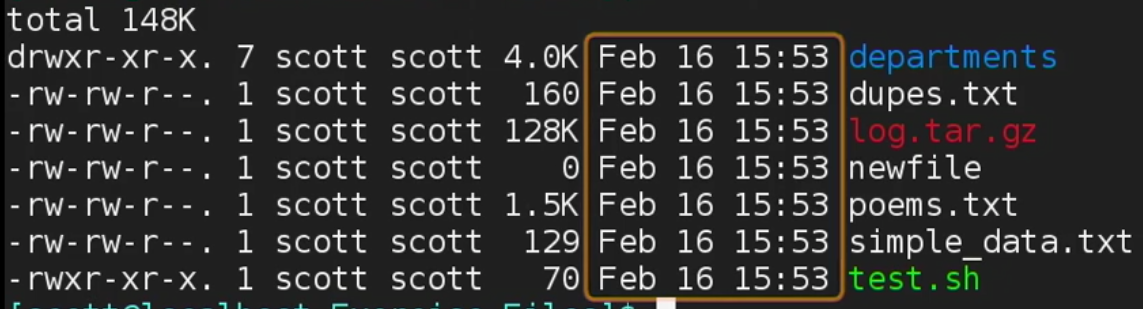


This set of colums show the owner of the file and the group setting of the file.

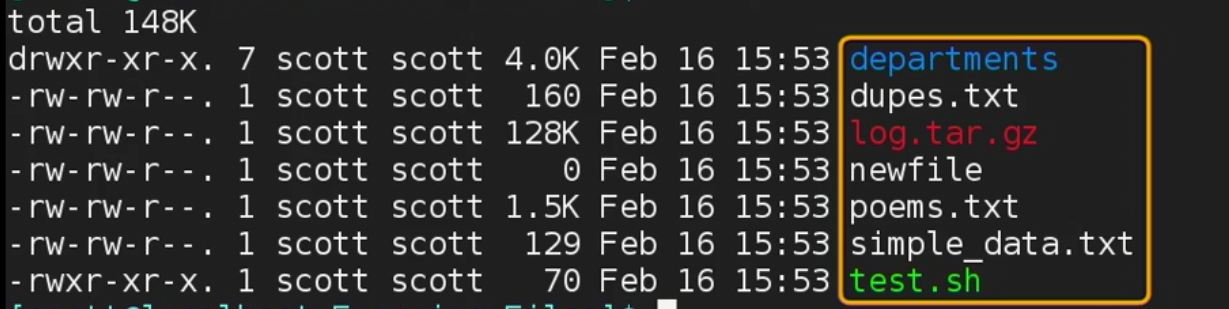


This set of colums represent the size of the file in bytes.

To get a proper size of a file type: **ls -l -h** or **ls -lh**

****

The set of columns is used to show the Date and Time when the file was modified.



This set of column is used to represent the file name or link name is case of link

**Create and remove folders**

**mkdir <folder\_name>**

To create folder/directory in the present working directory we use the **mkdir <folder\_name>**

**mkdir <folder\_name>**

Tocreate multiple folder:

**mkdir <folder-name><folder-name><folder-name> ...**

To create nested folder:

**mkdir -p <folder-name>**

This options create all the needed parent folder

**rmdir**

This command is use to delete a folder

**rmdir <folder-name>**

To remove a folder, the folder has to be empty. If the folder is not empty, you will get an error.

**Copy, move, and delete files and folders**

**cp**

This command is use to copy a file.

**cp <source><destination>**

**mv**

This command is use to move a file between folders and it can also be used to rename a file

**mv <source><destination>**

To rename a file:

**mv <file-name><new-file-name>**

To move a file to the current working directory:

**mv <file-name> .**

The ‘**.**’ is use to represent the current working directory

**Wildcards**Wildcards are the character that stands for the pattern.

The two most common wild cards are:

\* -- any number of any characters

? -- one of any character

example:

**mv \*.txt /home/user/Downloads**

This means any file with .txt extentions will be moved to Downloads folder

**mv /home/user/Download/\* .**

This means all the files in the Download folder will be moved to the current working directory because we use a dot (**.**) operator.

**rm**

This command is use to remove or delete a file. Note: The file will not be moved in the trash can or recycle bin, It will be deleted permanently.

**rm <file-name>**

example with ? Wildcard:

Files:

data1.txt, data2.txt, data3.txt, data4.txt

To delete all the file whose name starts with data:

**rm data?.txt**

This means delete all the filename whose name starts with data and then any one chars after it.

To delete all the folder and files recursievly:

**rm -r <folder-name>**

**Find Files from the command line**

Sometimes you need to find a file which you don’t know where it is.

**find**

This command is use to search a file in a directory hierarchy

**find <directory><option><file-name or pattern>**

example:

**find . -name “poe\*”**

**find . -name “do\*”**

**find . -name “d\*”**

**find . -name “\*d\*”**

The above commands will find a file in the present working directory with the given file name.

**find ~/Documents -name “\*d\*”**

The ‘**~**’ represent the home folder and the command says, look into the Documents folder with a file name which contains ‘d’ in it.

**Understanding user roles and sudo**

Linux is a multi-user environment. Means the files and folder can be kept separated in the users home folder. We can create file that only or another user can access.

At the command line we can swithch between user using **su** command. The **su** stands for switch user.

Example:

**su <username>**

The most common use of **su** command is to do some system administration tasks.

There are two basic user roles in linux:

\* normal user

\* super user / root user

The differece between these user is of privilages.

The noraml user:

> Can’t make changes to the system.

> It can’t install softwares.

> It can’t make changes to the system files

> It can’t browse other user’s home folder.

The Super user or Root User:

> Can make changes to the system

> It can install software

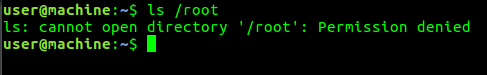
> It can start and stop services. Etc.

Normal use can be granted the ability to temporarily use root’s power through a command called **sudo**. Which is sort for super user do and substitue user do.

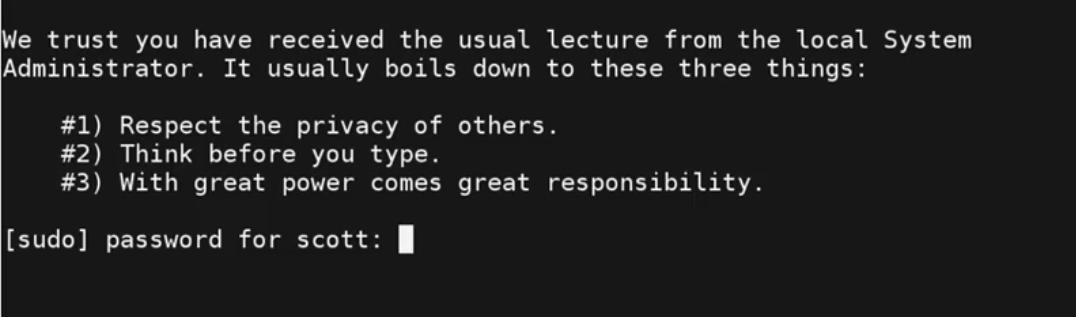
Note: It’s uncommon and a bad practice to login to root user to do common task. You only wants to borrow root power when you need it.

If you try this command:

**ls /root**



The first time a user uses **sudo**. They will see a little Note:



The above note reminds that using **sudo** is a big step. You will only see the dialouge once, the next time you use **sudo** it will not come.

To navigate as a super user, type **sudo ls /root**

**Note:** There will be a grace period, when you again type **sudo**. It won’t ask for the password for a certain amount of time.

To give up the previlages type: **sudo -k**

If you are doing a lot of work as a super or root user, you can switch to the root user by typing: **su root** //then type the root password

To switch back to the normal user, type: **exit**

**File Permissions**

****

**chmod**

This command is use to change the permission of a file.

There are two ways to represent a permission:

- Ocatal (e.g., 755, 644, and 777)

- Symbolic (e.g., a=r, g+w, and o-x)

