Linux Commands

Creating Directories and Files Using Linux CLI

mkdir - Make directory

Create single or a nested directory with `mkdir` command.

```shell
mkdir MyProject
```

The above command will create a brand new folder named as `MyProject` which will be initially empty

```shell
mkdir MyContent/src
```

Here, we are trying to create a src folder inside the MyContent folder, but if the MyContent folder doesn't already exist then this command will throw us an error:

```
mkdir: MyContent: No such file or directory
```

So, if we want that, the command should create the parent directory also in case it is not present and then inside the parent directory create the child directory we can use the `-p` flag altogether.

```bash
mkdir -p MyContent/src
```

Now, linux will create MyContent folder if it doesn't exist and then inside it, there will be a src directory created.

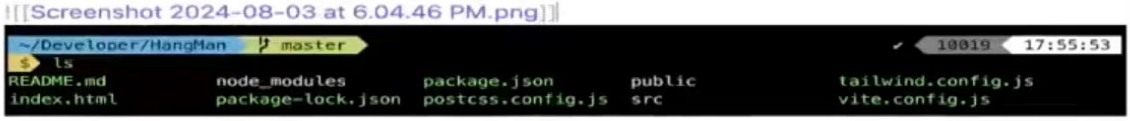
ls - Listing directory content

We can use the `ls` command to explore different way of listing files and folders present inside the current directory.

```shell
ls
```

The above command will list down all the files and folders inside a directory.

[[Screenshot 2024-08-03 at 6.04.46 PM.png]]



If we see, then we don't have much details about the files and folders here. If we want more details like time of creation, user, size etc we can use the `-l` flag.

```bash
ls -l # this will give long listing of the content inside a folder
```

```
~/Developer/HangMan  master                                  10020  17:56:38
$ ls -l
total 424
-rw-r--r--     1 sanketsingh  staff     451 21 Jul 12:08 README.md
-rw-r--r--     1 sanketsingh  staff     361 21 Jul 12:08 index.html
drwxr-xr-x   298 sanketsingh  staff    9536 27 Jul 12:43 node_modules
-rw-r--r--     1 sanketsingh  staff  190806 27 Jul 12:43 package-lock.json
-rw-r--r--     1 sanketsingh  staff     772 27 Jul 12:43 package.json
-rw-r--r--     1 sanketsingh  staff      80 21 Jul 12:12 postcss.config.js
drwxr-xr-x     3 sanketsingh  staff      96 21 Jul 12:08 public
drwxr-xr-x     9 sanketsingh  staff     288 27 Jul 13:07 src
-rw-r--r--     1 sanketsingh  staff     184 21 Jul 12:13 tailwind.config.js
-rw-r--r--     1 sanketsingh  staff     163 21 Jul 12:08 vite.config.js
```

Here the sizes are not in an understandable format, so we can use the `-lh` flag to have sizes represented in human readable form.

[[Screenshot 2024-08-03 at 6.06.33 PM.png]]

```
~/Developer/HangMan  master                                  10022  17:58:17
$ ls -lh
total 424
-rw-r--r--     1 sanketsingh  staff   451B 21 Jul 12:08 README.md
-rw-r--r--     1 sanketsingh  staff   361B 21 Jul 12:08 index.html
drwxr-xr-x   298 sanketsingh  staff   9.3K 27 Jul 12:43 node_modules
-rw-r--r--     1 sanketsingh  staff   186K 27 Jul 12:43 package-lock.json
-rw-r--r--     1 sanketsingh  staff   772B 27 Jul 12:43 package.json
-rw-r--r--     1 sanketsingh  staff    80B 21 Jul 12:12 postcss.config.js
drwxr-xr-x     3 sanketsingh  staff    96B 21 Jul 12:08 public
drwxr-xr-x     9 sanketsingh  staff   288B 27 Jul 13:07 src
-rw-r--r--     1 sanketsingh  staff   184B 21 Jul 12:13 tailwind.config.js
-rw-r--r--     1 sanketsingh  staff   163B 21 Jul 12:08 vite.config.js
```

If we want to list all the hidden files and folders using the `ls` command then we can give a `-a` flag, which will display normal folder and folders along with the hidden ones.

```
~/Developer/HangMan  master                                  10021  17:57:36
$ ls -a
.               .git            index.html      package.json    src
..              .gitignore      node_modules    postcss.config.js  tailwind.config.js
.eslintrc.cjs   README.md       package-lock.json  public       vite.config.js
```

We can also club the usage of multiple flags, like use `-lh` to have detailed listing and human readable sizes along with `-a` to enable this for hidden files and folders also.

[[Screenshot 2024-08-03 at 6.08.07 PM.png]]

```
~/Developer/HangMan  master                                  10023  17:58:48
$ ls -lh -a
total 440
drwxr-xr-x   15 sanketsingh  staff   480B  3 Aug 13:09 .
drwxr-xr-x@ 269 sanketsingh  staff   8.4K  3 Aug 17:46 ..
-rw-r--r--     1 sanketsingh  staff   597B 21 Jul 12:59 .eslintrc.cjs
drwxr-xr-x    12 sanketsingh  staff   384B  3 Aug 13:11 .git
-rw-r--r--     1 sanketsingh  staff   253B 21 Jul 12:08 .gitignore
-rw-r--r--     1 sanketsingh  staff   451B 21 Jul 12:08 README.md
-rw-r--r--     1 sanketsingh  staff   361B 21 Jul 12:08 index.html
drwxr-xr-x   298 sanketsingh  staff   9.3K 27 Jul 12:43 node_modules
-rw-r--r--     1 sanketsingh  staff   186K 27 Jul 12:43 package-lock.json
-rw-r--r--     1 sanketsingh  staff   772B 27 Jul 12:43 package.json
-rw-r--r--     1 sanketsingh  staff    80B 21 Jul 12:12 postcss.config.js
drwxr-xr-x     3 sanketsingh  staff    96B 21 Jul 12:08 public
drwxr-xr-x     9 sanketsingh  staff   288B 27 Jul 13:07 src
-rw-r--r--     1 sanketsingh  staff   184B 21 Jul 12:13 tailwind.config.js
```

0 backlinks

We can use the `-lt` flag to sort our content based on the modification time.

[[Screenshot 2024-08-03 at 6.08.35 PM.png]]

```
~/Developer/HangMan  master                                  10026  18:00:47
$ ls -lt -lh -a
total 440
drwxr-xr-x@ 269 sanketsingh  staff   8.4K  3 Aug 17:46 ..
drwxr-xr-x    12 sanketsingh  staff   384B  3 Aug 13:11 .git
drwxr-xr-x    15 sanketsingh  staff   480B  3 Aug 13:09 .
drwxr-xr-x     9 sanketsingh  staff   288B 27 Jul 13:07 src
-rw-r--r--     1 sanketsingh  staff   186K 27 Jul 12:43 package-lock.json
-rw-r--r--     1 sanketsingh  staff   772B 27 Jul 12:43 package.json
drwxr-xr-x   298 sanketsingh  staff   9.3K 27 Jul 12:43 node_modules
-rw-r--r--     1 sanketsingh  staff   597B 21 Jul 12:59 .eslintrc.cjs
-rw-r--r--     1 sanketsingh  staff   184B 21 Jul 12:13 tailwind.config.js
-rw-r--r--     1 sanketsingh  staff    80B 21 Jul 12:12 postcss.config.js
-rw-r--r--     1 sanketsingh  staff   163B 21 Jul 12:08 vite.config.js
drwxr-xr-x     3 sanketsingh  staff    96B 21 Jul 12:08 public
-rw-r--r--     1 sanketsingh  staff   361B 21 Jul 12:08 index.html
-rw-r--r--     1 sanketsingh  staff   253B 21 Jul 12:08 .gitignore
```

0 backlinks

So, here we have clubbed `-lt` with other flags to combine their usecases.

pwd - Print working directory

If we want to identify the current directory where we are executing the linux commands form the terminal then we can use the `pwd` command.

```shell
pwd
```

touch - Create empty files in Linux

To create a brand new empty file in the current working directory using linux command line interface we can use the `touch` command where we can write `touch` and mention name of the file we want to create and it will create a brand new file for us.

```shell
touch file1.txt
```

One advanced usecase is that, if we want to generate multiple files, ending with some sequence number then we can write

```bash
touch file{1..5}.txt
```

This command will create 5 files named as `file1.txt`, `file2.txt`, `file3.txt` and so on.

rm - Command used to remove files

If we want to delete an already existing file we can use the `rm` command which just takes the file name and then, if it exists deletes it.

```shell
rm file.txt
```

If we want to delete a directory then the `rm` command will not be able to directly delete the directory. We can use the `-r` flag in order to recursively delete the directory and everything like subfolders and files inside the directory.

cd - Change Directory

In linux if we want to move from one folder to another using the command line then we can use the `cd` command. This command will take us to the desired directory. There are multiple ways to use this command.

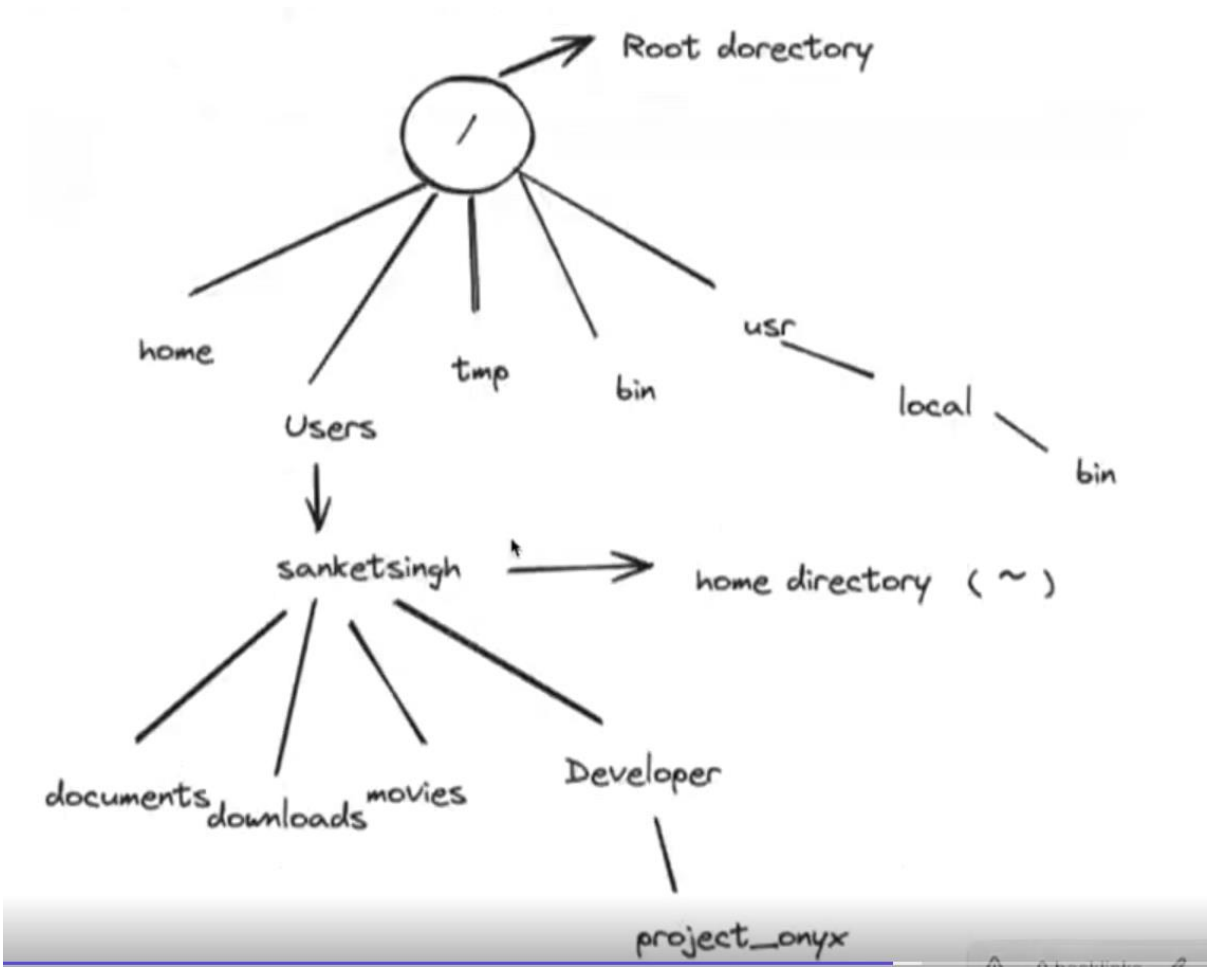
```shell
cd path_to_the_folder
```

To give the path to the folder we have two ways:

1. Absolute path
2. Relative path

To understand this path mechanism we need to know basic file structure in linux.

Typical Structure of Linux File Architecture



In Linux, at the top we have a root directory represented by `/` . Then we have a bunch of folders inside it. One of the folders is `Users` which has one more folder with your username say `sanketsingh` . This directory that starts from root and goes to sanketsingh i.e. `/Users/sanketsingh` is called as `home` directory which is in short form represented by ~ .

Relative Path

Relative path means that standing inside one folder how should we navigate to move to another folder.

Fox example in the image above let's say we are inside the `Developer` folder. Form this we want to go to `Project_Onyx`

```shell
cd Project_Onyx
```

Because Project_onyx is directly inside the Developer folder, we just moved there from the Developer folder. If we are inside any folder and we want to go back to the parent folder we can use `..` with cd.

```Shell
cd ..
```

If we are in Project_Onyx this `cd ..` will bring us back to the Developer folder.
If we put multiple `..` separated by `/` then we can come out of multiple nested folders.

Example: If we are inside the project_onyx, then

```Shell
cd ../..
```

Doing this will bring us first out from project_onyx to Developer and then from Developer to Home directory.

And in this way, you define the path from one folder to another. Say from Project_onyx we want to come to bin folder inside local

```Shell
cd ../../../../usr/local/bin
```

So here, we go 4 level up to the root directory and then from there navigated to bin.

Absolute path

In an absolute path, we give the complete path of a folder from anywhere and then linux will automatically transport us to that path, but giving complete path is mandatory.
So if we are inside Project_Onyx and we want to go to local/bin we can say

```Shell
cd /usr/local/bin
```

Here the first `/` represents the root, then `usr` then `local` and then `bin`.

In this way we don't need to navigate from one folder to another step by step.

If from anywhere we need to come back to the home directory we say

```Shell
cd ~
```

or

```Shell
cd /Users/sanketsingh
```

Note

In linux files and folders are separated by `/` (forward slash) but in windows they are separated by `\` (back slash).

man - Manual

If we want to see the details of any command, around how to use it and what it does, we can use the `man` command.

```Shell
man ls
```

And this will show us the details of the `ls` command that what it is, how to use it, all the flags associated etc.

cat - Concatenate and display

If we want to display results of a file then we can use the cat command. It will print wehatever is written inside the file.
And if we want to see contents of multiple files, then we can just give space separated information around name of the files and it will take the contents of the files and display it by concatenating it together and show it

```bash
cat file1.txt file2.txt
```

whoami - Display logged in user

If we want to see who is the logged in user in the machine, we can use the `whoami` command.

```Shell
whoami
```

This will just return the username of the logged in user.

uname - Display system information

If we want to see some system details, like kernel name, version etc then we can use this command

```Shell
uname # Show the kernel name
uname -r # Shows the kernel version only
uname -s # SHows the kernel name
uname -a # Deisplay all system info
```

echo - Printing something in linux CLI

If we want to print something in the CLI interface we can use `echo`.

```shell
echo "Hello world"
```

This will just output Hello World in the terminal.

>> - Dump the output of a command in a file

So if we want to dump / append output of a command in a file we can just write the command put a pair angle brackets (`>>`) and then give a file name, if the file doesn't exist it will create a file and dump the output of the command in that file, else if the file already exists it is going to just keep on append the outputs in the file.

```shell
ls >> sanket.txt
echo "Hello world" >> sanket.txt
```

This will append the output of ls and echo command in the text file.