LAB 8

Q1.

1.lockfile:

Explanation: `lockfile` is used to create semaphore files, which can be used to synchronize access to a resource among multiple processes.

lockfile /tmp/my_lock_file

- - - - - - - - - - - - - - - - - -

rm -f /tmp/my_lock_file

2. cksum:

Explanation: cksum calculates a cyclic redundancy check (CRC) checksum for a file. It is often used to verify the integrity of files.

cksum my_file.txt

3. comm

Explanation: comm compares two sorted files line by line and writes the results to the standard output. It is useful for finding commonalities or differences between files.

comm file2.txt file3.txt

4. csplit

Explanation: csplit is used to split a file into sections determined by context lines.

csplit myfile.txt /pattern/ {*}

5. chattr

Explanation: chattr is used to change the attributes of a file on a Linux file system. Commonly used for making files immutable or setting certain attributes.

chattr +i myfile.txt # Make the file immutable

6. touch

Explanation: touch is used to create an empty file or update the access and modification times of an existing file.

touch newfile.txt

Q2.

- 'cat ch1' command prints the content of the file '**ch1**' to the output.
- 'cat ch1 ch2 ch3 > "your-practical-group" concatenates the contents of the files ch1, ch2 and ch3, then redirects the output to a new file "G2" in my case.
- 'cat note5 >> notes', this command appends the content of the file note5 to the the end of the file notes.
- 'cat > temp1', this command lets you enter text interactively which is then saved to the temp1 file. It will continue accepting input until we signal the end of file by (Ctrl+D).

- ‘cat > temp2 << “taha_khan”
  This command creates a document temp2 in which text entered repetitively is directed to the file until my name is entered “taha_khan” to indicate the end of the input, it acts as a delimiter.

Q3.

1. cpio:

Explanation: cpio is a command-line utility for creating or extracting archives, or copying files from one place to another. It is often used in combination with the find command.

find . -depth -print | cpio -ov > archive.cpio

This command finds all files in the current directory and its subdirectories, then creates a cpio archive named archive.cpio

2. sort

Explanation: `sort` is used to sort lines of text files or input from a pipeline.

sort file.txt > sorted_file.txt

This command sorts the lines in file.txt and then saves the output to sorted_file.txt

3. fuser

Explanation: fuser is used to identify processes using files or sockets. It provides information about processes that are using a particular file or directory.

fuser -u /var/log/syslog

This command shows which processes are using the `/var/log/syslog` file along with the usernames of the processes.

4. file

Explanation: file is used to determine the file type of a given file. It examines the content of a file to make an educated guess about its type

file mydocument.pdf

This command will output information about the type of the file mydocument.pdf, such as whether it is a PDF document, plain text, etc.

Q4.

The z option in the tar command is used to compress or decompress a tarball using gzip. When creating a tarball (tar -czf), it compresses the files using gzip, and when extracting a tarball (tar -xzf), it decompresses the tarball using gzip.

Here are examples of using the z option:

a) Creating a compressed tarball
   tar -czf archive.tar.gz /path/to/directory

   -c: Create a new archive.
   -z: Compress the archive using gzip.
   -f: Specify the name of the archive file (archive.tar.gz in this case).

/path/to/directory: The directory you want to archive.

This command will create a compressed tarball named archive.tar.gz containing the contents of the specified directory.

b) Extracting a compressed tarball
tar -xzf archive.tar.gz

-x: Extract files from an archive.

-z: Decompress the archive using gzip.

-f: Specify the name of the archive file (archive.tar.gz in this case).

This command will extract the contents of the compressed tarball archive.tar.gz into the current directory.

The z option is useful for reducing the size of the archive, which can be beneficial when transferring or storing large amounts of data. Keep in mind that the compression ratio depends on the type of files being compressed; binary files generally compress more effectively than text files.

Q5.

The `cp` and `cpio` commands are both used for copying files, but they have different functionalities and use cases.

1) `cp` Command:

- Functionality: The primary purpose of the `cp` command is to copy files or directories from one location to another.

- Usage:

cp source_file destination

cp -r source_directory destination

- Supports recursive copying (`-r` or `-R`) for copying directories and their contents.

- Provides various options for preserving attributes, symbolic links, etc.

2) `cpio` Command:

- Functionality: The `cpio` command is primarily used for creating or extracting archives. It can copy files from one location to another, but its main strength lies in archiving and extracting.

- Usage:

find . -depth -print | cpio -ov > archive.cpio

cpio -idv < archive.cpio

- Features:

- Commonly used in conjunction with the `find` command to select files for archiving.

- Supports various archive formats (`cpio -o` for creating, `cpio -i` for extracting).

Key Differences:

1. Functionality:

- `cp`: Mainly used for straightforward copying of files and directories.

- `cpio`: Primarily used for creating or extracting archives, with copying as a secondary function.

2. Usage:

- `cp`: Simple syntax for copying files or directories.

- `cpio`: Involves additional steps and often used in combination with other commands like `find` for archiving.

3. Use Cases:

- `cp`: Suitable for basic file copying needs.

- `cpio`: Ideal for creating archives, especially when dealing with complex directory structures or when selective archiving is required.

In summary, while both commands involve copying files, `cp` is more straightforward for general copying tasks, while `cpio` is specialized for archiving and extracting files in a more controlled manner.

Q6.

mkdir /home/bkup

rsync -a --exclude=/home/bkup /home/ /home/bkup/

This command performs the initial backup. It copies the contents of your home folder to the /home/bkup/ directory, excluding the bkup folder itself to prevent infinite recursion.

Q7.

chmod 777

Owner: Read, write, and execute (rwx).

Group: Read, write, and execute (rwx).

Others: Read, write, and execute (rwx).

This permission setting (777) grants full read, write, and execute permissions to the owner, the group, and others for the specified file or directory. It provides the most permissive level of access, and anyone can do anything with the file or directory.

chmod 775:

Owner: Read, write, and execute (rwx).

Group: Read, write, and execute (rwx).

Others: Read and execute (r-x).

This permission setting (775) grants read, write, and execute permissions to the owner and the group, while others (those who are not the owner or in the group) have only read and execute permissions. It is a more restrictive setting compared to 777 and is often used when you want to give a group of users the ability to modify files within a directory, but not necessarily the ability to modify the directory itself.