

Bash and Terminal (for ubuntu and mac)

If we are using ubuntu terminal in windows then first we have to do cd..../ and then we have to go inside mnt directory and in this directory we have C drive where we can find our usual windows files and folders

1. **pwd** - print working directory

Terminal is nothing but another interface to do things in your machine (just like GUI).
Where is terminal

```
Ubuntu is already installed.  
Launching Ubuntu...  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
nthapa0000@LAPTOP-GCLF70VR:~$ pwd  
/home/nthapa0000  
nthapa0000@LAPTOP-GCLF70VR:~$
```

How can I run command such that i am in desktop folder (navigate to it)

2. **cd** - change directory

```
PS C:\Users\NTC> pwd  
  
Path  
----  
C:\Users\NTC  
  
PS C:\Users\NTC> cd  
PS C:\Users\NTC> cd Desktop  
PS C:\Users\NTC\Desktop>
```

How do I go back??

One folder before it by using the command **cd ..**

PS C:\Users\NTC>

Till now we have just know the present working directory and how to change the directory , this doesn't seem very useful till now, but here comes the next command

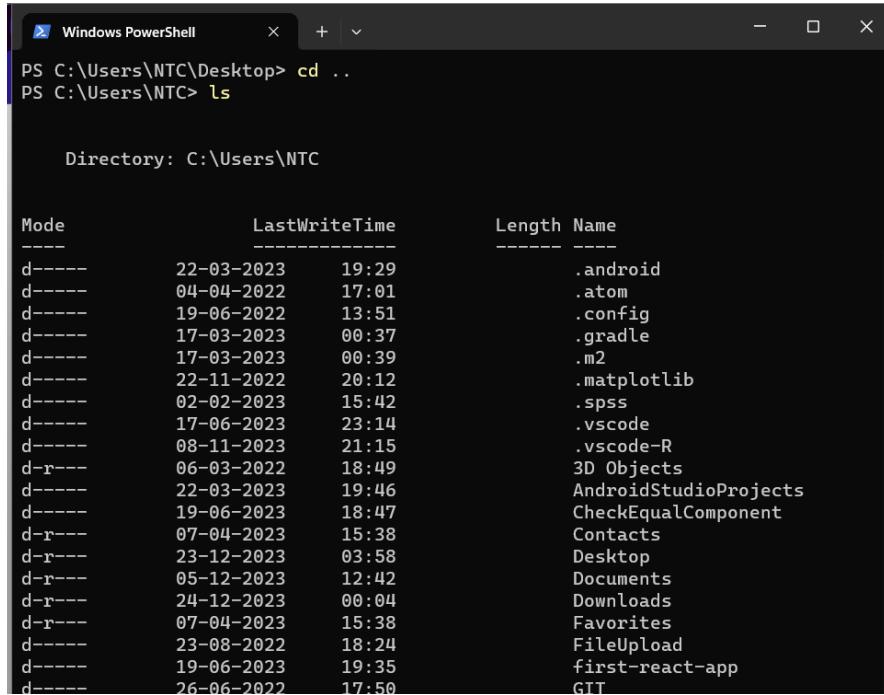
While using cd we can inside more than one folder(nested) at same time

PS C:\Users\NTC> cd Desktop\dev

If we want to go to two folders back(two levels back) i will use.

```
PS C:\Users\NTC\Desktop\dev> cd ..
```

3. **ls** : is used to list all the folders inside it
The command displays a list of files and directories in the current directory.

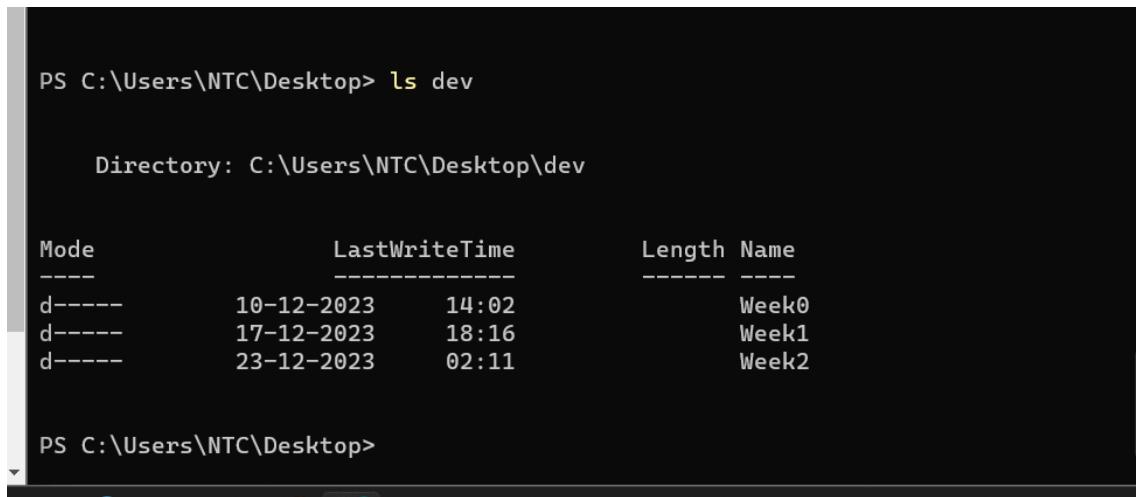


```
PS C:\Users\NTC\Desktop> cd ..
PS C:\Users\NTC> ls

Directory: C:\Users\NTC

Mode                LastWriteTime      Length Name
----                -----          ---- 
d-----        22-03-2023     19:29           .android
d-----        04-04-2022     17:01           .atom
d-----        19-06-2022     13:51           .config
d-----        17-03-2023     00:37           .gradle
d-----        17-03-2023     00:39           .m2
d-----        22-11-2022     20:12           .matplotlib
d-----        02-02-2023     15:42           .spss
d-----        17-06-2023     23:14           .vscode
d-----        08-11-2023     21:15           .vscode-R
d-r---        06-03-2022     18:49           3D Objects
d-----        22-03-2023     19:46           AndroidStudioProjects
d-----        19-06-2023     18:47           CheckEqualComponent
d-r---        07-04-2023     15:38           Contacts
d-r---        23-12-2023     03:58           Desktop
d-r---        05-12-2023     12:42           Documents
d-r---        24-12-2023     00:04           Downloads
d-r---        07-04-2023     15:38           Favorites
d-----        23-08-2022     18:24           FileUpload
d-----        19-06-2023     19:35           first-react-app
d-----        26-06-2022     17:50           GIT
```

Using the terminal we can navigate quite fast



```
PS C:\Users\NTC\Desktop> ls dev

Directory: C:\Users\NTC\Desktop\dev

Mode                LastWriteTime      Length Name
----                -----          ---- 
d-----        10-12-2023     14:02           Week0
d-----        17-12-2023     18:16           Week1
d-----        23-12-2023     02:11           Week2

PS C:\Users\NTC\Desktop>
```

We were on the desktop then we saw set of files and directories and then i chose, to see whats inside a folder dev on the desktop.

ls -l : what each folder within directory contain we can use this command.

```
webs
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop$ cd dev
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ ls -l
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 10 14:02 Week0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 17 18:16 Week1
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 23 02:11 Week2
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ ls -l week1
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 15 20:57 Assignment01-js
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 15 20:57 Assignmentweek-1
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 18 02:31 Async_Await_Promises
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 15 23:40 BasicsJsApi
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 16 15:55 FunctionNCallbacks
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 15 00:08 JSFoundation
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ ls -l week2
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 23 13:11 Http_Server
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 20 00:04 asyncJs
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$
```

How can i get the list of directories and know which was the last modified one

ls -t

```
./Week2/asyncJs:
.a.txt practice.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ ls -t
Week2 Week1 Week0
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ cd week1
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week1$ ls -t
Async_Await_Promises BasicsJsApi Assignment01-js
FunctionNCallbacks Assignmentweek-1 JSFoundation
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week1$ |
```

Lets say i want to see what is present inside each sub directory , we will use command

ls -R

This will give details about sub directories as well

```

drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 23 13:11 Http_Server
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 20 00:04 async.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ ls -R
.:
Week0 Week1 Week2

./Week0:
Week_0_Slides_.pdf ZerodhaLandingPage

./Week0/ZerodhaLandingPage:
'Screenshot (131).png' index1.html logo.svg
index.html landing_page_zerodha.png

./Week1:
Assignment01-js Async_Await_Promises FunctionNCallbacks
Assignmentweek-1 BasicsJsApi JSFoundation

./Week1/Assignment01-js:
README.md easy hard medium package-lock.json package.json tests

./Week1/Assignment01-js/easy:
anagram.js expenditure-analysis.js findLargestElement.js

./Week1/Assignment01-js/hard:
calculator.js todo-list.js

./Week1/Assignment01-js/medium:
countVowels.js palindrome.js times.js

./Week1/Assignment01-js/tests:
anagram.test.js expenditure-analysis.test.js todo-list.test.js
calculator.test.js findLargestElement.test.js
countVowels.test.js palindrome.test.js

./Week1/Assignmentweek-1:
offline-class-1

./Week1/Assignmentweek-1/offline-class-1:
Level-1 Level-2

```

We can also even combine these commands like **ls -lt**

```

nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ cd ..
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ ls -lt
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 23 02:11 Week2
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 17 18:16 Week1
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 10 14:02 Week0
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ 

```

The newest one are at the beginning(Which we worked on)

Not all the file are visible when we work on private key etc

To see the hidden file(eg git, yarn) by **ls -la** and then directory if we want to go even further into sub directory

```

nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ cd Week1
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/Week1$ ls -la
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 17 18:16 .
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 19 23:08 ..
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 15 20:57 Assignment01-js
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 15 20:57 Assignmentweek-1
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 18 02:31 Async_Await_Promises
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 15 23:40 BasicsJsApi
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 16 15:55 FunctionNCallbacks
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 15 00:08 JSFoundation
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/Week1$ 

```

Hidden files are those that end with a dot

Now if we see the hidden file inside all the sub-directories also we will use the command **ls -lRa**

```

nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/Week1$ cd ..
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ ls -lRa
.:
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 19 23:08 .
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 00:29 ..
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 10 14:02 Week0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 17 18:16 Week1
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 23 02:11 Week2

./Week0:
total 972
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 10 14:02 .
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 19 23:08 ..
-rwxrwxrwx 1 nthapa0000 nthapa0000 992412 Dec 10 14:01 Week_0_Slides_.pdf
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 10 10:14 ZerodhaLandingPage

./Week0/ZerodhaLandingPage:
total 236

```

If we want the reverse order in which the files were modified we will use the command **ls -lr**.

If we want to look all the JSON files in the directory

ls -IR | grep .json

```

nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls -lR | grep .json
n
-rwxrwxrwx 1 nthapa0000 nthapa0000 25553 Dec 23 13:13 package-lock.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 306 Dec 23 13:13 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 1157 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 879 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 1472 Dec 23 13:13 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 5299 Dec 23 13:13 json.js
-rwxrwxrwx 1 nthapa0000 nthapa0000 959 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 2208 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 1200 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 1075 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 1140 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 492 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 321 Dec 23 02:15 component.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 1138 Dec 23 02:15 package.json
-rwxrwxrwx 1 nthapa0000 nthapa0000 2082 Dec 23 02:15 package.json

```

We can also do it higher directory since we are doing it recursively

Using wildcards

ls *.js

Will find us all the js files inside a directory

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls *.js
ls: cannot access '*.js': No such file or directory
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls
Http_Server  asyncJs
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cd asyncJs
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/asyncJs$ ls *.js
practice.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/asyncJs$
```

ls -l to see sub-directories

ls -IR directory

We can use wildcards to specify the type of files

ls Zoo*

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls no*
ls: cannot access 'no*': No such file or directory
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls async*
a.txt  practice.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls a*
a.txt  practice.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$
```

Here we can see that using wildcard we can look into files of the folder whose name starts with a(which we wrote in command)

Now suppose you want to know what are the files/folders present in the parent directory

We can use the command **ls ..**

```
a.txt  practice.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls ..
Week0  Week1  Week2
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls ../../
'- give post and.txt'
'- use link icon for original post.txt'
01-starting-project
01-starting-setup Unit4
'21bma029 - Chrome.lnk'
ABC
'Adobe Illustrator 2022.lnk'
'Adobe Photoshop 2021 v22.5.3.561 (x64) Multilingual (Pre-Activated) [FileOR]' 
'Adobe Photoshop 2021.lnk'
Atom.lnk
'Basic Dsa'
Bootstrap
CodeCp
CourseFiles
DSA_LABPAPER
D_S_A
DebuggingReactApp
Dev
'Exercise Files - Photoshop Essentials!'
GATE
Grammarly.lnk
```

Note: while navigating the directory we can use the tab button to autocomplete the folder name.

4. **mkdir** (Make Directory)

(GUI way to create a new folder is to right-click on and press new folder)

Command to create a new directory

```
PS C:\Users\NTC\Desktop> mkdir test

Directory: C:\Users\NTC\Desktop

Mode                LastWriteTime         Length Name
----                -              -          -
d-----        24-12-2023     00:29           test

PS C:\Users\NTC\Desktop> |
```

This will create a new folder inside the Desktop

Creating a new folder.

```
BAWEEEEEE!!!!!
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ mkdir BashDemo
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls
BashDemo BashDemo.txt Http_Server asyncJs
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ |
```

We can also mix two command for example

```
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cd BashDemo
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mkdir LetsLearn && cd LetsLearn
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo/LetsLearn$ ../
-bash: ../: Is a directory
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo/LetsLearn$ cd ..
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
LetsLearn
nthaapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

Let's see how can we create a nested directory

We will use **mkdir -p**

```

[nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ cd ..
nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2$ ls
BashDemo BashDemo.txt Http Server async.js
nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2$ cd BashDemo
nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mkdir -p frontend/scripts
nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mkdir -p frontend/css
nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mkdir -p frontend/img
nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mkdir -p frontend/statics
nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
[nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2$ ls -l frontend
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:57 css
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:58 img
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:57 scripts
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:58 statics
[nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2$ ls -LR frontend
frontend:
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:57 css
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:58 img
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:57 scripts
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:58 statics

frontend/css:
total 0

frontend/img:
total 0

frontend/scripts:
total 0

frontend/statics:
total 0
[nthapa0000@LAPTOP-GCLF70VR:~/mnt/c/Users/NTC/Desktop/dev/week2$ ]

```

5. **New-Item** (This command will be used in windows terminal to create a new file) (**touch** in ubuntu)

```

PS C:\Users\NTC\Desktop\test> New-Item index.js

Directory: C:\Users\NTC\Desktop\test

Mode          LastWriteTime        Length Name
----          -----          ----
-a---  24-12-2023    00:33           0 index.js

PS C:\Users\NTC\Desktop\test>

```

It will create an index.js file inside the test folder.

Example

To create a new file called index.txt inside the test folder

```
PS C:\Users\NTC\Desktop\test> New-Item index.txt

Directory: C:\Users\NTC\Desktop\test

Mode                LastWriteTime     Length Name
----                -----          ----  --
-a----    24-12-2023      00:35           0  index.txt

PS C:\Users\NTC\Desktop\test>
```

touch

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev$ cd week2
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ touch BashDemo.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls
BashDemo.txt  Http_Server  asyncJs
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$
```

6. cat (this command will be used to see(display) the contents of a file.)

Example:

Suppose we have two files inside the test folder and want to read the contents of the index.txt file.

```
PS C:\Users\NTC\Desktop\test> cat index.txt
Hi there here is Nishu Wuaah!
PS C:\Users\NTC\Desktop\test> |
```

From the desktop folder, we can also use a relative directory to access the content of the file.

```
PS C:\Users\NTC\Desktop\test> cd ..
PS C:\Users\NTC\Desktop> cat test/index.txt
Hi there here is Nishu Wuaah!
PS C:\Users\NTC\Desktop> |
```

cat command can also be used to add data to a new you have created or concatenate data to the existing file. (**ctrl + d** to go out of writing)

cat > BashDemo.txt

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ls
BashDemo.txt  Http_Server  asyncJS
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cat > BashDemo.txt
xt
Baweee Doon , Baweee Doon mai idhar
Baweee Doon , Baweee Doon mai idhar
Khooni jhullun hai idhar
Baweee Doon , Baweee Doon mai idhar
Baweee Doon , Baweee Doon mai idhar
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cat BashDemo.txt
Baweee Doon , Baweee Doon mai idhar
Baweee Doon , Baweee Doon mai idhar
Khooni jhullun hai idhar
Baweee Doon , Baweee Doon mai idhar
Baweee Doon , Baweee Doon mai idhar
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cat > BashDemo.txt
xt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cat > BashDemo.txt
xt
Lets see what happens?
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cat BashDemo.txt
Lets see what happens?
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$
```

If we use the `cat >` command again on the same file previous data will be overwritten

If we want we will use `>>` operator

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cat BashDemo.txt
Lets see what happens?
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cat >> BashDemo.txt
BAWEEEEEE!!!!
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ cat BashDemo.txt
Lets see what happens?
BAWEEEEEE!!!!
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ |
```

7. `vi` (vim command) (ubuntu command)(we can use it in window terminal if we install it)
Let us edit file, We have to go to insert to edit the file (press `i` to get in insert mode)

Now to escape this is another task :

Press **Esc**(to exit the insert mode) type :**q!** This will exit the vi mode. It will exit without saving to save we have to type:**wq!**

8. **mv**: lets us move the file

```
PS C:\Users\NTC\Desktop\test> mv index.txt C:\Users\NTC\Desktop\test\test-2  
PS C:\Users\NTC\Desktop\test> ls
```

Directory: C:\Users\NTC\Desktop\test

Mode	LastWriteTime	Length	Name
d----	24-12-2023 01:16		test-2
-a---	24-12-2023 00:33	0	index.js

PS C:\Users\NTC\Desktop\test>

We can move files/folder in the single step wherein gui it may take several steps

We can also use this command to rewrite the file name

```
mv old_name.js new_name.js
```

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
[LetsLearn] frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ touch script.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
[LetsLearn] Frontend script.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mv script.js runtime_script.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
[LetsLearn] Frontend runtime_script.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

Moving a file inside another folder

```
LetsLearn Frontend runtime_script.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mv runtime_script.js frontend/scripts
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls frontend/scripts
runtime_script.js
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

While moving we can also change the name

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ touch style.css
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
[LetsLearn] Frontend style.css
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mv style.css frontend/css/frontend.css
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls frontend/css
frontend.css
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

Also generally it is very fast compared to GUI moving a file

We also need to copy files sometime

9. **cp** : (let us copy the file/folder)

```
cp source_file.txt destination_file.txt
```

We can also copy directories recursively from one place to another.

```
PS C:\Users\NTC\Desktop\test\test-2> cp index.txt test
PS C:\Users\NTC\Desktop\test\test-2> ls

Directory: C:\Users\NTC\Desktop\test\test-2

Mode                LastWriteTime        Length Name
----                -----          ---- - 
-a----       24-12-2023     00:37            29 index.txt
-a----       24-12-2023     00:37            29 test

PS C:\Users\NTC\Desktop\test\test-2>
```

Now we will create a file and then make a copy of it inside another file

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
[Let'sLearn] Frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ touch index.html
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ cp index.html frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls frontend
css img index.html scripts statics
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

Now copying a folder inside another folder

We need to use **-r**

Example 1: We create a folder called test inside a folder called frontend. Then we create a folder called backend in the parent directory and then copy the folder test from frontend to backend

```

nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mv style.css frontend/css/frontend.css
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls frontend/css
frontend.css
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
Let'sLearn Frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ touch index.html
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ cp index.html frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls frontend
css img index.html scripts statics
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
Let'sLearn Frontend index.html
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ cd frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
css img index.html scripts statics
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mkdir test
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
css img index.html scripts statics test
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ cd ..
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
Let'sLearn Frontend index.html
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ mkdir backend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
Let'sLearn backend frontend index.html
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ cp -r frontend/test backend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls backend
test
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

Another example

```

PS C:\Users\NTC\Desktop\test\test-2> cd ../
PS C:\Users\NTC\Desktop\test> mkdir test3

Directory: C:\Users\NTC\Desktop\test

Mode          LastWriteTime    Length Name
----          -----          ---- 
d-----      24-12-2023     01:30   test3

PS C:\Users\NTC\Desktop\test> ls

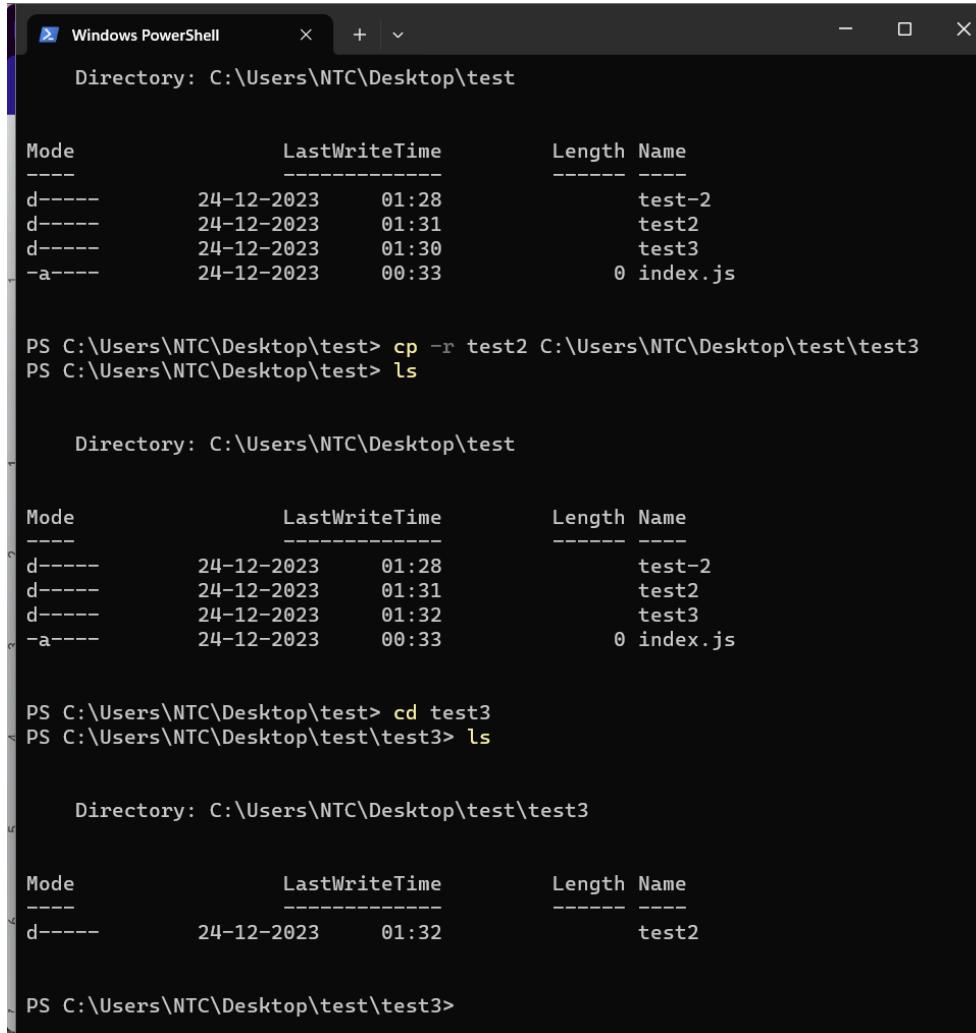
Directory: C:\Users\NTC\Desktop\test

Mode          LastWriteTime    Length Name
----          -----          ---- 
d-----      24-12-2023     01:28   test-2
d-----      24-12-2023     01:30   test3
-a---      24-12-2023     00:33   0 index.js

PS C:\Users\NTC\Desktop\test> cp -r test3 test2
PS C:\Users\NTC\Desktop\test> ls
```

Here **-r** stands for recursive we have to type it when we are copying a folder since a folder can have many files inside it.

Operations from the terminal are way faster than simple GUI



Windows PowerShell

```
Directory: C:\Users\NTC\Desktop\test

Mode                LastWriteTime         Length Name
----                -----          ---- -  
d-----        24-12-2023     01:28            test-2
d-----        24-12-2023     01:31            test2
d-----        24-12-2023     01:30            test3
-a----        24-12-2023     00:33           0 index.js

PS C:\Users\NTC\Desktop\test> cp -r test2 C:\Users\NTC\Desktop\test\test3
PS C:\Users\NTC\Desktop\test> ls

Directory: C:\Users\NTC\Desktop\test

Mode                LastWriteTime         Length Name
----                -----          ---- -  
d-----        24-12-2023     01:28            test-2
d-----        24-12-2023     01:31            test2
d-----        24-12-2023     01:32            test3
-a----        24-12-2023     00:33           0 index.js

PS C:\Users\NTC\Desktop\test> cd test3
PS C:\Users\NTC\Desktop\test\test3> ls

Directory: C:\Users\NTC\Desktop\test\test3

Mode                LastWriteTime         Length Name
----                -----          ---- -  
d-----        24-12-2023     01:32           0 test2

PS C:\Users\NTC\Desktop\test\test3>
```

10. nvm (node version manager) let us install the node

11. npm (node package manager)

The world has a lot of libraries, locally we create index.js files, a lot of time index.js will require complex/external dependencies(packages) (which need to be installed or imported) eg express, it is present in the npm registry, to bring these packages we use command **npm install express** it will create a new folder in the working directory by the name of **node_modules** and there will be a lot of folders and one of the important ones is **express** (since we imported it) and also there is a file created by the name of **package.json**

There are thousands of libraries/packages and to bring them locally we have to use npm command

To see the packages we see this

<https://npmjs.com/package/express>

12. node (on typing in the terminal we get into the shell, where we can write the Node.js/Js code we can exit from here by clicking **ctrl+ c** twice)

We can also run js code, by making a new file(a.js) through the terminal(**New-item**) and then editing/inserting in the file with **vi command** and then using **node** command to run it
node a.js

```
PS C:\Users\NTC\Desktop\test> New-item a.js

Directory: C:\Users\NTC\Desktop\test

Mode                LastWriteTime         Length Name
----                <-----              ----- 
-a----       24-12-2023     01:44            0 a.js

PS C:\Users\NTC\Desktop\test> cat a.js
console.log("Hello Everyone!")
PS C:\Users\NTC\Desktop\test> node a.js
Hello Everyone!
PS C:\Users\NTC\Desktop\test> |
```

13. clear (to clear terminal)

The job of the shell or command line interface is to understand and interpret the prompt and kernel execute it.

14. rm

Using **rm** to remove a file

```
less
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
Let'sLearn backend Frontend index.html
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ rm i
ndex.html
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
Let'sLearn backend Frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

If we want to delete the folder

rm -r folder_name

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
Let'sLearn backend Frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ rm -
r backend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls
Let'sLearn frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

15. chmod (Change File permissions)

Modify the read, write, and execute permissions of a file.

Permission

There are three permissions for a file which are read, write, and execute

There are three sets of this for the user, group, and other permission

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ ls -
l
total 0
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 24 23:54 Let'sLearn
drwxrwxrwx 1 nthapa0000 nthapa0000 4096 Dec 25 00:33 Frontend
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

Two ways we can decide what permissions for a particular file/folder

1. Symbolic way: r,x,w
2. Numeric way (4: read, 2: write and 1:execute)

chmod needs to know a few things

1. Who is the function for users, groups, and others (**ugo**)
2. It needs to know whether we are adding(+) or revoking(-) the permission
3. What type of permissions (**rwx**) read, write or execute
4. Folder and file you want to set permission for. If the folder you have to add **-R** (recursive)

Examples

chmod u-x file1.txt

chmod g+wx nescript.js

chmod 664 1234(foldername)

chmod 777 1234 (all permissions)

16. echo (display a message in the terminal)

echo "Hello world"

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ echo
"Hello World"
Hello World
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

Let's display our current path

echo \$PATH

```
Hello World
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ echo
$PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr
/local/games:/usr/lib/wsl/lib:/mnt/c/WINDOWS/system32:/mnt/c/WINDOWS:/mnt/c/
WINDOWS/System32/Wbem:/mnt/c/WINDOWS/System32/WindowsPowerShell/v1.0:/mnt/c
/Windows/System32/OpenSSH:/mnt/c/Program Files/Git/cmd:/mnt/c/Program Files
/nodejs:/mnt/c/Program Files/MATLAB/R2022b/bin:/mnt/c/Program Files/MySQL/M
ySQL Shell 8.0/bin:/mnt/c/Users/NTC/AppData/Local/Programs/Python/Python311
/Scripts:/mnt/c/Users/NTC/AppData/Local/Programs/Python/Python311/:/mnt/c/U
ser/NTC/AppData/Local/Programs/Python/Python37-32/Scripts:/mnt/c/Users/NTC
/AppData/Local/Programs/Python/Python37-32:/mnt/c/Users/NTC/AppData/Local/M
icrosoft/WindowsApps:/mnt/c/Users/NTC/AppData/Local/Programs/Microsoft VS Co
de/bin:/mnt/c/Users/NTC/AppData/Local/atom/bin:/mnt/c/Users/NTC/AppData/Loca
l/Microsoft/WindowsApps:/mnt/c/Users/NTC/AppData/Local/Programs/Hyper/resour
ces/bin:/mnt/c/Users/NTC/AppData/Roaming/npm:/mnt/c/MinGW/bin:/snap/bin
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

Miscellaneous commands

1. head (let us see the first ten rows of a file)

We created a file called newfile.txt and put some entries

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ cat
> newfile.txt
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
```

Let see the **head** command

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ head
newfile.txt
1
2
3
4
5
6
7
8
9
10
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDem Show desktop
```

Let's say we want to see the top 20 rows from a file

head -20 newfile.txt

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ head  
-20 newfile.txt  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

2. tail (let us see the last ten rows of a file)

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ tail  
newfile.txt  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

What if we want to few certain sections of file only, we will combine both commands

We will use pipe operator (|)

command 1 | command 2

Whatever comes from command 1 flows to command 2

Example: We want 5 lines before the last 10 lines of the file

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ tail  
-15 newfile.txt | head -5  
10  
11  
12  
13  
14  
15  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ |
```

3. wc

To see the no. of lines, words, and no.of characters,s, etc of a file

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ wc -n  
ewfile.txt  
24 24 63 newfile.txt  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

4. grep

To find the no. of occurrences of words/files etc

It is used to find patterns and certain phrases within a file or directory.

```
grep "1" newfile.txt
```

```
grep "1" newfile.txt | wc
```

```
grep "1" newfile.txt | wc -l ( count of no. of line )
```

```
24 24 63 newfile.txt  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ grep  
"1" newfile.txt  
1  
10  
12  
13  
14  
15  
16  
17  
18  
19  
21  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ grep  
"1" newfile.txt | wc  
11 11 32  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ grep  
"1" newfile.txt | wc -l  
11  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

grep -c "1" newfile.txt (no. of the line where it occurred) (case sensitive)

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ grep  
-c "1" newfile.txt  
11  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

grep -h "1" newfile.txt (will return all the lines where it matched) (Case sensitive)

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$ grep  
-c "1" newfile.txt  
11  
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo$
```

grep -hi “1” newfile.txt (will return all the lines where it matched and now it is **case insensitive**)

Now I want to see the occurrence of “1” in a directory

grep -hir “1” (it will tell where it occurred because of -r)

```
    "version": "2.0.1",
    "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.
1.tgz",
    "integrity": "sha512-RwNA9Z/7PrK06rYLIzFMlaF+l73iwpzsqRIFgbMLbTc
LD6cOao82TaWefPXQvB2FOC4AjuYSEndS7N/mTCbkdQ==",
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/toidentifier/-/toidentif
ier-1.0.1.tgz",
    "integrity": "sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXp
fpuwjKenlSox/20/BTLZUtEe+JG7s5YhEz608PlAHRA==",
    "version": "1.6.18",
    "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18
.tgz",
    "integrity": "sha512-TkRkr9sUTxEH8MdFuCSP7VizJyzRNMjj2J2do
2Jr3Kym598JVdEksuzPQcnLFPW4kyQ+iA+ma9BGm06XQBy8g==",
    "mime-types": "~2.1.24"
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tg
z",
    "integrity": "sha512-pjy2bYhSsufwWLkwPc+l3cN7+wuJLK6uz0YdJE0lQDb
16jo/YlPi4mb8agUKVC8BF7V8NuzyPNqRksA3hztkQ==",
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/utils-merge/-/utils-merg
e-1.0.tgz",
    "integrity": "sha512-pMZTvIkT1d+TFGvDOqod0clx0QWkkgi6Tdoa8gC8ffG
AAqz9pzPTZWAbbsHHoED/zMtkv/VoYTyyShUn81hA==",
    "version": "1.1.2",
    "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
    "integrity": "sha512-BNGbWLfd0eUpabhlxUVm0j8uuvREyTh5ovRa/dyow/B
qAbZJyC+5FU+IzQOzmAKzYqYRAISoRhdQr3eIZ/PXqg==",
    "version": "1.0.0",
    "test": "echo \"Error: no test specified\" && exit 1"
    "body-parser": "^1.20.2",
    "express": "4.18.2"
  https://n0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2$ ^C
```

Now we want to match the whole word not see where it occurred in between

grep -hinw “1” newfile.txt

```
n0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2[Bashdemo
$ grep -hinw "1" newfile.txt
1:1
n0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2[Bashdemo
$ grep -hiw "1" newfile.txt
1
n0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2[Bashdemo
$
```

-o only give matched words

history to view all the commands you run.

NOTE:

Bash is also a language we can also automate this process in the form of a script
Example:

On the first line of our script, we must specify which interpreter we would like to use to parse our script. In this scenario, it is Bash. Thus, we must put the shebang in the first line of our script.

```

$ ls
Let'sLearn frontend newfile.txt permission.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ touch newscript.sh
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ ls
Let'sLearn Frontend newfile.txt newscript.sh permission.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ cat > newscript.sh
#!/bin/bash
echo "Hello World!"
mkdir automated_dir
cd automated_dir && touch newscript_file.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ cat newscript.sh
#!/bin/bash
echo "Hello World!"
mkdir automated_dir
cd automated_dir && touch newscript_file.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ bash newscript.sh
Hello World!
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ ls
Let'sLearn frontend newscript.sh
automated_dir newfile.txt permission.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ ls -l automated_dir
total 0
-rwxrwxrwx 1 nthapa0000 nthapa0000 0 Dec 25 03:01 newscript_file.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ ls automated_dir
newscript_file.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/Bashdemo
$ |

```

Here we created a file newscript.sh file and then executed it by command **bash newscript.sh**

Let's use grep in **log.txt** (log file)

First we will make a file called log.txt and do the following operations

- Looking for ERROR in the log.txt
- grep “ERROR” log.txt

```

nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$ touch log.txt
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$ ls
LetsLearn  frontend  newfile.txt  permission.txt
automated_dir  log.txt  newscript.sh
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$ cat > log.txt
Timestamp      Category      Message
1598843202    INFO        Booting up system
1598843402    INFO        Booting up critical service: Authorization
1598863888    ERROR       User anonymous attempt to access protected re
source without credentials
1598863901    ERROR       Requested resource not found
1598864411    INFO        User admin logged out
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$ grep "ERROR" log.txt
1598863888    ERROR       User anonymous attempt to access protected re
source without credentials
1598863901    ERROR       Requested resource not found
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$
```

- Lets get all the information except INFO (generally in log file we want to see all the information except the INFO)

grep -v "INFO" log.txt

```

nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$ grep -v "INFO" log.txt
Timestamp      Category      Message
1598863888    ERROR       User anonymous attempt to access protected re
source without credentials
1598863901    ERROR       Requested resource not found
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$
```

- Now suppose we want to see 5 line after the error message

grep -A 5 ERROR log.txt

```

nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$ grep -A 5 "ERROR" log.txt
1598863888    ERROR       User anonymous attempt to access protected re
source without credentials
1598863901    ERROR       Requested resource not found
1598864411    INFO        User admin logged out
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$ |
```

5. sed (We can replicate what we can do with the grep command)

- Looking for ERROR in the log.txt

```
sed -n '/ERROR/ p' log.txt
```

sed's substitute command has the following structure:

's/pattern/replacement/'

```
sed 's/ERROR/CRITICAL/' log.txt
```

```
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$ sed 's/ERROR/CRITICAL/' log.txt
Timestamp          Category      Message
1598843202        INFO         Booting up system
1598843402        INFO         Booting up critical service: Authorization
1598863888        CRITICAL    User anonymous attempt to access protected resource without credentials
1598863901        CRITICAL    Requested resource not found
1598864411        INFO         User admin logged out
nthapa0000@LAPTOP-GCLF70VR:/mnt/c/Users/NTC/Desktop/dev/week2/BashDemo
$
```

Now we want to create a backup of log.txt file since we are making lot of changes

```
sed -ibackup 's/ERROR/CRITICAL/' log.txt
```

6. awk

Command

```
3   The sed's substitute command has the following structure:
4   's/pattern/replacement/'

5
6   sed 's/ERROR/CRITICAL/' log.txt
7   sed -ibackup 's/ERROR/CRITICAL/' log.txt
8   sed '3 s/ERROR/CRITICAL/' log.txt
9   sed '3,5 s/ERROR/CRITICAL/' log.txt
10  sed -n '3,/ERROR/ p' log.txt
11
12  awk
13  awk [options] script file
14
15  How patterns are define: '(pattern){action}'
16  awk '/ERROR/{print $0}' log.txt
17  awk '{gsub(/ERROR/, "CRITICAL")}{print}' log.txt
18  awk 'BEGIN {print "LOG SUMMARY\n-----"} {print} END {print "\nEND OF LOG SUMMARY"}' log.txt
19  awk '{print $1, $2}' log.txt
20  awk -F "," '{print $1, $2}' log.txt
21  awk '{count[$2]++} END {print count["ERROR"]}' log.txt
22  awk '{ if ($1 > 1598863888 ) {print $0} }' log.txt
```

