

Assignment

links-

(Soft-link)

1. MODIFY any SOFT link and observe original?

```
aneed@LAPTOP-ESM7NGU3:~/aneed$ touch a
aneed@LAPTOP-ESM7NGU3:~/aneed$ cat > a
soft-link
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s a b
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s b c
aneed@LAPTOP-ESM7NGU3:~/aneed$ cat c
soft-link
aneed@LAPTOP-ESM7NGU3:~/aneed$ tree
.
├── a
├── b -> a
└── c -> b

0 directories, 3 files
aneed@LAPTOP-ESM7NGU3:~/aneed$
```

2. MODIFY original file of SOFT link and observe?

```
aneed@LAPTOP-ESM7NGU3:~/aneed$ ls
aneed@LAPTOP-ESM7NGU3:~/aneed$ touch ab
aneed@LAPTOP-ESM7NGU3:~/aneed$ cat > ab
MODIFY original file of SOFT link and observe
aneed@LAPTOP-ESM7NGU3:~/aneed$
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s ab bc
aneed@LAPTOP-ESM7NGU3:~/aneed$ ls
ab  bc
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s ab cd
aneed@LAPTOP-ESM7NGU3:~/aneed$ cat cd
MODIFY original file of SOFT link and observe
aneed@LAPTOP-ESM7NGU3:~/aneed$ tree
.
├── ab
├── bc -> ab
└── cd -> ab

0 directories, 3 files
aneed@LAPTOP-ESM7NGU3:~/aneed$ ls -l
total 4
-rw-r--r-- 1 aneed aneed 48 Apr  2 23:34 ab
lrwxrwxrwx 1 aneed aneed  2 Apr  2 23:34 bc -> ab
lrwxrwxrwx 1 aneed aneed  2 Apr  2 23:35 cd -> ab
aneed@LAPTOP-ESM7NGU3:~/aneed$
```

3. Remove any SOFT link and observe original?

```
aneed@LAPTOP-ESM7NGU3:~/aneed$ ls
ab
aneed@LAPTOP-ESM7NGU3:~/aneed$ cat > ab
1. Remove any SOFT link and observe original
aneed@LAPTOP-ESM7NGU3:~/aneed$ cat ab
1. Remove any SOFT link and observe original
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s ab ba
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s ba cd
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s ba dc
aneed@LAPTOP-ESM7NGU3:~/aneed$ tree
├── ab
├── ba -> ab
├── cd -> ba
└── dc -> ba

0 directories, 4 files
aneed@LAPTOP-ESM7NGU3:~/aneed$ rm ba
aneed@LAPTOP-ESM7NGU3:~/aneed$ tree
├── ab
├── cd -> ba
└── dc -> ba

0 directories, 3 files
aneed@LAPTOP-ESM7NGU3:~/aneed$
```

4. Remove original file of SOFT link and observe?

```
aneed@LAPTOP-ESM7NGU3:~/aneed$ touch link1
aneed@LAPTOP-ESM7NGU3:~/aneed$ cat link1
aneed@LAPTOP-ESM7NGU3:~/aneed$ cat > link1
2. Remove original file of SOFT link and observe
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s link1 link1.1
aneed@LAPTOP-ESM7NGU3:~/aneed$ ln -s link1.1 link1.2
aneed@LAPTOP-ESM7NGU3:~/aneed$ tree
├── link1
├── link1.1 -> link1
└── link1.2 -> link1.1

0 directories, 3 files
aneed@LAPTOP-ESM7NGU3:~/aneed$ rm link1
aneed@LAPTOP-ESM7NGU3:~/aneed$ tree
├── link1.1 -> link1
└── link1.2 -> link1.1

0 directories, 2 files
aneed@LAPTOP-ESM7NGU3:~/aneed$
```

(Hard-link)

1. MODIFY any HARD link and observer original?

```
aneed@LAPTOP-ESM7NGU3:~$ mkdir khan
aneed@LAPTOP-ESM7NGU3:~$ cd khan
aneed@LAPTOP-ESM7NGU3:~/khan$ touch ab
aneed@LAPTOP-ESM7NGU3:~/khan$ cat > ab
1. MODIFY any HARD link and observer original?
aneed@LAPTOP-ESM7NGU3:~/khan$
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab bc
aneed@LAPTOP-ESM7NGU3:~/khan$ ln bc cd
aneed@LAPTOP-ESM7NGU3:~/khan$ ln cd ef
aneed@LAPTOP-ESM7NGU3:~/khan$ cat ef
1. MODIFY any HARD link and observer original?
aneed@LAPTOP-ESM7NGU3:~/khan$ tree

├── ab
├── bc
├── cd
└── ef

0 directories, 4 files
aneed@LAPTOP-ESM7NGU3:~/khan$
```

2. MODIFY Original file of HARD link and observe?

```
aneed@LAPTOP-ESM7NGU3:~/khan$ cd
aneed@LAPTOP-ESM7NGU3:~$ rm -rf khan
aneed@LAPTOP-ESM7NGU3:~$ mkdir khan
aneed@LAPTOP-ESM7NGU3:~$ cd khan
aneed@LAPTOP-ESM7NGU3:~/khan$ touch ab
aneed@LAPTOP-ESM7NGU3:~/khan$ cat > ab
2. MODIFY Original file of HARD link and observe?
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab bc
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab cd
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab ef
aneed@LAPTOP-ESM7NGU3:~/khan$ tree

├── ab
├── bc
├── cd
└── ef

0 directories, 4 files
aneed@LAPTOP-ESM7NGU3:~/khan$
```

3. Remove any HARD link and observe original?

```
aneed@LAPTOP-ESM7NGU3:~/khan$ cd
aneed@LAPTOP-ESM7NGU3:~$ rm -rf khan
aneed@LAPTOP-ESM7NGU3:~$ mkdir khan
aneed@LAPTOP-ESM7NGU3:~$ cd khan
aneed@LAPTOP-ESM7NGU3:~/khan$ touch ab
aneed@LAPTOP-ESM7NGU3:~/khan$ cat > ab
2. MODIFY Original file of HARD link and observe?
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab bc
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab cd
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab ef
aneed@LAPTOP-ESM7NGU3:~/khan$ tree
.
├── ab
├── bc
├── cd
└── ef

0 directories, 4 files
aneed@LAPTOP-ESM7NGU3:~/khan$ rm ef
aneed@LAPTOP-ESM7NGU3:~/khan$ tree
.
├── ab
├── bc
└── cd

0 directories, 3 files
aneed@LAPTOP-ESM7NGU3:~/khan$
```

4. Remove original file of HARD link and observe?

```
aneed@LAPTOP-ESM7NGU3:~/khan$ cd
aneed@LAPTOP-ESM7NGU3:~$ rm -rf khan
aneed@LAPTOP-ESM7NGU3:~$ mkdir khan
aneed@LAPTOP-ESM7NGU3:~$ cd khan
aneed@LAPTOP-ESM7NGU3:~/khan$ touch ab
aneed@LAPTOP-ESM7NGU3:~/khan$ cat > ab
2. MODIFY Original file of HARD link and observe?
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab bc
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab cd
aneed@LAPTOP-ESM7NGU3:~/khan$ ln ab ef
aneed@LAPTOP-ESM7NGU3:~/khan$ tree
.
├── ab
├── bc
├── cd
└── ef

0 directories, 4 files
aneed@LAPTOP-ESM7NGU3:~/khan$ rm ef
aneed@LAPTOP-ESM7NGU3:~/khan$ tree
.
├── ab
├── bc
└── cd

0 directories, 3 files
aneed@LAPTOP-ESM7NGU3:~/khan$ rm ab
aneed@LAPTOP-ESM7NGU3:~/khan$ tree
.
├── bc
└── cd

0 directories, 2 files
```

9. Comparison among soft,hard,copy?

Hard Link :

A hard link acts as a copy (mirrored) of the selected file. It accesses the data available in the original file. If the earlier selected file is deleted, the hard link to the file will still contain the data of that file.

Soft Link :

A soft link (also known as Symbolic link) acts as a pointer or a reference to the file name. It does not access the data available in the original file. If the earlier file is deleted, the soft link will be pointing to a file that does not exist anymore.

Copy :

cp command for copying files from one location to another. This command can also copy directories (folders). The syntax of this command is: cp [... file/directory-sources] [destination] [file/directory-sources] specifies the sources of the files or directories you want to copy.

10. what are wget , curl commands differences?

Curl and wget both have unique features that make them suitable for different tasks.

Curl, for example, is often used to interact with APIs and can handle a variety of data formats. It also supports authentication, cookies, and more. Here are some of key features of curl –

Supports a wide range of protocols, including HTTP, HTTPS, FTP, FTPS, SCP, SFTP, and more.

Wget, on other hand, is primarily designed for downloading files and has a number of features that make it a reliable tool for this task. Here are some of key features of wget –

Supports HTTP and FTP protocols.

Can download recursively to download all linked files.

Scripts-

1. write a script to print the date and redirect it to output.txt?

```
GNU nano 6.2 one.sh *
echo $(date) > output.txt
```

```
aneed@LAPTOP-ESM7NGU3: ~$ nano one.sh
aneed@LAPTOP-ESM7NGU3: ~$ chmod 777 one.sh
aneed@LAPTOP-ESM7NGU3: ~$ ./one.sh
aneed@LAPTOP-ESM7NGU3: ~$ cat output.txt
Mon Apr 3 10:56:15 IST 2023
aneed@LAPTOP-ESM7NGU3: ~$
```

2. create a file or folder using the date as the name?

```
GNU nano 6.2
mkdir $(date +"%y-%m-%d")
```

```
aneed@LAPTOP-ESM7NGU3: ~/aneed2$ nano 2.sh
aneed@LAPTOP-ESM7NGU3: ~/aneed2$ chmod 777 2.sh
aneed@LAPTOP-ESM7NGU3: ~/aneed2$ ./2.sh
aneed@LAPTOP-ESM7NGU3: ~/aneed2$ ls
2.sh  23-04-03
aneed@LAPTOP-ESM7NGU3: ~/aneed2$
```

3. i) Create a bash script to print the local time, date, username of your system, and your current path.

```
aneed@LAPTOP-ESM7NGU3:~/aneed3$ nano 3.sh
aneed@LAPTOP-ESM7NGU3:~/aneed3$ chmod 777 3.sh
aneed@LAPTOP-ESM7NGU3:~/aneed3$ ./3.sh
01:04:12
04/03/23
aneed
/home/aneed/aneed3
aneed@LAPTOP-ESM7NGU3:~/aneed3$
```

```
GNU nano 6.2 3.sh *
echo $(date +%X") && echo $(date +%D") && whoami && pwd
```

ii) After printing, redirect the output into a file called output.txt

```
GNU nano 6.2                                     three.sh *
(echo $(date +%X") && echo $(date +%D") && whoami && pwd ) > output.txt
```

```
aneed@LAPTOP-ESM7NGU3:~$ nano three.sh
aneed@LAPTOP-ESM7NGU3:~$ chmod 777 three.sh
aneed@LAPTOP-ESM7NGU3:~$ ./three.sh
aneed@LAPTOP-ESM7NGU3:~$ cat output.txt
11:16:41
04/03/23
aneed
/home/aneed
aneed@LAPTOP-ESM7NGU3:~$
```


iii) Insert output.txt into a new directory, where the directory name is the current timestamp.

```
aneed@LAPTOP-ESM7NGU3:~/aneed$ cd
aneed@LAPTOP-ESM7NGU3:~$ mkdir $(date +"%X") && mv output.txt $(date +"%X")
aneed@LAPTOP-ESM7NGU3:~$ ls
1          Apr          ahmed     aneed5    file1      one.sh
10:36:53   IST          aneed     b         first.sh   ouput.txt
11:56:22   Sat          aneed2    c         first.sh.save out.txt
3.sh       'Sat Apr 1 10:33:40 IST 2023' aneed3    dat.sh    khan       test1
9.txt.save a          aneed4    file      my.txt     three.sh
aneed@LAPTOP-ESM7NGU3:~$ ls
1          Apr          ahmed     aneed5    file1      one.sh
10:36:53   IST          aneed     b         first.sh   ouput.txt
11:56:22   Sat          aneed2    c         first.sh.save out.txt
3.sh       'Sat Apr 1 10:33:40 IST 2023' aneed3    dat.sh    khan       test1
9.txt.save a          aneed4    file      my.txt     three.sh
aneed@LAPTOP-ESM7NGU3:~$
```

4. Create a bash script to execute the date every 2 minutes once on Saturdays only.

6. Print the count of the number of files in a file called count.txt?

```
Windows PowerShell  X  aneed@LAPTOP-ESM7NGU3:  X  +  v
aneed@LAPTOP-ESM7NGU3:~$ ls |wc -l > count.txt
aneed@LAPTOP-ESM7NGU3:~$ cat count.txt
37
aneed@LAPTOP-ESM7NGU3:~$
```

7. Create files dynamically every day at 12 AM where the file name is a date?

```
GNU nano 6.2 /tmp/crontab.HnVzvj/crontab
0 0 * * * /home/aneed/$(date +%d/%m/%y).txt

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
```

CRONTAB SCRIPTS LABS:

1. Write a script to print the current directory and username and redirect it to a file called output.txt?

```
Windows PowerShell  x  aneed@LAPTOP-ESM7NGU3:  x  +  v
aneed@LAPTOP-ESM7NGU3:~$ nano aneed.sh
aneed@LAPTOP-ESM7NGU3:~$ chmod 777 aneed.sh
aneed@LAPTOP-ESM7NGU3:~$ ./aneed.sh
aneed@LAPTOP-ESM7NGU3:~$ cat output.txt
Sat Apr 15 23:34:18 IST 2023
aneed
/home/aneed
aneed@LAPTOP-ESM7NGU3:~$
```

```
GNU nano 6.2  aneed.sh *
(echo "$(date)"
echo "$(whoami)"
echo "$(pwd)") >> output.txt
```

2. Create a file with the current timestamp as its name inside a folder with the current date as its name?

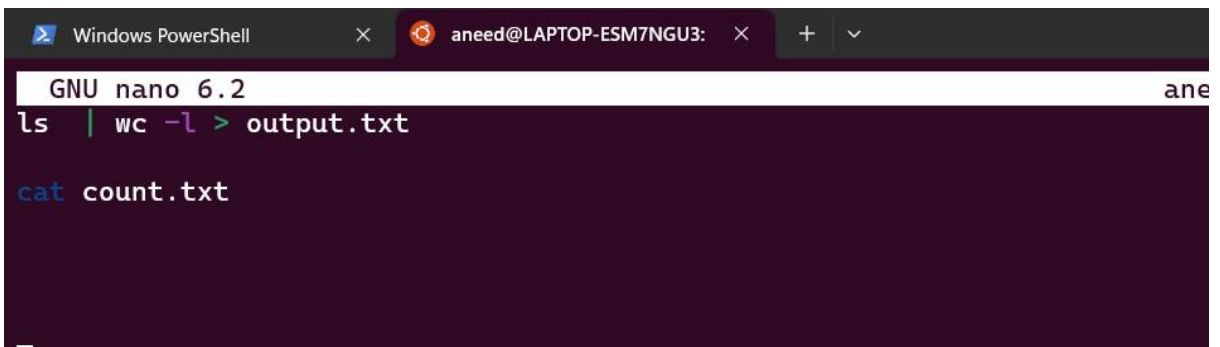
```
Windows PowerShell  x  aneed@LAPTOP-ESM7NGU3:  x  +  v
aneed@LAPTOP-ESM7NGU3:~$ mkdir $(date +%d/%m/%y) && cd $(date +%d/%m/%y) && touch $(date).txt
```

3. Create a bash script to print the local time, date, username of your system, and your current path and redirect the output into a file where the directory name is the current timestamp.

```
GNU nano 6.2
(echo "$(date)"
echo "$(whoami)"
echo "$(pwd)") >> output.txt

mkdir $(date)
mv output.txt $(date)
```

4. Write a script to print the count of the number of files in a folder and redirect the count to a file called count.txt.



The screenshot shows a terminal window with two tabs: 'Windows PowerShell' and 'aneed@LAPTOP-ESM7NGU3:'. The active tab is the terminal. The prompt is 'GNU nano 6.2'. The user has entered the command `ls | wc -l > output.txt`. Below this, the user has entered `cat count.txt`.



The screenshot shows a terminal window with two tabs: 'Windows PowerShell' and 'aneed@LAPTOP-ESM7NGU3:'. The active tab is the terminal. The user has entered the following commands: `nano aneed.sh`, `chmod 777 aneed.sh`, and `./aneed.sh`. The output of the script is `37`. The prompt is now `aneed@LAPTOP-ESM7NGU3:~$`.

(Reference: Google about word count and pipe commands in Linux)

5. Create a bash script to execute the date every 2 minutes once on weekends only?

```
Windows PowerShell  aneed@LAPTOP-ESM7NGU3:
GNU nano 6.2
date
```

```
Windows PowerShell  aneed@LAPTOP-ESM7NGU3:
GNU nano 6.2 /tmp/crontab.XiQQe8/cr
*/2 * * * 5-6 /home/aneed/aneed.sh
```

6. Take a backup of a folder daily twice?

```
Windows PowerShell  aneed@LAPTOP-ESM7NGU3:
GNU nano 6.2 /tmp/crontab.XiQQe8/crontab *
0 0,12 * * * cp -r /home/aneed/aneed.sh/backup
```