Linux Assignment

Links:

Soft link:

```
richansar@Richs-MacBook-Air softlink % tree
.
├── 1.txt
├── 11.txt -> 1.txt
├── 2.txt
└── 22.txt -> 2.txt

1 directory, 4 files
```

1. MODIFY any SOFT link and observer original?
⇒ The original file is also modified.

2. MODIFY original file of SOFT link and observe?
⇒ All the soft links are modified when the original is modified.

3. Remove any SOFT link and observe the original?
⇒ Original stays available and unchanged.

4. Remove original file of SOFT link and observe?
⇒ All the soft links of the original files are removed when the original is deleted.

Hard link:

```
richansar@Richs-MacBook-Air hardlink % tree
.
├── 1.txt
├── 11.txt
├── 2.txt
└── 22.txt

1 directory, 4 files
```

1. MODIFY any HARD link and observer 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.

3. Remove any HARD link and observer original?
⇒ removal of hard links doesn't affect 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.

3. Comparision among soft,hard,copy?

Soft link	Hard link
1. The path to the file, not the data in the file	Direct access to the source file's data
2. Changes made to one softlink will be reflected in the source file, and vice versa.	Changes to one hard link will have an effect on all hard linked files.
3. The removal of the source file removes all soft links.	The deletion of any hard linked file has no effect on other hard linked files.

4. What are wget , curl commands differences?

wget	curl
1. Primarily used for downloading files from the web using transfer protocols	Tool to transfer data from or to a server
2. 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
3. Used to download entire websites or parts of the website to view offline	Can be used in scripts to automate file transfer

---

Crontabs:

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

Mac Terminal:

```
richansar@Richs-MacBook-Air ~ % touch script.sh
richansar@Richs-MacBook-Air ~ % ls
1.js            Downloads       Pictures        original.txt    script.sh
Applications    Library         Public          pico.save       softlink.txt
Desktop         Movies          a.txt           posts           test1
Documents       Music           homebrew        scr1.sh
richansar@Richs-MacBook-Air ~ % nano script.sh
richansar@Richs-MacBook-Air ~ % chmod +x script.sh

richansar@Richs-MacBook-Air ~ % ./script.sh

Current directory: /Users/richansar
Username: richansar
richansar@Richs-MacBook-Air ~ % cat output.txt
Current directory: /Users/richansar
Username: richansar
richansar@Richs-MacBook-Air ~ %
```

nano/script code:

```
# get the current directory
cur_dir=$(pwd)

# get the current username
username=$(whoami)

# printing current directory and username
echo "current directory: $cur_dir"
echo "username: $username"

# redirecting to "output.txt"
echo "current directory: $cur_dir" > output.txt
echo "username: $username" >> output.txt
```

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

Mac Terminal:

```
richansar@Richs-MacBook-Air ~ % touch scr2.sh
richansar@Richs-MacBook-Air ~ % ls
1.js            Downloads       Pictures        original.txt    scr1.sh
Applications    Library         Public          output.txt      scr2.sh
Desktop         Movies          a.txt           pico.save       script.sh
Documents       Music           homebrew        posts           softlink.txt
richansar@Richs-MacBook-Air ~ % nano scr2.sh
richansar@Richs-MacBook-Air ~ % chmod +x scr2.sh

richansar@Richs-MacBook-Air ~ % ./scr2.sh
richansar@Richs-MacBook-Air ~ % ls
1.js            Documents       Music           homebrew        posts
2023-03-28      Downloads       Pictures        original.txt    scr1.sh
Applications    Library         Public          output.txt      scr2.sh
Desktop         Movies          a.txt           pico.save       script.sh
richansar@Richs-MacBook-Air ~ % cd 2023-03-28
richansar@Richs-MacBook-Air 2023-03-28 % ls
2023-03-28_19-50-26.txt
richansar@Richs-MacBook-Air 2023-03-28 %
```

script code:

```
mkdir "$(date +"%Y-%m-%d")" && touch "$(date +"%Y-%m-%d")/$(date +"%Y-%m-%d_%H-%M-%S").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.
Mac Terminal:

```
richansar@Richs-MacBook-Air ~ % touch scr3.sh
richansar@Richs-MacBook-Air ~ % ls
1.js            Documents       Music           homebrew        posts
2023-03-28      Downloads       Pictures        original.txt    scr1.sh
Applications    Library         Public          output.txt      scr2.sh
Desktop         Movies          a.txt           pico.save       scr3.sh
richansar@Richs-MacBook-Air ~ % nano scr3.sh
richansar@Richs-MacBook-Air ~ % chmod +x scr3.sh

richansar@Richs-MacBook-Air ~ % ./scr3.sh

Local Time: 11:18:32 PM
Local Date: Tuesday, March 28, 2023
Username: richansar
```

Continuation:

```
richansar@Richs-MacBook-Air ~ % ls
1.js                    Downloads               a.txt                   scr1.sh
2023-03-28              Library                 homebrew                scr2.sh
2023-03-28_23-18-32     Movies                  original.txt            scr3.sh
Applications            Music                   output.txt              script.sh
Desktop                 Pictures                pico.save               softlink.txt
Documents               Public                  posts                   test1
richansar@Richs-MacBook-Air ~ % cd 2023-03-28_23-28-32
cd: no such file or directory: 2023-03-28_23-28-32
richansar@Richs-MacBook-Air ~ % cat 2023-03-28_23-28-32
cat: 2023-03-28_23-28-32: No such file or directory
richansar@Richs-MacBook-Air ~ % cd 2023-03-28_23-18-32
richansar@Richs-MacBook-Air 2023-03-28_23-18-32 % ls
output.txt
richansar@Richs-MacBook-Air 2023-03-28_23-18-32 % cat output.txt
Local Time: 11:18:32 PM
Local Date: Tuesday, March 28, 2023
Username: richansar
Current Path: /Users/richansar
richansar@Richs-MacBook-Air 2023-03-28_23-18-32 %
```

Script code:

```
# get the current timestamp
timestamp=$(date +"%Y-%m-%d_%H-%M-%S")

# creating new directory with the current timestamp
mkdir $timestamp

# get the local time and date
date=$(date +"%A, %B %d, %Y")
time=$(date +"%r")

# get the current username and path
username=$(whoami)
path=$(pwd)

# printing the local time, date, username, and path
echo "Local Time: $time"
echo "Local Date: $date"
echo "Username: $username"
echo "Current Path: $path"

# redirecting everything to "output.txt"
echo "Local Time: $TIME" > $TIMESTAMP/output.txt
echo "Local Date: $DATE" >> $TIMESTAMP/output.txt
echo "Username: $USERNAME" >> $TIMESTAMP/output.txt
echo "Current Path: $PATH" >> $TIMESTAMP/output.txt
```

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.

Mac Terminal:

```
richansar@Richs-MacBook-Air ~ % touch scr4.sh
richansar@Richs-MacBook-Air ~ % nano scr4.sh
richansar@Richs-MacBook-Air ~ % chmod +x scr4.sh

richansar@Richs-MacBook-Air ~ % ./scr4.sh

There are       25 files in this folder.
richansar@Richs-MacBook-Air ~ % cat count.txt
25
richansar@Richs-MacBook-Air ~ %
```

script code:

```
count=$(ls -1 | wc -l)

echo "There are $count files in this folder."

echo $count > count.txt
```

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

Crontab source script:

```
*/2 * * * 6,7 /Users/richansar/scr9.sh
```

6. Take a backup of a folder daily twice?

Crontab source script:

```
0 0,12 * * * cp -r /Users/richansar/test1 /Users/richansar/backup
```

Scripts:

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

Mac Terminal:

```
richansar@Richs-MacBook-Air ~ % touch richoutput2.txt
richansar@Richs-MacBook-Air ~ % nano scr5.sh
richansar@Richs-MacBook-Air ~ % chmod +x scr5.sh
richansar@Richs-MacBook-Air ~ % ./scr5.sh
richansar@Richs-MacBook-Air ~ % cat richoutput2.txt
Wed Mar 29 06:25:00 IST 2023
richansar@Richs-MacBook-Air ~ %
```

script code:

```
date > richoutput2.txt
```

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

script code:

```
richansar@Richs-MacBook-Air ~ % touch $(date +%Y-%m-%d).txt

richansar@Richs-MacBook-Air ~ % ls
1.js                    Library                 original.txt
2023-03-28              Movies                  output.txt
2023-03-28_23-18-32     Music                   pico.save
2023-03-29.txt          Pictures                posts
Applications            Public                  richoutput1.txt
Desktop                 a.txt                   richoutput2.txt
Documents               count.txt               scr1.sh
Downloads               homebrew                scr2.sh
richansar@Richs-MacBook-Air ~ %
```

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

Mac Terminal:

```
richansar@Richs-MacBook-Air ~ % nano scr9.sh
richansar@Richs-MacBook-Air ~ % chmod +x scr9.sh
richansar@Richs-MacBook-Air ~ % crontab -e

crontab: installing new crontab
richansar@Richs-MacBook-Air ~ % ./scr9.sh
2023-03-30 14:51:25
richansar@Richs-MacBook-Air ~ %
```

script code:

```
date +"%Y-%m-%d %H:%M:%S"
```

Crontab source script:

```
*/2 * * * 6 /Users/richansar/scr9.sh
```

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

Crontab source script:

```
0 0 1,15 * * cp -r /Users/richansar/backup /Users/richansar/test1
```

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

script code:

```
richansar@Richs-MacBook-Air ~ % find Documents -type f | wc -l
       8
richansar@Richs-MacBook-Air ~ %
```

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

Crontab source script:

```
0 0 * * * touch /Users/richansar/$(date +\%Y-\%m-\%d).txt
```

Mac Terminal:

```
richansar@Richs-MacBook-Air ~ % crontab -e

crontab: installing new crontab
richansar@Richs-MacBook-Air ~ % ls

1.js                    Rich-project
2023-03-28              backup
2023-03-28_23-18-32     count.txt
2023-03-29.txt          hardlink
```