CEW LAB # 08

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

- lockfile

  The lockfile command is used to create a lock file, which is a simple file that serves as a signal or indicator that a resource or file is currently being used by another process. It helps prevent conflicts when multiple processes or programs try to access the same resource simultaneously.

```
lockfile mylockfile.lock
```

- cksum

  The cksum command is used to calculate a unique checksum (a fixed-size numerical value) for a file. This checksum provides a way to verify the integrity of a file and detect any changes or corruption that may have occurred.

  E.g.: Suppose you have a file named document.txt, and you want to calculate the checksum for it using the cksum command.

```
$ cksum document.txt
```

  When you run this command, cksum will process the contents of document.txt and perform calculations to generate a unique checksum value. It will then display the checksum value along with the byte count and the file name. For example, the command might output something like this:

```
123456789 1000 document.txt
```

  In this example, 123456789 is the checksum value, 1000 represents the byte count of the file, and document.txt is the name of the file.To verify the file's integrity at a later time, you can rerun the cksum command on the file and compare the new checksum value with the one you obtained initially. If the checksum values match, it means the file has not been modified or corrupted. If the checksum values differ, it indicates that the file has undergone changes or corruption.

- comm

  The comm command is used to compare two sorted files line by line and display the lines that are common, unique to the first file, or unique to the second file. It helps identify differences and similarities between two text files.

```
comm file1.txt file2.txt
```

- csplit

  The csplit command is used to split a file into multiple parts based on specific patterns or line numbers. It allows you to divide a file into smaller sections and create separate files for each section.

  E.g.Suppose you have a file named mydata.txt, and you want to split it into separate files whenever a line matching a specific pattern occurs. Let's say the pattern you're looking for is the string "### SECTION ###".

  ```
  csplit mydata.txt '/### SECTION
  ```

- chattr

  The chattr command is used to change the attributes or properties of a file or directory in Linux systems. It allows you to set special flags on files that control various aspects of their behavior, such as making files immutable, undeletable, or append-only.

  E.g.You can use the chattr command with the +i option to make the file immutable,

  ```
  chattr +i important.txt
  ```

- touch

  The touch command updates the access and modification timestamps of a file or creates a new file if it doesn't exist. It is commonly used to create empty files or update the timestamp of existing files.

  E.g.
  ```
  touch myfile.txt
  ```

2. What do the following do?

- cat ch1

  display the content of file ch1 in the terminal.

- cat ch1 ch2 ch3 > "your-practical-group"

  This command concatenates the contents of files ch1, ch2, and ch3 and redirects the output to a new file named "your-practical-group".

- cat note5 >> notes

  This command appends the contents of the file 'note5' to the end of the file 'notes'.

- cat > temp1

  This command allows you to enter content interactively, which will be saved to the file temp1. After executing this command, anything you type will be written to temp1 until you press Ctrl+D to indicate the end of input.

- cat > temp2 << "yourname"

  This command allows you to enter content interactively and saves it to the file temp2. However, it uses a here-document (<<) with the delimiter set as "yourname". This means that you can enter multiple lines of text until you type the delimiter (in this case, "yourname"), and then press Enter. The entered content will be saved to the file temp2.

3. **Practice the following commands and explain each:**

- cpio
  The cpio command is used for creating and extracting archives in Linux. It is primarily used in combination with other commands like find to create or extract archives of files and directories.
- sort
  The sort command is used to sort lines of text in a file or from standard input in lexicographical order. It can sort lines alphabetically, numerically, or based on various criteria.
- fuser
  The fuser command is used to identify processes that are currently using or have opened specific files, directories, or sockets in a Linux system. It helps you determine which processes have active connections or access to a particular resource.
- file
  The file command is used to determine the type of a file or the nature of its contents. It examines the file's data and provides information about its format, encoding, or other characteristics.

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

The -z option in the tar command is used to enable gzip compression or decompression when creating or extracting archive files. It is commonly used in conjunction with the -c (create) or -x (extract) options. Here's how the -z option works:

1. **Creating a compressed archive**: When creating a new archive, the -z option allows you to compress the archive using gzip compression. By specifying -z, the resulting archive file will have the extension .tar.gz. For example:

```
tar -czf archive.tar.gz file1.txt file2.txt directory/
```

In this example, the tar command creates a compressed archive named archive.tar.gz containing file1.txt, file2.txt, and the directory/ directory.

2. **Extracting from a compressed archive**: When extracting files from an existing compressed archive, the -z option is used to decompress the archive on-the-fly. By specifying -z, tar will automatically detect the compression format (gzip) and decompress the archive while extracting the files. For example:

```
tar -xzf archive.tar.gz
```

In this example, the tar command extracts the files from the archive.tar.gz compressed archive. The compressed archive is automatically decompressed during the extraction process.

5. **Differentiate between cp and cpio command?**

- <u>cp</u>

  cp (short for copy) is used to copy files and directories from one location to another. cp typically takes source and destination arguments, specifying the files or directories to be copied and the destination location

  cp provides options to control various aspects of copying, such as preserving file attributes (-p option), copying recursively (-R or -r option), and handling symbolic links (-L or -P option).

- <u>cpio</u>

  cpio (short for copy in and out) is primarily used to create and extract archives, allowing you to copy multiple files and directories together in a single archive file.

  cpio usually operates on file lists provided through standard input (piped from other commands) or specified as arguments. It reads the list of files or directories to be included in the archive or extracted from the archive. cpio offers more flexibility in terms of creating and extracting archives. It supports different archive formats and offers options to preserve file attributes, specify the archive format, control verbosity, etc.

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.**

   1. **Command using compression (-z) option:**

```
rsync -az --progress /home/user/ /home/bkup/
```

   2. **Command using compression and incremental (-z --link-dest) option:**

```
rsync -az --link-dest=/home/bkup/previous /home/user/ /home/bkup/current
```

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

   777 grants full read, write, and execute permissions to the owner, group, and others, allowing complete access to the file or directory. With 775 permissions, the owner and group have full read, write, and execute permissions, allowing complete access. Others, however, only have read and execute permissions, restricting write access.