Introduction to Globbing

Globbing: It is the operation that recognizes the patterns and does the job of file path expansion.

When we work with files and directories, we can provide wildcards or patterns to match certain files and directories.
Such as

`*` (Asterisk): List contents of files inside every directory starting with hello and any number after it.

```Bash
$ ls hello*
    hello1:

    hello2:

    hello3:
```

`!` (Negation): List all contents of directories excluding a certain one.

```Bash
$ ls hello[!1]
    hello2:

    hello3:
```

`[3-4]` (Ranges): List all the contents of a file lying in a particular range

```Bash
$ ls hello[2-3]
    hello2:

    hello3:
```

`*.c` (Extension): List all the files that end with a `.c` extension.

```Bash
$ ls *.c
    example.c  sample.c
```

`*[!.c]` (Negation combined with extension): List all the files except a certain pattern

```bash
$ ls *[!.c]
     backup3

    hello1:

    hello2:

    hello3:
```

NOTE: Such operations can be performed using commands like `rm -rf` as well

Introduction to archiving and compressing

What is the difference between archiving and compressing?

Archiving is the process of combining one or more files into one single file format, for portability purposes
If we have two files called `file1.txt` and `file2.txt`, archiving will cause these files to become one single file and its size with be equal to the sum of the sizes of the files.
`file1.txt` -> 1kB
`file2.txt` -> 3kB
Resulting archived file: `file3.tar` -> 1kB + 3kB = 4kB

Compressing is the process of compressing one single file into a file that uses much less space.
If we have multiple files that we want to compress, we must first archive it to make it one file and then compress it.
After compressing the aforementioned `file3.tar`, we get `file3.tar.gz`.
`.tar`: Indicates the files have been archived using `tar`.
`.gz`: Indicates the file has been compressed using `gzip`(gunzip).

Archiving

In order to archive files, we use a command-line utility called `tar` (Tape ARchiver).

- To archive files and directories:

```bash
$ tar cvf source.tar ./*
```

Explanation:
`tar`: The `tar` command itself
`c`: Create archive
`v`: Verbose - Prints all actions being performed to the terminal (Disabled by default)
`f`: File
`source.tar`: The name of the archive (can be anything).
`./*`: Everything inside the current directories
The command can be read as, archive EVERYTHING INSIDE CURRENT DIRECTORY and CREATE a FILE with VERBOSITY called SOURCE.TAR.

- To extract a tar file or .tar.gz file:

```bash
$ tar xvf source.tar ./*
```

Explanation:
`x`: Extract

Compressing

In order to compress archived files or any files, we use a utitlity called `gzip` (gunzip).

- Compress a file using `gzip`:

```bash
$ gzip source.tar
```

This results in a file called `source.tar.gz`

Creating and managing users and groups

NOTE: Commands shown here must be run as root user by `sudo`.

Working with users

- Check what user you are currently logged in as

```bash
$ whoami
user
```

- Create a user:

```bash
$ sudo useradd -m person
```

The `-m` flag tells `useradd` to create a `HOME` directory.

- Set password for that user:

```
$ sudo passwd person
    New Password:
    Retype Password:
    Password updated successfully.
```

NOTE: In order to switch to a user, you must provide a password for it using `passwd`

- Switch to a user:

```
$ su person
Password:
```

Now you should be switched to that user.

- In order to delete a user

```
$ sudo userdel person
```

Working with groups

- To check all the groups that have been created, do,

```
$ cat /etc/group
```

The `group` file contains all the groups that are created.

- Create a group

```
$ sudo groupadd customers
```

- Delete a group

```
$ sudo groupdel customers
```

- Add a user to a group

```
$ sudo usermod -a -G customers person
```

Explanation
- `-a` Means append
- `-G` To Group

NOTE: Put the name of the group before the user