

Links-

```
muqtadar@QUICKSILVER:~$ touch b.txt
muqtadar@QUICKSILVER:~$ cat > b.txt
original text
^C
muqtadar@QUICKSILVER:~$ ln -s b.txt b1.txt
muqtadar@QUICKSILVER:~$ cat b1.txt
original text
muqtadar@QUICKSILVER:~$ cat >> b1.txt
soft link text
^C
muqtadar@QUICKSILVER:~$ cat b.txt
original text
soft link text
muqtadar@QUICKSILVER:~$ cat >> b.txt
original file edit
^C
muqtadar@QUICKSILVER:~$ cat b1.txt
original text
soft link text
original file edit
```

1. MODIFY any SOFT link and observer original?

⇒ **The original file also modifies**

2. MODIFY original file of SOFT link and observe?

⇒ **All the soft links are modified when original is modified**

```

muqtadar@QUICKSILVER:~$ ln -s b1.txt b2.txt
muqtadar@QUICKSILVER:~$ cat b2.txt
original text
soft link text
original file edit
muqtadar@QUICKSILVER:~$ rm b2.txt
muqtadar@QUICKSILVER:~$ cat b1.txt
original text
soft link text
original file edit
muqtadar@QUICKSILVER:~$ rm b.txt
muqtadar@QUICKSILVER:~$ cat b1.txt
cat: b1.txt: No such file or directory
muqtadar@QUICKSILVER:~$

```

1. Remove any SOFT link and observe original?

⇒ Original stays available and unchanged

2. Remove original file of SOFT link and observe?

⇒ All the soft links of the original files are removed when the original is deleted

```

muqtadar@QUICKSILVER:~$ touch n1.txt
muqtadar@QUICKSILVER:~$ ln n1.txt n11.txt
muqtadar@QUICKSILVER:~$ cat > n1.txt
original edit
^C
muqtadar@QUICKSILVER:~$ cat n11.txt
original edit
muqtadar@QUICKSILVER:~$ cat >> n11.txt
hard link edit
^C
muqtadar@QUICKSILVER:~$ cat n1.txt
original edit
hard link edit
muqtadar@QUICKSILVER:~$

```

1. MODIFY any HARD link and observe original?

⇒ The main file is also modified

2. MODIFY Original file of HARD link and observe?

⇒ all the hard links of the original file are modified

```
muqtadar@QUICKSILVER:~$ ln n11.txt n12.txt
muqtadar@QUICKSILVER:~$ cat n12.txt
original edit
hard link edit
```

```
muqtadar@QUICKSILVER:~$ rm n12.txt
muqtadar@QUICKSILVER:~$ cat n11.txt
original edit
hard link edit
muqtadar@QUICKSILVER:~$ rm n1.txt
muqtadar@QUICKSILVER:~$ cat n11.txt
original edit
hard link edit
```

3. Remove any HARD link and observe original?

⇒ removal of hard link doesn't effect the original

4. Remove original file of HARD link and observe?

⇒ unlike soft link, removal of the original file does not remove the hard link.

—

Comparison among soft,hard,copy?

Soft link	Hard link	copy - cp
Like to the path of the file not the data in the file	Direct link to the data of the source file	Copies the content of the source to the destination file
Changes to one soft link will reflect in the source file and vice versa	Changes to one hard link will reflect in all the hard-linked files.	
Removal of the source file removes all the soft links	Removal of any hard-linked file does not affect other hard-linked files	
Syntax: Ln -s "source" "link"	Syntax: Ln "source" "link"	Syntax: cp "source" "copy"

what are wget , curl commands differences?

wget	curl
Primarily used for downloading files from the web using transfer protocols	Tool to transfer data from or to a server
Downloads files into the current working directory by default	It can download the file from a specific URL and save it at the same name as on the remote server
Used to download entire websites or parts of the website to view offline	Can be used in scripts to automate file transfer

Scripts-

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

```
(echo " Date : $(date +%d-%m-%y ) ") >> output.txt
```

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

```
muqtadar@QUICKSILVER:~/assignments$ mkdir $(date +%d-%m-%y)
muqtadar@QUICKSILVER:~/assignments$ ls
28-03-23
```

3.

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

```
echo "Date: $(date +%d%m%y)"
echo "Time:$(date +%T)"
echo "User Name:$(whoami)"
echo "Current path:$(pwd)"
```

```
Date: 28-03-23
Time:23:03:43
User Name:muqtadar
Current path:/home/muqtadar/assignments
```

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

```
muqtadar@QUICKSILVER:~/assignments$ cat output.txt
Date: 28-03-23
Time:23:04:39
User Name:muqtadar
Current path:/home/muqtadar/assignments
```

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

cat

```
muqtadar@QUICKSILVER:~/assignments$ mkdir $(date +%d-%m-%y,%H:%M:%S) && mv output.txt $(date +%d-%m-%y,%H:%M:%S)^
muqtadar@QUICKSILVER:~/assignments$ cd 29-03-23,14:52:18
muqtadar@QUICKSILVER:~/assignments/29-03-23,14:52:18$ ls
output.txt
```

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

```
GNU nano 6.2 d2.sh
(echo "$(date)") >> output.txt
```

```
GNU nano 6.2 /tmp/crontab.AU3GPZ/crontab
*/2 * * * 6 /home/muqtadar/assignments/d2.sh
```

5. Take a backup of a folder every month twice?

```
GNU nano 6.2 /tmp/crontab.ZQN8xE/crontab
0 0 1,15 * * cp -r /home/muqtadar/assignments/main /home/muqtadar/assignments/backup
# Edit this file to suit your needs
```

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

```
muqtadar@QUICKSILVER:~$ ls |wc -l > count.txt
muqtadar@QUICKSILVER:~$ cat count.txt
72
```

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

```
GNU nano 6.2 /tmp/crontab.glsbCl/crontab *
0 0 * * * touch /home/muqtadar/assignments/$(date +%d/%m/%y).txt
```

CRONTAB SCRIPTS LABS:

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

```
GNU nano 6.2 b4.
(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?

```
muqtadar@QUICKSILVER:~$ 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 called output.txt. Insert output.txt into a new directory, 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.
(Reference: Google about word count and pipe commands in Linux)

```
ls | wc -l > count.txt
cat count.txt
```

```
muqtadar@QUICKSILVER:~$ nano b6.sh
muqtadar@QUICKSILVER:~$ bash b6.sh
80
```

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

```
GNU nano 6.2                                     date.sh
date
```

```
GNU nano 6.2                                     /tmp/clonead72nqxqv/clone
*/2 * * * 6,7 /home/muqtadar/assignments/date.sh
```


6. Take a backup of a folder daily twice?

```
GNU nano 6.2 /tmp/crontab.YDQc6B/crontab
0 0,12 * * * cp -r /home/muqtadar/main /home/muqtadar/backup
# Edit this file to introduce tasks to be run by cron
```

Bonus question:

Create a file with the current timestamp as its name once per minute inside a folder with the current timestamp as its name once per 4 minutes