

ASSIGNMENT 2:

LINKS:

- SOFT LINKS:

1. MODIFY any SOFT link and observe original?

```
tasneem@SyedAllaBaksh: ~/test
tasneem@SyedAllaBaksh:~$ mkdir test && cd test
tasneem@SyedAllaBaksh:~/test$ touch original.txt
tasneem@SyedAllaBaksh:~/test$ ln -s original.txt s11.txt
tasneem@SyedAllaBaksh:~/test$ ls
original.txt  s11.txt
tasneem@SyedAllaBaksh:~/test$ cat > s11.txt
im writing in s11.txt
tasneem@SyedAllaBaksh:~/test$ cat s11.txt
im writing in s11.txt
tasneem@SyedAllaBaksh:~/test$ cat original.txt
im writing in s11.txt
tasneem@SyedAllaBaksh:~/test$ cat >> s11.txt
im appending
tasneem@SyedAllaBaksh:~/test$ cat s11.txt
im writing in s11.txt
im appending
tasneem@SyedAllaBaksh:~/test$ cat original.txt
im writing in s11.txt
im appending
tasneem@SyedAllaBaksh:~/test$
```

From what I observe is, if I try to write or append some content in soft link file, it'll show that content in original file as well.

2. MODIFY original file of SOFT link and observe?

```
tasneem@SyedAllaBaksh: ~/test
tasneem@SyedAllaBaksh:~/test$ cat > original.txt
im writing in original .txt

tasneem@SyedAllaBaksh:~/test$ cat original.txt
im writing in original .txt

tasneem@SyedAllaBaksh:~/test$ cat >> original.txt
im appending from org.txt
tasneem@SyedAllaBaksh:~/test$ cat original.txt
im writing in original .txt

im appending from org.txt
tasneem@SyedAllaBaksh:~/test$ cat s11.txt
im writing in original .txt

im appending from org.txt
tasneem@SyedAllaBaksh:~/test$
```

My observation is when I write some content in original file or append, it'll show that same content in it's soft link file as well.

1. Remove any SOFT link and observe original?

```

tasneem@SyedAllaBaksh: ~/test
tasneem@SyedAllaBaksh:~/test$ ls
original.txt  sl1.txt
tasneem@SyedAllaBaksh:~/test$ cat original.txt
im writing in original .txt

im appending from org.txt
tasneem@SyedAllaBaksh:~/test$ cat sl1.txt
im writing in original .txt

im appending from org.txt
tasneem@SyedAllaBaksh:~/test$ rm sl1.txt
tasneem@SyedAllaBaksh:~/test$ ls
original.txt
tasneem@SyedAllaBaksh:~/test$ cat original.txt
im writing in original .txt

im appending from org.txt
tasneem@SyedAllaBaksh:~/test$

```

If I remove the soft link file, it doesn't effect on its original file content. (Its like breaking a chain).

2. Remove original file of SOFT link and observe?

```

tasneem@SyedAllaBaksh: ~/test
tasneem@SyedAllaBaksh:~/test$ ls
original.txt  sl1.txt
tasneem@SyedAllaBaksh:~/test$ cat original.txt
im writing in original .txt

im appending from org.txt
tasneem@SyedAllaBaksh:~/test$ cat sl1.txt
im writing in original .txt

im appending from org.txt
tasneem@SyedAllaBaksh:~/test$ rm original.txt
tasneem@SyedAllaBaksh:~/test$ ls
sl1.txt
tasneem@SyedAllaBaksh:~/test$ cat sl1.txt
cat: sl1.txt: No such file or directory
tasneem@SyedAllaBaksh:~/test$

```

Here, if we remove the original file, then its soft link file also be removed. Even if we try to see the content in soft link file, it'll throw an error as "NO SUCH FILE OR DIRECTORY".

So, basically what I understood from the above scenarios is that, soft link act as a reference of the original file. That's why, when we try to modify the original files, it'll show effect on its soft link file as well.

- **HARD LINKS:**

1. MODIFY any HARD link and observer original?

```
tasneem@SyedAllaBaksh: ~/hlink
tasneem@SyedAllaBaksh:~$ cd hlink
tasneem@SyedAllaBaksh:~/hlink$ touch 1.txt
tasneem@SyedAllaBaksh:~/hlink$ ln 1.txt h11.txt
tasneem@SyedAllaBaksh:~/hlink$
tasneem@SyedAllaBaksh:~/hlink$ ls
1.txt  h11.txt
tasneem@SyedAllaBaksh:~/hlink$ cat > h11.txt
im from h11.txt
tasneem@SyedAllaBaksh:~/hlink$ cat 1.txt
im from h11.txt
tasneem@SyedAllaBaksh:~/hlink$ cat h11.txt
im from h11.txt
tasneem@SyedAllaBaksh:~/hlink$
```

Here, whatever content we add in hard link file, we can see that same content in original file as well.

2. MODIFY Original file of HARD link and observe?

```
tasneem@SyedAllaBaksh: ~/hlink
tasneem@SyedAllaBaksh:~/hlink$ ls
1.txt  h11.txt
tasneem@SyedAllaBaksh:~/hlink$ cat > 1.txt
im from 1.txt
tasneem@SyedAllaBaksh:~/hlink$ cat >> 1.txt
im appended one
tasneem@SyedAllaBaksh:~/hlink$ cat 1.txt
im from 1.txt
im appended one
tasneem@SyedAllaBaksh:~/hlink$ cat h11.txt
im from 1.txt
im appended one
tasneem@SyedAllaBaksh:~/hlink$
```

When I append or write content in original file, the contents of hard link file also change accordingly.

3. Remove any HARD link and observer original?

When I remove the hard link file and check the original file, the contents that were written from hard link file won't delete, it'll save like that only.

```

tasneem@SyedAllaBaksh: ~/hlink
tasneem@SyedAllaBaksh:~/hlink$ ls
1.txt  h11.txt
tasneem@SyedAllaBaksh:~/hlink$ cat >> h11.txt
im from h11.txt
tasneem@SyedAllaBaksh:~/hlink$ cat 1.txt
im from 1.txt
im appended one
im from h11.txt
tasneem@SyedAllaBaksh:~/hlink$ rm h11.txt
tasneem@SyedAllaBaksh:~/hlink$ ls
1.txt
tasneem@SyedAllaBaksh:~/hlink$ cat 1.txt
im from 1.txt
im appended one
im from h11.txt
tasneem@SyedAllaBaksh:~/hlink$

```

4. Remove original file of HARD link and observe?

```

tasneem@SyedAllaBaksh: ~/hlink
tasneem@SyedAllaBaksh:~/hlink$ ls
1.txt  h11.txt
tasneem@SyedAllaBaksh:~/hlink$ rm 1.txt
tasneem@SyedAllaBaksh:~/hlink$ cat h11.txt'
>
>
-bash: unexpected EOF while looking for matching `''
-bash: syntax error: unexpected end of file
tasneem@SyedAllaBaksh:~/hlink$ cat h11.txt'
tasneem@SyedAllaBaksh:~/hlink$ cat h11.txt
im from 1.txt
im appended one
im from h11.txt
tasneem@SyedAllaBaksh:~/hlink$

```

Here, if I remove the original file, it doesn't effect on hard link files content which were written in original file. Means, whatever we created in that original file will be saved in its hard link file, it won't remove.

- **Comparison among soft, hard, copy?**

(A). Soft link act as a reference of the original file. If we try to modify the original files, it'll also show its effects on its S.L files. If we remove or delete the original file, the soft link file also be deleted. Even if we try to access the S.L files, it throws an error. (It's like breaking the chain).

(B). Hard link act as a copy of its original file. It'll show the contents of original file as well, same as S.L files. But, if we remove or delete the original file, the data presented in that original file, will still available in it's hard link file, it doesn't effect on hard link files. We can still the H.L files by commands if we want to.

(C) Copy: if we try to copy a file, it'll duplicate that file data, and if we try to make some changes in that file, the copied file won't show any changes. Also, the copied file won't be deleted if we delete the original file.

- The only difference between copy and S.L files is, if we remove or modify the original file, the S.L file also be removed or modify accordingly.
- Difference between copy and hard link files are very simple, H.L files act as a mirrored file, if we delete the original file, the H.L file won't get effected same as copied file, but can modified according the original file, which we can't see in copied files.

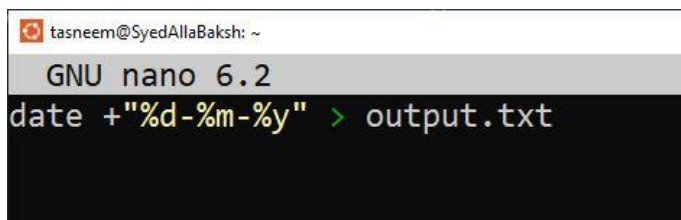
- **what are wget, curl commands differences?**

wget and curl commands help us to download the files from the server, we just have to paste the URL link. The only difference between wget and curl command is that:

1. wget <---- help us to download the files
2. curl <---- help us to transfer the data or from the server (URL)

SCRIPTS:

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



```
tasneem@SyedAllaBaksh: ~
GNU nano 6.2
date +%d-%m-%y > output.txt
```

(nano commands)



```
tasneem@SyedAllaBaksh:~$ nano test.sh
tasneem@SyedAllaBaksh:~$ ./test.sh
-bash: ./test.sh: Permission denied
tasneem@SyedAllaBaksh:~$ sudo chmod +x test.sh
tasneem@SyedAllaBaksh:~$ ./test.sh
tasneem@SyedAllaBaksh:~$ ls
01:06:47  1.py      12fghgAS.txt  Abc.txt      a123.txt     error.txt     log.txt       raname.txt    st
1          1.txt     1a.txt       BC12.txt     abc.txt      h1            'my name'    rename.txt    test.sh
1.c       123.txt  4.txt       Sunday26March  bc1.txt     hlink         out.txt      serverslog.txt test1
1.js      12a.txt  ABC.txt     a.out        cclog.txt   index.html   output.txt   serverslogs.txt test2
tasneem@SyedAllaBaksh:~$ cat output.txt
27-03-23
tasneem@SyedAllaBaksh:~$
```

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

```
tasneem@SyedAllaBaksh: ~  
tasneem@SyedAllaBaksh:~$ touch $(date +%d-%m-%y")  
tasneem@SyedAllaBaksh:~$ ls  
01:06:47 1.py 12fghgAS.txt ABC.txt a.out cclog.txt index.html output.txt serverslogs.txt test2  
1 1.txt 1a.txt Abc.txt a123.txt error.txt log.txt raname.txt st  
1.c 123.txt 27-03-23 BC12.txt abc.txt hl 'my name' rename.txt test.sh  
1.js 12a.txt 4.txt Sunday26March bc1.txt hlink out.txt serverslog.txt test1  
tasneem@SyedAllaBaksh:~$
```

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

```
tasneem@SyedAllaBaksh: ~  
GNU nano 6.2 date.sh *  
echo $(date +%r, %d-%m-%y"), $(whoami), [$(pwd)] > output.txt
```

(nano commands)

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

```
tasneem@SyedAllaBaksh: ~  
tasneem@SyedAllaBaksh:~$ nano date.sh  
tasneem@SyedAllaBaksh:~$ sudo chmod +x date.sh  
tasneem@SyedAllaBaksh:~$ ./date.sh  
tasneem@SyedAllaBaksh:~$ ls  
01:06:47 1.py 12fghgAS.txt ABC.txt a.out cclog.txt hlink out.txt serverslog.txt test1  
1 1.txt 1a.txt Abc.txt a123.txt date.sh index.html output.txt serverslogs.txt test2  
1.c 123.txt 27-03-23 BC12.txt abc.txt error.txt log.txt raname.txt st  
1.js 12a.txt 4.txt Sunday26March bc1.txt hl 'my name' rename.txt test.sh  
tasneem@SyedAllaBaksh:~$ cat output.txt  
12:43:22 AM, 28-03-23, tasneem, [/home/tasneem]  
tasneem@SyedAllaBaksh:~$
```

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

```
tasneem@SyedAllaBaksh: ~/23-03-28(01:30:58.1679947258)(+0530)  
tasneem@SyedAllaBaksh:~$ nano date.sh  
tasneem@SyedAllaBaksh:~$ ./date.sh  
tasneem@SyedAllaBaksh:~$ cat output.txt  
01:30:02 AM, 28-03-23, tasneem, [/home/tasneem]  
tasneem@SyedAllaBaksh:~$ mkdir $(date +%y-%m-%d(%T.%s)(%z)) && mv output.txt $(date +%y-%m-%d(%T.%s)(%z))  
tasneem@SyedAllaBaksh:~$ ls  
1 123.txt 27-03-23 Sunday26March cclog.txt index.html rename.txt test1  
1.c 12a.txt 4.txt a.out date.sh log.txt serverslog.txt test2  
1.js 12fghgAS.txt ABC.txt a123.txt error.txt 'my name' serverslogs.txt  
1.py 1a.txt Abc.txt abc.txt hl out.txt st  
1.txt '23-03-28(01:30:58.1679947258)(+0530)' BC12.txt bc1.txt hlink raname.txt test.sh  
tasneem@SyedAllaBaksh:~$ cd '23-03-28(01:30:58.1679947258)(+0530)'  
tasneem@SyedAllaBaksh:~/23-03-28(01:30:58.1679947258)(+0530)$ ls  
output.txt  
tasneem@SyedAllaBaksh:~/23-03-28(01:30:58.1679947258)(+0530)$ cat output.txt  
01:30:02 AM, 28-03-23, tasneem, [/home/tasneem]  
tasneem@SyedAllaBaksh:~/23-03-28(01:30:58.1679947258)(+0530)$
```


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

tasneem@SyedAllaBaksh: ~

```
GNU nano 6.2
echo $(date +"%d-%m-%y") >> date.txt
```

(commands in nano)

tasneem@SyedAllaBaksh: ~

```
GNU nano 6.2
*/2 * * * sat /home/tasneem/date.txt
```

(commands in crontab -e)

tasneem@SyedAllaBaksh: ~

```
tasneem@SyedAllaBaksh:~$ nano 2min.sh
tasneem@SyedAllaBaksh:~$ ./2min.sh
tasneem@SyedAllaBaksh:~$ ls
1          12a.txt          4.txt      a123.txt    error.txt    out.txt      test.sh
1.c        12fghgAS.txt  ABC.txt    abc.txt     hl           rename.txt   test1
1.js       1a.txt        Abc.txt    bc1.txt     hlink        rename.txt   test2
1.py       '23-03-28(01:30:58.1679947258)(+0530)''
1.txt      27-03-23      BC12.txt   cclog.txt   index.html   serverslog.txt
123.txt    2min.sh       Sunday26March
tasneem@SyedAllaBaksh:~$ sudo service cron start
* Starting periodic command scheduler cron
tasneem@SyedAllaBaksh:~$ crontab -e
No modification made
tasneem@SyedAllaBaksh:~$ crontab -e
crontab: installing new crontab
tasneem@SyedAllaBaksh:~$
```

[OK]

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

tasneem@SyedAllaBaksh: ~

```
GNU nano 6.2
mkdir -p backup/tdir && cp -r test1 backup/tdir
```

(commands in nano)

```
0 8 1,15 * * /home/tasneem/twice.sh
```

(commands in crontab -e)

tasneem@SyedAllaBaksh: ~

```
tasneem@SyedAllaBaksh:~$ nano twice.sh
tasneem@SyedAllaBaksh:~$ sudo chmod 777 twice.sh
tasneem@SyedAllaBaksh:~$ ./twice.sh
tasneem@SyedAllaBaksh:~$ ls
1          '23-03-28(01:30:58.1679947258)(+0530)''  Sunday26March  count1        local.sh      serverslogs.txt
1.c        27-03-23      a.out          a123.txt       date           log.txt       st
1.js       29-03-23      abc.txt        date.sh        date.txt      'my name'    test.sh
1.py       29-03-23,02:23:29  bc1.txt       dynac.sh       date.txt      output.txt   test1
1.txt      2min.sh      cclog.txt     error.txt      hlink         rename.txt   test2
123.txt    4.txt        count.txt     hl             rename.txt   serverslog.txt
12a.txt    ABC.txt      backup         hlink          s.sh          user.sh
12fghgAS.txt Abc.txt      count.txt     index.html     serverslog.txt wkend.txt
1a.txt     BC12.txt
tasneem@SyedAllaBaksh:~$ crontab -e
crontab: installing new crontab
tasneem@SyedAllaBaksh:~$
```

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

```

tasneem@SyedAllaBaksh: ~
tasneem@SyedAllaBaksh:~$ mkdir count && cd count && touch 1.txt 2.txt 3.txt 4.txt 21.txt
tasneem@SyedAllaBaksh:~/count$ ls
1.txt 2.txt 21.txt 3.txt 4.txt
tasneem@SyedAllaBaksh:~/count$ ls *.txt|wc -l
5
tasneem@SyedAllaBaksh:~/count$ cd
tasneem@SyedAllaBaksh:~$

```

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

```

tasneem@SyedAllaBaksh: ~

```

```

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

```

(commands in nano)

```

0 0 * * * /home/tasneem/dynac.sh

```

(commands in crontab -e)

```

tasneem@SyedAllaBaksh: ~
tasneem@SyedAllaBaksh:~$ nano dynac.sh
tasneem@SyedAllaBaksh:~$ ./dynac.sh
-bash: ./dynac.sh: Permission denied
tasneem@SyedAllaBaksh:~$ sudo chmod 777 dynac.sh
[sudo] password for tasneem:
tasneem@SyedAllaBaksh:~$ ./dynac.sh
tasneem@SyedAllaBaksh:~$ crontab -e
crontab: installing new crontab
tasneem@SyedAllaBaksh:~$ ls
1      1.txt      1a.txt      2min.sh     BC12.txt    abc.txt     count       error.txt   log.txt     rename.txt  test.sh
1.c    123.txt     '23-03-28(01:30:58.1679947258)(+0530)'  4.txt       Sunday26March  backup     date.sh     hl          serverslog.txt  test1
1.js   12a.txt     27-03-23    ABC.txt     a.out       bc1.txt     date.txt    hlink       out.txt     serverslogs.txt  test2
1.py   12fghgAS.txt 29-03-23    Abc.txt     a123.txt    cclog.txt   dynac.sh    index.html  raname.txt  st
tasneem@SyedAllaBaksh:~$

```

CRONTAB SCRIPTS LABS:

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

```

tasneem@SyedAllaBaksh: ~
GNU nano 6.2                                     user.sh
echo $(pwd) $(whoami) > output.txt

```

(nano commands)


```
tasneem@SyedAllaBaksh: ~$ nano user.sh
tasneem@SyedAllaBaksh:~$ sudo chmod 777 user.sh
[sudo] password for tasneem:
tasneem@SyedAllaBaksh:~$ ./user.sh
tasneem@SyedAllaBaksh:~$ ls
1      1.txt      2min.sh      BC12.txt      abc.txt      count      error.txt      log.txt      raname.txt      st      user.sh
1.c    123.txt      '23-03-28(01:30:58.1679947258)(+0530)'  4.txt      Sunday26March  backup      date.sh      hl      'my name'      rename.txt      test.sh
1.js   12a.txt      27-03-23      ABC.txt      a.out      bc1.txt      date.txt      hlink      out.txt      serverslog.txt      test1
1.py   12fghgAS.txt 29-03-23      Abc.txt      a123.txt    cclog.txt    dynac.sh      index.html  output.txt      serverslogs.txt  test2
tasneem@SyedAllaBaksh:~$ cat output.txt
/home/tasneem tasneem
tasneem@SyedAllaBaksh:~$
```

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

```
tasneem@SyedAllaBaksh: ~/29-03-23
tasneem@SyedAllaBaksh:~$ mkdir $(date +"%d-%m-%y") && cd $(date +"%d-%m-%y") && touch $(date +"%d-%m-%y,%H:%M:%S")
tasneem@SyedAllaBaksh:~/29-03-23$ ls
29-03-23,02:03:33
tasneem@SyedAllaBaksh:~/29-03-23$ tree
.
├── 29-03-23,02:03:33

0 directories, 1 file
tasneem@SyedAllaBaksh:~/29-03-23$
```

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.

```
tasneem@SyedAllaBaksh: ~/29-03-23,02:23:29
tasneem@SyedAllaBaksh:~$ nano local.sh
tasneem@SyedAllaBaksh:~$ sudo chmod 777 local.sh
tasneem@SyedAllaBaksh:~$ ./local.sh
tasneem@SyedAllaBaksh:~$ ls
1      12a.txt      2min.sh      a.out      count      hlink      output.txt      test.sh
1.c    12fghgAS.txt ABC.txt      a123.txt    date.sh      index.html  rename.txt      test1
1.js   1a.txt      ABC.txt      abc.txt     date.txt     local.sh    serverslog.txt  test2
1.py   '23-03-28(01:30:58.1679947258)(+0530)'  backup      dynac.sh    log.txt      serverslogs.txt user.sh
1.txt  27-03-23    BC12.txt     bc1.txt     error.txt    'my name'   st
123.txt 29-03-23    Sunday26March cclog.txt   hl      out.txt
tasneem@SyedAllaBaksh:~$ cat output.txt
02:22:36 29-03-23 tasneem /home/tasneem
tasneem@SyedAllaBaksh:~$ mkdir $(date +"%d-%m-%y,%T") && mv output.txt $(date +"%d-%m-%y,%T")
tasneem@SyedAllaBaksh:~$ ls
1      12fghgAS.txt 4.txt      abc.txt      dynac.sh      'my name'      test.sh
1.c    1a.txt      ABC.txt     backup      error.txt     out.txt        test1
1.js   '23-03-28(01:30:58.1679947258)(+0530)'  bc1.txt     h1          raname.txt    test2
1.py   27-03-23    BC12.txt    cclog.txt   hlink         rename.txt     user.sh
1.txt  29-03-23    Sunday26March count      index.html   serverslog.txt serverslogs.txt
123.txt 29-03-23,02:23:29 a.out      a123.txt    date.txt     log.txt       st
12a.txt 2min.sh
tasneem@SyedAllaBaksh:~$ cd 29-03-23,02:23:29
tasneem@SyedAllaBaksh:~/29-03-23,02:23:29$ ls
output.txt
tasneem@SyedAllaBaksh:~/29-03-23,02:23:29$ cat output.txt
02:22:36 29-03-23 tasneem /home/tasneem
tasneem@SyedAllaBaksh:~/29-03-23,02:23:29$
```

tasneem@SyedAllaBaksh: ~

GNU nano 6.2

local.sh

```
echo $(date +"%T %d-%m-%y") $(whoami) $(pwd) > output.txt
```

(nano commands)

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.

tasneem@SyedAllaBaksh: ~

GNU nano 6.2

```
cd count1 && ls *.txt|wc -l > count.txt
```

(nano commands)

tasneem@SyedAllaBaksh: ~/count1

```
tasneem@SyedAllaBaksh:~$ nano count.sh
```

```
tasneem@SyedAllaBaksh:~$ ./count.sh
```

```
tasneem@SyedAllaBaksh:~$ ls
```

1	12a.txt	29-03-23,02:23:29	Sunday26March	cclog.txt	date.sh	index.html	raname.txt	test.sh
1.c	12fghgAS.txt	2min.sh	a.out	count	date.txt	local.sh	rename.txt	test1
1.js	1a.txt	4.txt	a123.txt	count.sh	dynac.sh	log.txt	s.sh	test2
1.py	'23-03-28(01:30:58.1679947258)(+0530)'	ABC.txt	abc.txt	count.txt	error.txt	'my name'	serverslog.txt	twice.sh
1.txt	27-03-23	Abc.txt	backup	count1	hl	out.txt	serverslogs.txt	user.sh
123.txt	29-03-23	BC12.txt	bc1.txt	date	hlink	output.txt	st	wkend.txt

```
tasneem@SyedAllaBaksh:~$ cd count1
```

```
tasneem@SyedAllaBaksh:~/count1$ cat count.txt
```

```
6
```

```
tasneem@SyedAllaBaksh:~/count1$
```

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

tasneem@SyedAllaBaksh: ~

```
tasneem@SyedAllaBaksh:~$ nano s.sh
```

```
tasneem@SyedAllaBaksh:~$ ./s.sh
```

```
-bash: ./s.sh: No such file or directory
```

```
tasneem@SyedAllaBaksh:~$ ./s.sh
```

```
tasneem@SyedAllaBaksh:~$ ls
```

1	1a.txt	Abc.txt	cclog.txt	error.txt	output.txt	test1
1.c	'23-03-28(01:30:58.1679947258)(+0530)'	BC12.txt	count	hl	raname.txt	test2
1.js	27-03-23	Sunday26March	count.txt	hlink	rename.txt	user.sh
1.py	29-03-23	a.out	count1	index.html	s.sh	wkend.txt
1.txt	29-03-23,02:23:29	a123.txt	date	local.sh	serverslog.txt	
123.txt	2min.sh	abc.txt	date.sh	log.txt	serverslogs.txt	
12a.txt	4.txt	backup	date.txt	'my name'	st	
12fghgAS.txt	ABC.txt	bc1.txt	dynac.sh	out.txt	test.sh	

```
tasneem@SyedAllaBaksh:~$ crontab -e
```

```
crontab: installing new crontab
```

```
tasneem@SyedAllaBaksh:~$
```

tasneem@SyedAllaBaksh: ~

GNU nano 6.2

```
date +"%T" >> wkend.txt
```

(commands in nano)

```
*/2 * * * sat /home/tasneem/s.sh
```

(commands in crontab -e)

6. Take a backup of a folder daily twice?

tasneem@SyedAllaBaksh: ~

GNU nano 6.2

```
zip -r backup2.zip sample
```

(commands in nano)

```
0 6,18 * * * /home/tasneem/backup2.sh
```

(commands in crontab -e)

tasneem@SyedAllaBaksh: ~

```
tasneem@SyedAllaBaksh:~$ mkdir sample && cd sample && touch 1.txt 2.txt 3.txt
```

```
tasneem@SyedAllaBaksh:~/sample$ ls
```

```
1.txt 2.txt 3.txt
```

```
tasneem@SyedAllaBaksh:~/sample$ cd
```

```
tasneem@SyedAllaBaksh:~$ nano backup2.sh
```

```
tasneem@SyedAllaBaksh:~$ sudo chmod 777 backup2.sh
```

```
[sudo] password for tasneem:
```

```
tasneem@SyedAllaBaksh:~$ ./backup2
```

```
-bash: ./backup2: No such file or directory
```

```
tasneem@SyedAllaBaksh:~$ ./backup2.sh
```

```
adding: sample/ (stored 0%)
```

```
adding: sample/1.txt (stored 0%)
```

```
adding: sample/2.txt (stored 0%)
```

```
adding: sample/3.txt (stored 0%)
```

```
tasneem@SyedAllaBaksh:~$ sudo service cron start
```

```
* Starting periodic command scheduler cron
```

```
tasneem@SyedAllaBaksh:~$ crontab -e
```

```
crontab: installing new crontab
```

```
tasneem@SyedAllaBaksh:~$
```

```
tasneem@SyedAllaBaksh:~$ crontab -e
```

```
crontab: installing new crontab
```

```
tasneem@SyedAllaBaksh:~$ ls
```

```
1 12a.txt
```

```
1.c 12fghgAS.txt
```

```
1.js 1a.txt
```

```
1.py 2.txt
```

```
1.txt '23-03-28(01:30:58.1679947258)(+0530)'
```

```
123.txt 27-03-23
```

```
tasneem@SyedAllaBaksh:~$
```

```
29-03-23
```

```
29-03-23,02:23:29
```

```
2min.sh
```

```
3.txt
```

```
30-03-23
```

```
4.txt
```

```
ABC.txt
```

```
Abc.txt
```

```
BC12.txt
```

```
Sunday26March
```

```
a.out
```

```
a123.txt
```

```
abc.txt
```

```
backup
```

```
backup2.sh
```

```
backup2.zip
```

```
bc1.txt
```

```
cclog.txt
```

```
count
```

```
count.sh
```

```
count.txt
```

```
count1
```

```
date
```

```
date.sh
```

```
date.txt
```

```
dynac.sh
```

```
error.txt
```

```
h1
```

```
hlink
```

```
index.html
```

```
local.sh
```

```
log.txt
```

```
'my name'
```

```
out.txt
```

```
output.txt
```

```
raname.txt
```

```
rename.txt
```

```
s.sh
```

```
sample
```

```
serverslog.txt
```

```
serverslogs.txt
```

```
st
```

```
test.sh
```

```
test1
```

```
test2
```

```
twice.sh
```

```
user.sh
```

```
wkend.txt
```

[OK]