Activity 2_B Files & Directories

Steps

Login to your machine.

Perform below steps.

1. **mkdir** - It will create a directory/folder which can contain files and sub-directories.

   Let's use ls command to see if created or not.

   $ mkdir work

   ```
   [nishant@ip-172-16-1-243 ~]$ mkdir work
   [nishant@ip-172-16-1-243 ~]$ ls
   data.txt  work
   ```

   $ cd work

   ```
   [nishant@ip-172-16-1-243 ~]$ cd work/
   [nishant@ip-172-16-1-243 work]$ pwd
   /home/nishant/work
   ```

   How to create directories and subdirectories in it. For that we will use **-p option**

   $ mkdir -p dir1/subdir1/subdir1a
   $ ls
   $ cd dir1
   $ ls
   $ cd subdir1/
   $ ls

```
[nishant@ip-172-16-1-243 work]$ mkdir -p dir1/subdir1/
[nishant@ip-172-16-1-243 work]$ ls
dir1  file1
[nishant@ip-172-16-1-243 work]$ cd dir1/
[nishant@ip-172-16-1-243 dir1]$ ls
subdir1
[nishant@ip-172-16-1-243 dir1]$ cd subdir1/
[nishant@ip-172-16-1-243 subdir1]$ ls
subdir1a
[nishant@ip-172-16-1-243 subdir1]$
```

2. **touch -** Use to create empty files.
   Let's create a file, check with ls & ls -l, then put some content in it with echo command and then check the content with cat command.

   $ touch file1
   $ ls -l
   $ echo "Hello" > file1
   $ cat file1

```
[nishant@ip-172-16-1-243 work]$ touch file1
[nishant@ip-172-16-1-243 work]$ ls
file1
[nishant@ip-172-16-1-243 work]$ ls -l
total 0
-rw-r--r--. 1 nishant nishant 0 Apr  2 06:17 file1
[nishant@ip-172-16-1-243 work]$ echo "Hello" > file1
[nishant@ip-172-16-1-243 work]$ cat file1
Hello
```

   Let's create multiple files using touch command in 1 go. It will create 4 empty files, name as file1.txt file2.txt and so on.

   $ touch file{1..4}.txt
   $ ls -l

```
[nishant@ip-172-16-1-243 work]$ touch file{1..4}.txt
[nishant@ip-172-16-1-243 work]$ ls -l
total 4
-rw-r--r--. 1 nishant nishant 6 Apr  2 06:17 file1
-rw-r--r--. 1 nishant nishant 0 Apr  2 07:12 file1.tx
-rw-r--r--. 1 nishant nishant 0 Apr  2 07:12 file2.tx
-rw-r--r--. 1 nishant nishant 0 Apr  2 07:12 file3.tx
-rw-r--r--. 1 nishant nishant 0 Apr  2 07:12 file4.tx
[nishant@ip-172-16-1-243 work]$
```

3. **Now Let's see how to delete files.**

$ rm file1.txt

```
[nishant@ip-172-16-1-243 work]$ rm file1.txt
[nishant@ip-172-16-1-243 work]$ ls
file1  file2.txt  file3.txt  file4.txt
```

what if you want to delete all the created files in command.

$ rm file*.txt

```
[nishant@ip-172-16-1-243 work]$ ls
file1  file2.txt  file3.txt  file4.txt
[nishant@ip-172-16-1-243 work]$ rm file*.txt
[nishant@ip-172-16-1-243 work]$ ls
file1
[nishant@ip-172-16-1-243 work]$
```

4. **Let's see how to delete directories**. We can use both **rm and rmdir** commands to delete a directory, where rmdir will only remove empty directories.
   Switch to **your** work directory. Replace your home directory with **nishant**
   $ cd /home/nishant/work
   $ mkdir empty_dir

   Now try to delete directory with both the commands.

   $ rm empty_dir
   It will give an error.

```
[nishant@ip-172-16-1-243 work]$ ls
dir1  empty_dir  file1
[nishant@ip-172-16-1-243 work]$ rm empty_dir/
rm: cannot remove 'empty_dir/': Is a directory
[nishant@ip-172-16-1-243 work]$
```

Now try to delete it with rmdir, it will be deleted.

$ rmdir empty_dir

```
[nishant@ip-172-16-1-243 work]$ rmdir empty_dir/
[nishant@ip-172-16-1-243 work]$ ls
dir1  file1
[nishant@ip-172-16-1-243 work]$
```

If you remember when we created dir1 we also create subdirectories inside it.

```
[nishant@ip-172-16-1-243 work]$ ls dir1/
subdir1
[nishant@ip-172-16-1-243 work]$
```

As we know, rmdir can only delete empty directories. So we will use rm command with **-r option,** here **-r** denotes recursively.

$ rmdir dir1
$ rm dir1/
$ rm -r dir1/
$ ls

```
[nishant@ip-172-16-1-243 work]$ rmdir dir1/
rmdir: failed to remove 'dir1/': Directory not empt
[nishant@ip-172-16-1-243 work]$ rm dir1/
rm: cannot remove 'dir1/': Is a directory
[nishant@ip-172-16-1-243 work]$ rm -r dir1/
[nishant@ip-172-16-1-243 work]$ ls
file1
[nishant@ip-172-16-1-243 work]$
```

5. **Redirection operators echo with > and >>**
   echo “Hi this is you lab”
   echo “This is your Lab” > file1
   echo “this is your lab, it also include previous line” >> file1

6. **Symbolic link --- >** These links are equivalent to shortcuts in windows.

   Create a directory in /tmp and cd into it.
   # **mkdir /tmp/work && cd /tmp/work**

   Create directory from a up to i with -p (parent option) and cd to directory “e”
   # **mkdir -p a/b/c/d/e/f/g/h/i**
   # **cd a/b/c/d/e**

```
nishant@DDC4-L-13447ZH:/tmp$ mkdir work
nishant@DDC4-L-13447ZH:/tmp$ cd work/
nishant@DDC4-L-13447ZH:/tmp/work$ ls
nishant@DDC4-L-13447ZH:/tmp/work$ ls -l
total 0
nishant@DDC4-L-13447ZH:/tmp/work$ mkdir -p a/b/c/d/e/f/g/h/i
nishant@DDC4-L-13447ZH:/tmp/work$ cd a/b/c/d/e/
```

   Create a empty file and enter a text in to this using echo command.
   # **touch avc.txt**
   # **echo "hello abcfile inside e directory" > abc.txt**

   Now check its content.
   # **cat abc.txt**

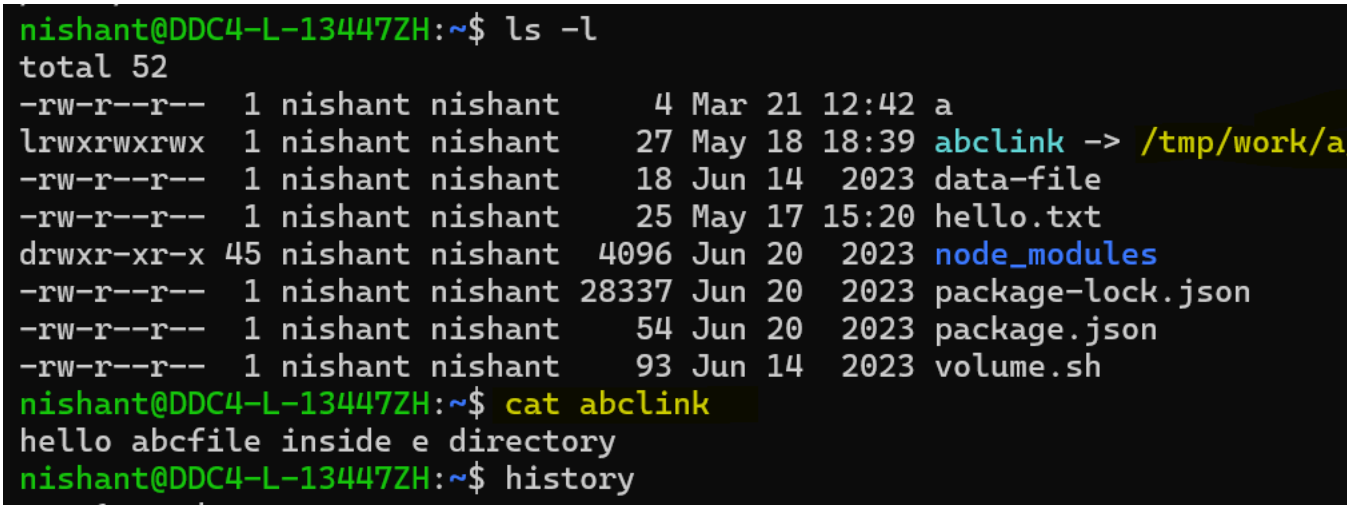
```
nishant@DDC4-L-13447ZH:/tmp/work/a/b/c/d/e$ touch abc.txt
nishant@DDC4-L-13447ZH:/tmp/work/a/b/c/d/e$ echo "hello abcfile inside
nishant@DDC4-L-13447ZH:/tmp/work/a/b/c/d/e$ cat abc.txt
hello abcfile inside e directory
```

   Now switch to the directory where you want the shortcut link. I want it at home directory,
   # **cd**

   Now run below command to create link. - Syntax ln -sf path_to_file linkname
   # **ln -sf /tmp/work/a/b/c/d/e/abc.txt abclink**

   Check if it’s created or not.

ls -l
cat abclink



Commands to go through.

Command	Uses
whoami	To show logged in user name
pwd	To show Present working Directory
touch filename	To Create empty Files
ls	To List the created files & directories
ls -l	To List the created files & directories with more information - To identify the files & directories Ex - time, permision
ls -la	To show hidden files as well - a =All
echo	To print a message/content on terminal/redirect to file > Overwrite - delete old - add new >> Apend
cat	To Display the content of the file.

Editor - vi Filename	Editor to write files. 1. Command mode 2. Press i --> Go to insert mode. (Allows you to write/delete/modify) 3. Esc -- > To come out of Insert mode & be in command mode. 4. :w - To save the file 5 :q - To quite the vi editor 6 :wq - To save and exit				
cd	To swtich between directories. cd Home directory cd - (Last folder) cd ../ (1 back) cd ../../ (2back)				
mkdir direc_name	To create directories				
rmdir direc_name	To delete empty directory				
rm -r direc_name	To Delete directory and it's inside content				
rm -r *	Everyting	data.txt	hello.txt	java.txt	
rm -r *.txt		data.txt	hello.txt	java.txt	
rm -r *.py		data.txt	hello.txt	java.txt	