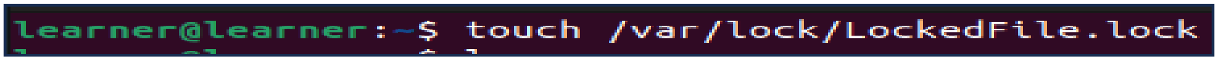
LAB SESSION# 08

1. Explain what the following commands do (with examples) and practice them:

- Lockfile
- Cksum
- Comm
- Csplit
- Chattr
- Touch

lockfile: This command is used to create lock files, which can be used to synchronize access to a file or resource.

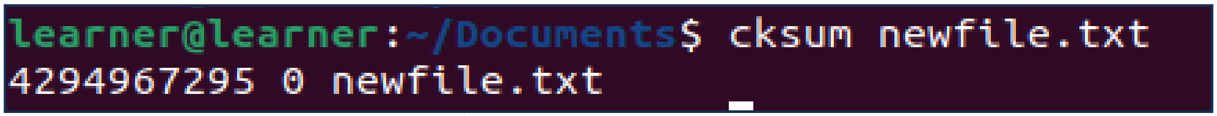
Example:



```
learner@learner:~$ touch /var/lock/LockedFile.lock
```

cksum: This command calculates a checksum value for a file, which can be used to verify file integrity.

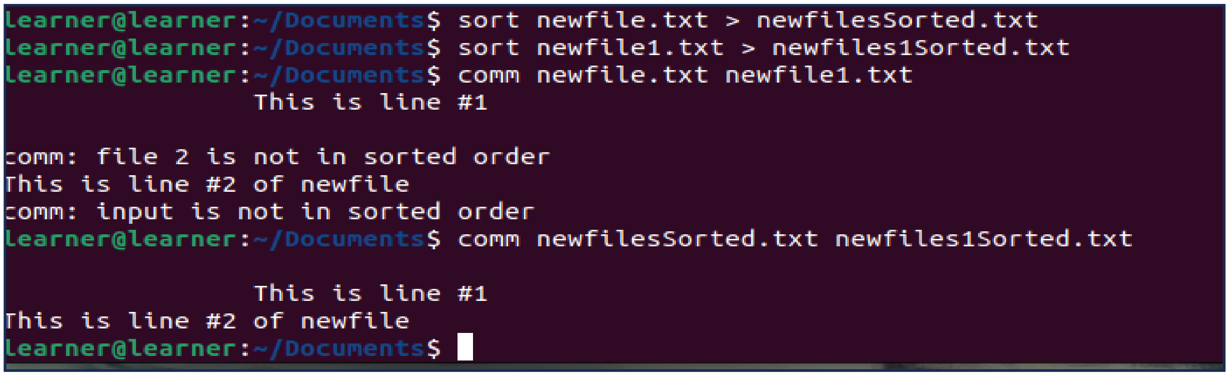
Example:



```
learner@learner:~/Documents$ cksum newfile.txt
4294967295 0 newfile.txt
```

comm: This command compares two sorted files line by line and displays the lines that are common or unique to each file.

Example:



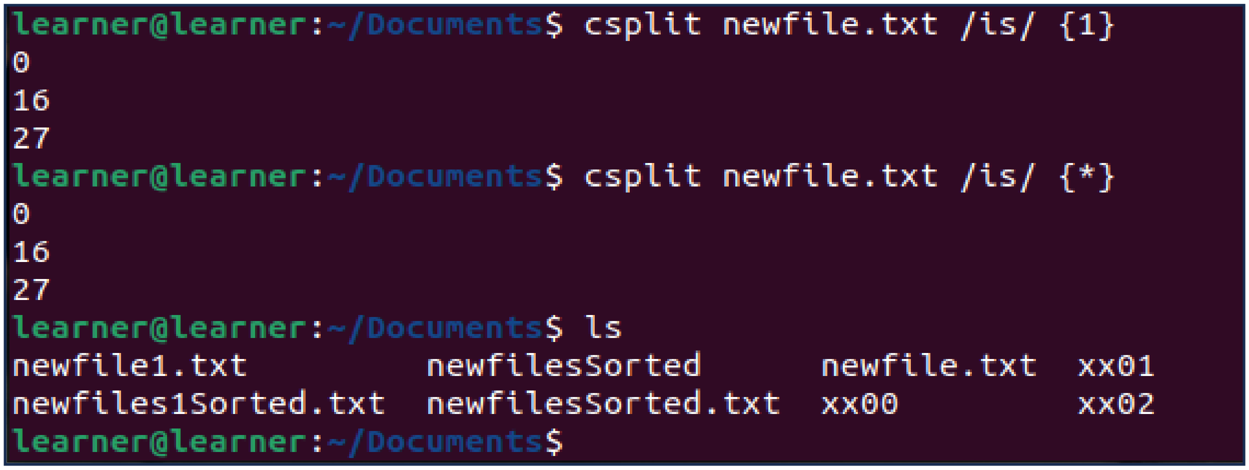
```
learner@learner:~/Documents$ sort newfile.txt > newfilesSorted.txt
learner@learner:~/Documents$ sort newfile1.txt > newfiles1Sorted.txt
learner@learner:~/Documents$ comm newfile.txt newfile1.txt
                This is line #1

comm: file 2 is not in sorted order
This is line #2 of newfile
comm: input is not in sorted order
learner@learner:~/Documents$ comm newfilesSorted.txt newfiles1Sorted.txt

                This is line #1
This is line #2 of newfile
learner@learner:~/Documents$
```

csplit: This command is used to split a file into multiple smaller files based on specific criteria.

Example:



```
learner@learner:~/Documents$ csplit newfile.txt /is/ {1}
0
16
27
learner@learner:~/Documents$ csplit newfile.txt /is/ {*}
0
16
27
learner@learner:~/Documents$ ls
newfile1.txt          newfilesSorted      newfile.txt   xx01
newfiles1Sorted.txt   newfilesSorted.txt  xx00          xx02
learner@learner:~/Documents$
```

chattr: This command is used to change the file attributes, such as making a file immutable or undeletable.

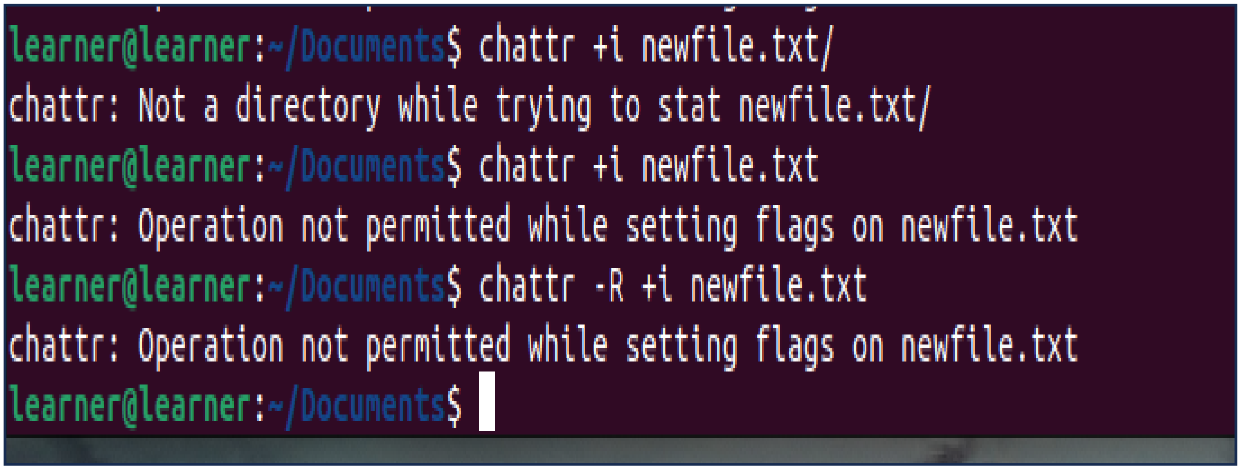
Common options are;

- -R : Recursively processes directories and their contents.
- -V : Be verbose, showing files processed.

Common attributes are;

- +i: Sets the immutable attribute, making the file unable to be modified, renamed, or deleted. This is a powerful way to protect critical system files.
- -i: Removes the immutable attribute, allowing the file to be modifies, renamed, or deleted.

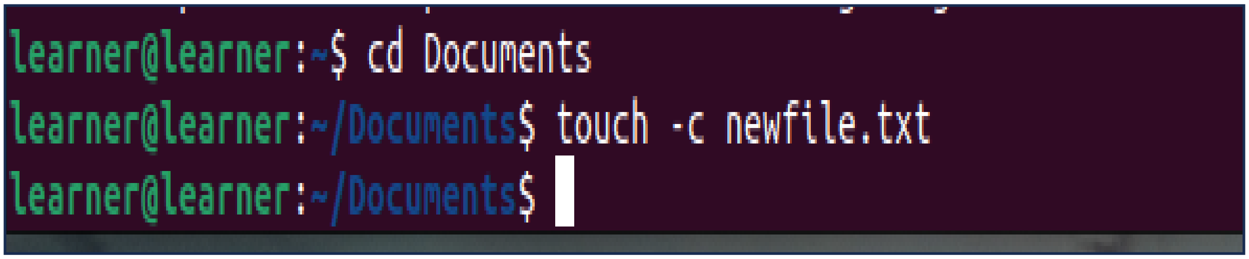
Example:



```
learner@learner:~/Documents$ chattr +i newfile.txt/
chattr: Not a directory while trying to stat newfile.txt/
learner@learner:~/Documents$ chattr +i newfile.txt
chattr: Operation not permitted while setting flags on newfile.txt
learner@learner:~/Documents$ chattr -R +i newfile.txt
chattr: Operation not permitted while setting flags on newfile.txt
learner@learner:~/Documents$
```

touch: This command is used to update the access and modification timestamps of a file, or create a new file if it doesn't exist.

Example:



```
learner@learner:~$ cd Documents
learner@learner:~/Documents$ touch -c newfile.txt
learner@learner:~/Documents$
```

The -c stops the touch file to create an empty file if the “newfile.txt” doesn't exist. It only updates the access and modification times for an existing file.

2. What do the following do:

- cat ch1
- cat ch1 ch2 ch3 > “your-practical-group”
- cat note5 >> notes

- cat > temp1
- cat > temp2 << “yourname”

Ans:

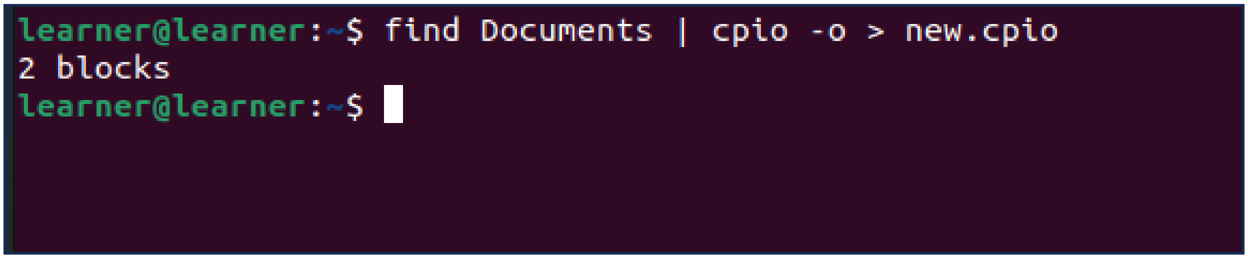
- **cat ch1 :** This command displays the content of the file named “ch1”, if the file exists , otherwise displays “No such file”(error message).
- **cat ch1 ch2 ch3 > G1:** This command concatenates the contents of files "ch1," "ch2," and "ch3" and redirects the combined output to a new file named "G1." If the file “G1” already exists, it will be overwritten.
- **cat note5 >> notes:** This command appends the content of the file "note5" to the end of the file "notes." If "notes" does not exist, it will be created.
- **cat > temp1:** This command allows us to input text from the keyboard, and the entered text will be saved into the file "temp1." To end the input and save the file, we should press Ctrl+D.
- **cat > temp2 << “QurratulAin”:** This command also allows us to input text from the keyboard, but it will be saved into the file "temp2" until a line is entered that matches the specified delimiter (QurratulAin). As soon as the line with the delimiter comes the input stops.

3. Practice the following commands and explain each:

- cpio
- sort
- fuser
- file

cpio: cpio stands for "copy in, copy out." It is a command-line utility for copying files into or out of a cpio.

Example:

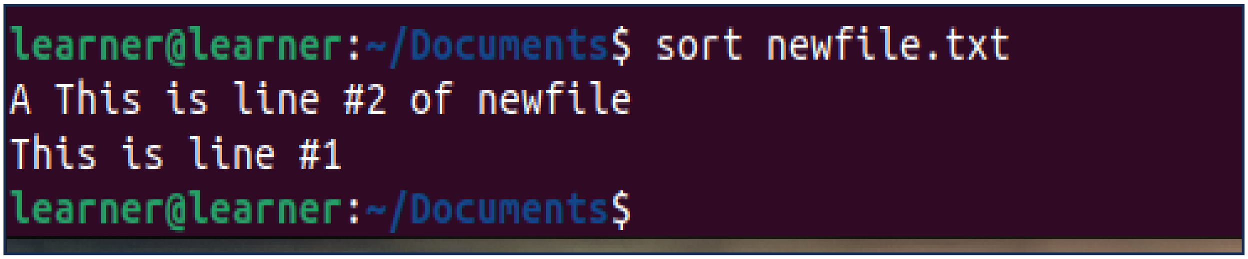


```
learner@learner:~$ find Documents | cpio -o > new.cpio
2 blocks
learner@learner:~$ 
```

The 2 block in the output is indicating that the cpio archive new.cpio has a size of 2 * 512 bytes, which is 1024 bytes. This size is based on the files and directories present in the "Documents" that were archived.

sort: The **sort** command is used to sort lines of text files in ascending or descending order.

Example:

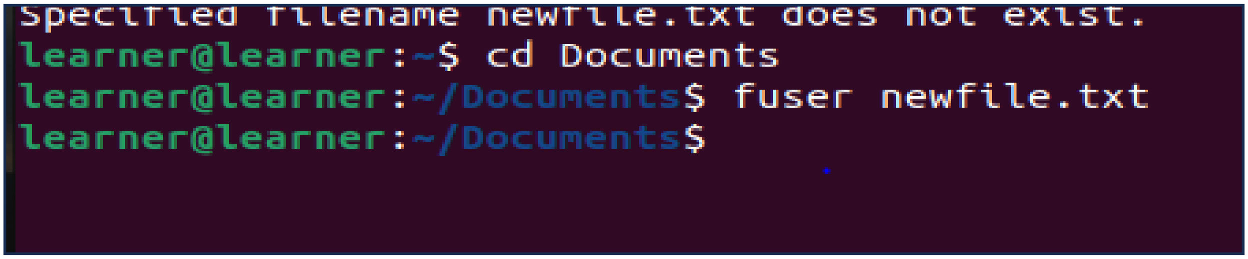


```
learner@learner:~/Documents$ sort newfile.txt
A This is line #2 of newfile
This is line #1
learner@learner:~/Documents$
```

Here the output is the alphabetically sorted lines of the file.

fuser: The **fuser** command is used to identify processes that are using a particular file or file system. It can be helpful in troubleshooting scenarios to determine which processes are accessing a file or directory.

Example:

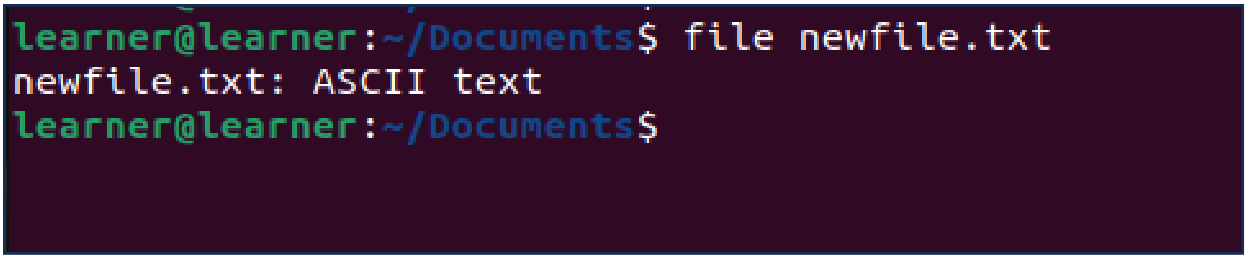


```
Specified filename newfile.txt does not exist.
learner@learner:~$ cd Documents
learner@learner:~/Documents$ fuser newfile.txt
learner@learner:~/Documents$
```

The output displaying nothing indicates that no processes is actively using the newfile.txt at this time.

file: The file command is used to determine the type of a file. It examines the file's contents and provides information about its type, such as whether it is a text file, binary file, or a specific type of data file.

Example:

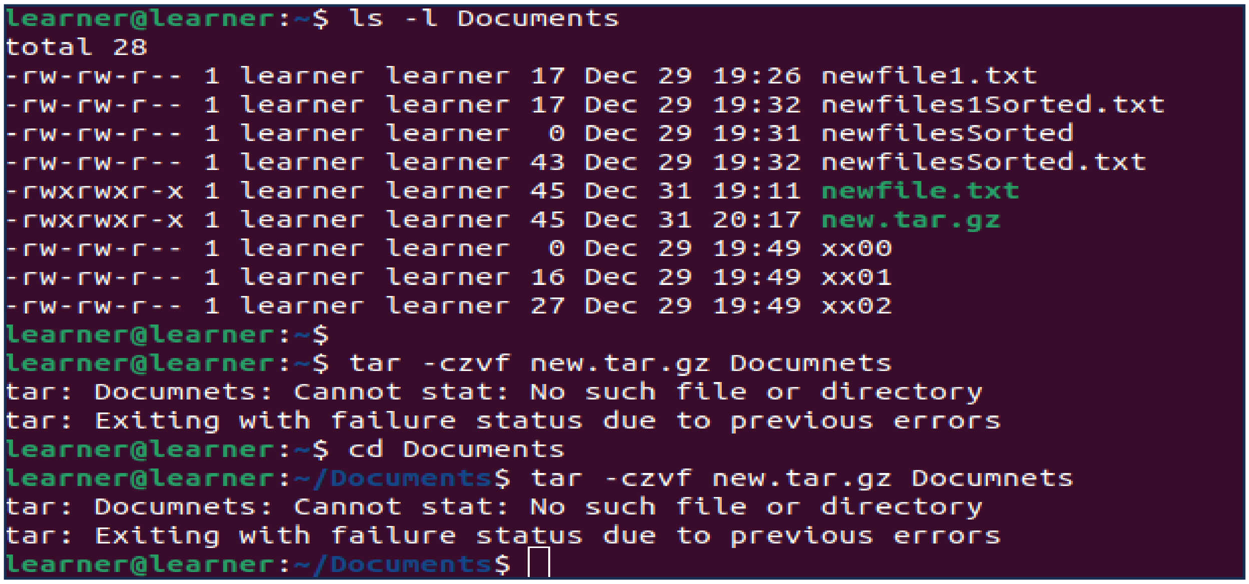


```
learner@learner:~/Documents$ file newfile.txt
newfile.txt: ASCII text
learner@learner:~/Documents$
```

4. What does the z option of the tar command do? Explain with examples.

The "z" option in the tar command is used to compress or decompress a tar archive using gzip. This option is commonly used when creating or extracting compressed tar archives.

Example:



```
learner@learner:~$ ls -l Documents
total 28
-rw-rw-r-- 1 learner learner 17 Dec 29 19:26 newfile1.txt
-rw-rw-r-- 1 learner learner 17 Dec 29 19:32 newfiles1Sorted.txt
-rw-rw-r-- 1 learner learner  0 Dec 29 19:31 newfilesSorted
-rw-rw-r-- 1 learner learner 43 Dec 29 19:32 newfilesSorted.txt
-rwxrwxr-x 1 learner learner 45 Dec 31 19:11 newfile.txt
-rwxrwxr-x 1 learner learner 45 Dec 31 20:17 new.tar.gz
-rw-rw-r-- 1 learner learner  0 Dec 29 19:49 xx00
-rw-rw-r-- 1 learner learner 16 Dec 29 19:49 xx01
-rw-rw-r-- 1 learner learner 27 Dec 29 19:49 xx02
learner@learner:~$
learner@learner:~$ tar -czvf new.tar.gz Documnets
tar: Documnets: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
learner@learner:~$ cd Documents
learner@learner:~/Documents$ tar -czvf new.tar.gz Documnets
tar: Documnets: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
learner@learner:~/Documents$
```

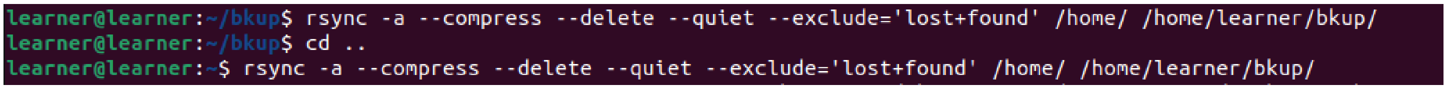
- **-c:** Create a new archive.
- **-z:** Filter the archive through gzip.
- **-v:** Verbosely list the files processed.
- **-f:** Use archive file specified.

5. Differentiate between cp and cpio command?

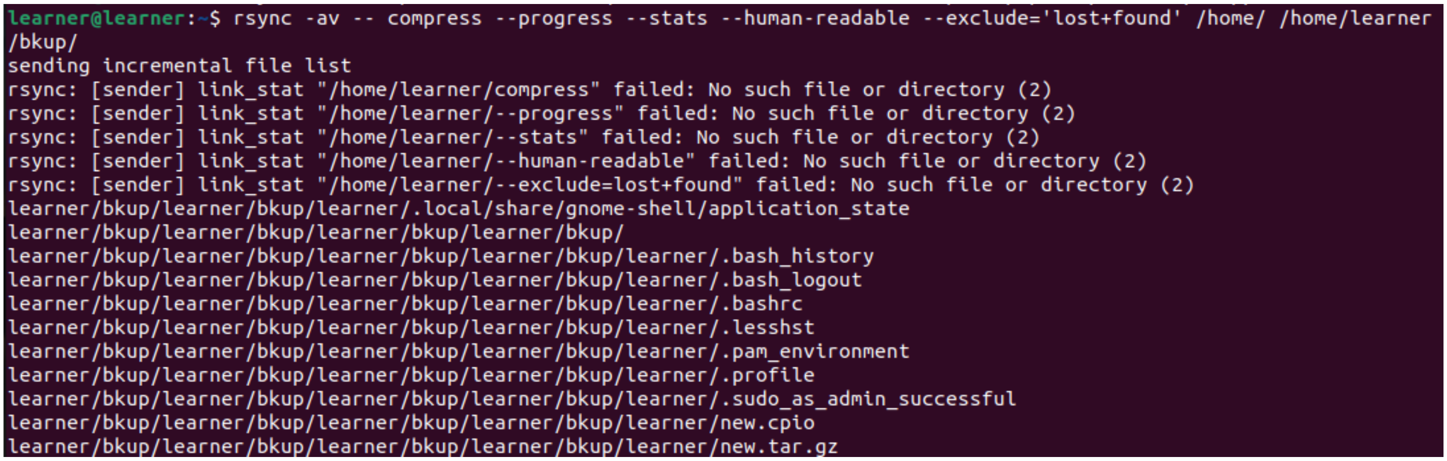
Ans: The **cp** command is primarily used to copy files or directories from one location to another while the **cpio** is used for creating or extracting archives. The cpio command is used often in combination with other commands like **find** to handle multiple files or directories. The cp has options for preserving attributes, recursive copying, forceful copy, etc. While the cpio has options for creating or extracting archives in various formats.

6. Write two commands to take the backup of your home-folder and all sub-folders. The destination folder should be /home/bkup. NOTE: size of backup should be smaller than original folder.

Ans:



```
learner@learner:~/bkup$ rsync -a --compress --delete --quiet --exclude='lost+found' /home/ /home/learner/bkup/
learner@learner:~/bkup$ cd ..
learner@learner:~$ rsync -a --compress --delete --quiet --exclude='lost+found' /home/ /home/learner/bkup/
```



```
learner@learner:~$ rsync -av -- compress --progress --stats --human-readable --exclude='lost+found' /home/ /home/learner
/bkup/
sending incremental file list
rsync: [sender] link_stat "/home/learner/compress" failed: No such file or directory (2)
rsync: [sender] link_stat "/home/learner/--progress" failed: No such file or directory (2)
rsync: [sender] link_stat "/home/learner/--stats" failed: No such file or directory (2)
rsync: [sender] link_stat "/home/learner/--human-readable" failed: No such file or directory (2)
rsync: [sender] link_stat "/home/learner/--exclude=lost+found" failed: No such file or directory (2)
learner/bkup/learner/bkup/learner/.local/share/gnome-shell/application_state
learner/bkup/learner/bkup/learner/bkup/learner/bkup/
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/.bash_history
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/.bash_logout
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/.bashrc
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/.lesshst
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/.pam_environment
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/.profile
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/.sudo_as_admin_successful
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/new.cpio
learner/bkup/learner/bkup/learner/bkup/learner/bkup/learner/new.tar.gz
```

7. What is the difference between the permissions 777 and 775 of the chmod command?

Ans:

- **Permission 777:** Owner, Group and Others have read, write and execute permissions.
- **Permission 775:** Owner and Group have read, write and execute permissions. Others have read and execute permissions and no write permissions.