Types of Files in Linux:

In Linux everything is treated as File.

All files are divided into 3 types

1) Normal or Ordinary files:

These files contain data. It can be either text files (like abc.txt) OR binary files (like images, videos etc).

2) Directory Files:

 These files represent directories.

 In windows, we can use folder terminology where as in linux we can use directory terminology.

 Directory can contains files and sub directories.

3) Device Files:

In Linux, every device is represented as a file. By using this file we can communicate with that device.

Note: short-cut commands to open and close terminal

ctrl+alt+t To open terminal

ctrl+d To close terminal

How to check File Type:

In Ubuntu, blue color files represents directories and all remaining are considered as normal files. This color conventions are varied from flavour to flavour. Hence it is not standard way to check file type.

We have to use 'ls -l' command.

drwxr-xr-x 102 durgasoft durgasoft 4096 Nov 17 21:19 magic

-rw-r--r-- 1 durgasoft durgasoft 0 Nov 17 21:24 sunny.txt

The first character represents the type of file.

d □ Directory File

- □ Normal File

l □ Link File

c □ Character Special File

b □ Block Special File

s □ Socket File

Note: c, b, s are representing system files and mostly used by super user (also known as root user or admin user)

File System Navigation Commands:

1) Every directory implicitly contains 2 directories . and ..

. Represents Current Directory

.. Represents Parent Directory

2) $ cd .

Changes to Current Directory (Useless)

3) $ cd ..

Changes to Parent Directory

4) $cd

If we are not passing any argument, then changes to user home directory.

5) $ cd ~

~ Means User Home Directory.

It will Changes to User Home Directory.

6) $ cd -

- Means Previous Working Directory.

It will Changes to Previous Working Directory.

Linux File System Hierarchy:

Linux file system has Tree Like Structure.

It starts with root(/).

/ is the topmost directory

This root directory contains the following important sub directories.

bin,sbin,lib,etc,dev,opt,home,usr,tmp,media etc

1) bin Directory:

bin means binary. This directory contains all binary executables related to our linux commands.

2) sbin Directory:

sbin means systembin. It contains all binary executables related to high end admin

(super user OR root) commands.

Eg: Disk partitioning, network management etc

3) etc Directory:

 This directory contains all system configuration files. These configurations can be used

to customize behaviour of linux os.

 All users information available in /etc/passwd file.

 All groups information available in /etc/group file.

 Hosts information (ip address and dns names) available in /etc/hosts file.

4) tmp Directory:

 tmp means temporary. It contains all temporary files created in the current session.

 If any file is required only for the current session, then create that file inside tmp

directory. These files will be deleted automatically at the time of system shutdown.

 If any file which is required permanently, then it is not recommended to create inside

tmp directory.

5) dev Directory:

 dev means device.

 In Linux, everything is treated as a file including devices also. i.e every device is

represented as a file. By using these files, we can communicate with the devices.

 All device related files will be stored inside dev directory

home Directory:

 As linux is multi user operating system, for every user a separate directory will be

created to hold his specific data like videos, images, documents etc. All these user

directories will be stored inside home directory.

 $ ls /home

Pwd - the command **pwd** to print the current working directory −

$pwd

/user0/home/amrood

Cd - the cd command to do more than just change to a home directory. You can use it to change to any directory by specifying a valid absolute or relative path. The syntax is as given below –

$cd dirname

Here, dirname is the name of the directory that you want to change to. For example, the command –

$cd /usr/local/bin
$

Mkdir -Creating Directories

We will now understand how to create directories. Directories are created by the following command −

$mkdir dirname
$mkdir /tmp/test-dir

$

This command creates the directory test-dir in the /tmp directory. The mkdir command produces no output if it successfully creates the requested directory.

If you give more than one directory on the command line, mkdir creates each of the directories. For example, −

$mkdir docs pub

Removing Directories

Directories can be deleted using the **rmdir** command as follows −

$rmdir dirname
$

Note − To remove a directory, make sure it is empty which means there should not be any file or sub-directory inside this directory.

You can remove multiple directories at a time as follows −

$rmdir dirname1 dirname2 dirname3
$

The above command removes the directories dirname1, dirname2, and dirname3, if they are empty. The **rmdir** command produces no output if it is successful.

Listing Files

To list the files and directories stored in the current directory, use the following command −

$ls

Here is the sample output of the above command −

$ls

bin hosts lib res.03

ch07 hw1 pub test_results

ch07.bak hw2 res.01 users

docs hw3 res.02 work

A path is a unique location to a file or a folder in a file system of an OS.A path to a file is a combination of / and alpha-numeric characters.

Absolute Path-name

An absolute path is defined as the specifying the location of a file or directory from the root directory(/).

To write an absolute path-name:

Start at the root directory (/) and work down.

Write a slash (/) after every directory name (last one is optional)

For Example :

$cat abc.sql

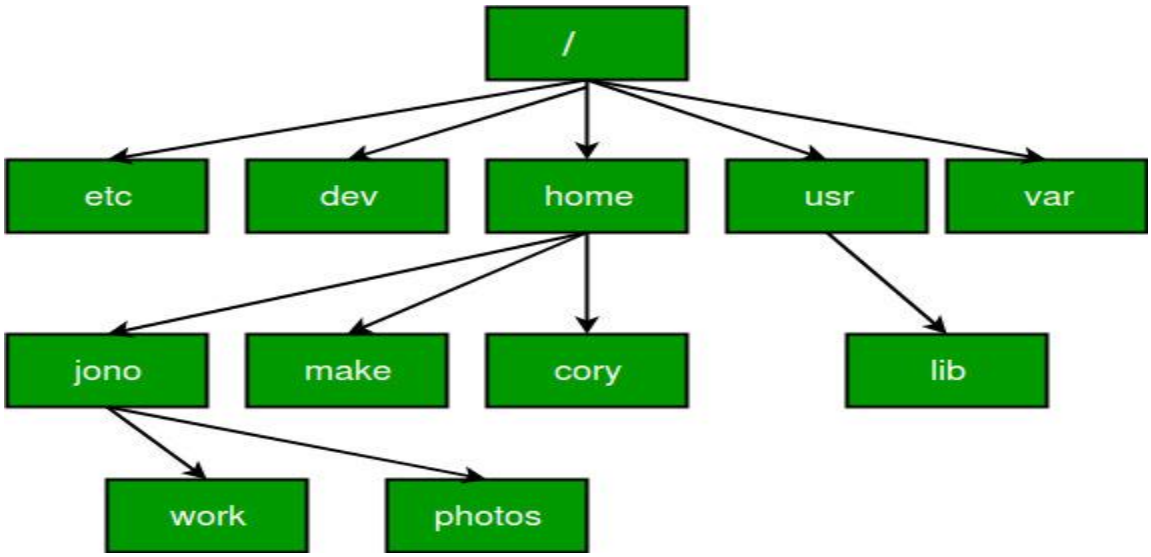
will work only if the fie “abc.sql” exists in your current directory. However, if this file is not present in your working directory and is present somewhere else say in /home/kt , then this command will work only if you will use it like shown below:

cat /home/kt/abc.sql

An absolute path is defined as specifying the location of a file or directory from the root directory(/). In other words,we can say that an absolute path is a complete path from start of actual file system from / directory.

Relative path is defined as the path related to the present working directly(pwd). It starts at your current directory and **never starts with a /** .

To be more specific let's take a look on the below figure in which if we are looking for photos then absolute path for it will be provided as */home/jono/photos* **but assuming that we are already present in jono directory then the relative path for the same can be written as simple** *photos*.



Example of Absolute and Relative Path

Suppose you are currently located in home/kt and you want to change your directory to home/kt/abc. Let's see both the absolute and relative path concepts to do this:

1. **Changing directory with relative path concept :**
2. $pwd
3. /home/kt
4. $cd abc

5. $pwd
6. /home/kt/abc
7. **Changing directory with absolute path concept:**
8. $pwd
9. /home/kt
10. $cd /home/kt/abc
11. $pwd

    /home/kt/abc

Handling Ordinary files in LINUX
Listing Files

To list the files and directories stored in the current directory, use the following command −

$ls

Example

Here is the sample output of the above command −

$ls

```
bin        hosts  lib     res.03
ch07       hw1    pub     test_results
ch07.bak   hw2    res.01  users
docs       hw3    res.02  work
```

The command **ls** supports the **-l** option which would help you to get more information about the listed files –

$ls -l
total 1962188

```
drwxrwxr-x  2 amrood amrood      4096 Dec 25 09:59 uml
-rw-rw-r--  1 amrood amrood     20480 Nov 25  2007 webthumb.tar
-rw-rw-r--  1 amrood amrood      5654 Aug  9  2007 yourfile.mid
-rw-rw-r--  1 amrood amrood    166255 Aug  9  2007 yourfile.swf
drwxr-xr-x 11 amrood amrood      4096 May 29  2007 zlib-1.2.3
$
```

Here is the information about all the listed columns −

First Column − Represents the file type and the permission given on the file. Below is the description of all type of files.

Second Column − Represents the number of memory blocks taken by the file or directory.

Third Column − Represents the owner of the file. This is the Unix user who created this file.

Fourth Column − Represents the group of the owner. Every Unix user will have an associated group.

Fifth Column − Represents the file size in bytes.

Sixth Column − Represents the date and the time when this file was created or modified for the last time.

Seventh Column − Represents the file or the directory name.

Creating a File

$ cat > abc.txt

This is my first File

$

Display Content of a File

You can use the **cat** command to see the content of a file. Following is a simple example to see the content of the above created file −

$ cat filename

This is unix file....I created it for the first time.....

I'm going to save this content in this file.

$

You can display the line numbers by using the -b option along with the cat command as follows −

$ cat -b filename

1 This is unix file....I created it for the first time.....

2 I'm going to save this content in this file.

$

Deleting a File

$rm filel1

rm abc.txt

The 'rm' means remove. This command is used to remove a file. The command line doesn't have a recycle bin or trash unlike other GUI's to

recover the files. Hence, be very much careful while using this command. Once you have deleted a file, it is removed permanently.

Renaming of Files

Both moving and renaming activities can be performed by using single command: mv

$ mv

mv dir1/* dir2

All files of dir1 will be moved to dir2. After executing this command, dir1 will come empty.

Renaming of Files:

mv oldname newname

mv file1.txt file2.txt

file1.txt will be renamed to file2.txt

Renaming of Directories:

mv dir1 dir2

dir1 will be renamed to dir2

Paging Output

more Command:

We can use more command to view file content page by page.

more file1.txt

- It will display first page.
- Enter → To view next line
- Space Bar → To view next page
- q → To quit/exit