Linux Commands

1. You have a file with permissions -rw-r--r--, and you run chmod +x file.sh. What happens?

Ans: The file with permission -rw-r--r-- denotes owner: read/write, group: read, others: read. When chmod +x command is executed for the file then execution bit is added for everyone. Then the permissions become -rwxr-xr-x, So the file becomes executable by everyone.

2. What is the difference between chmod 744 file.txt and chmod u=rwx,go=r file.txt?

Ans: Both the command denotes the same permissions -rwxr--r–. The difference lies between specify mode. chmod 744 is numeric way of declaring permission and chmod u=rwx,go=r is symbolic and it is more readable.

3. What is the sticky bit, and when should you use it?

When sticky bit (t) is declared for the file only the root user or owner can delete or rename files even if others have write access. It should be used for directories like /tmp so that many users have write access but they can't delete or rename it.

Eg:

```
chmod +t bench.py
ls -ld    # Shows drwxrwxrwt
```

4. You are told to give the owner full access, group only execute, and others no permissions. What symbolic command achieves this?

```
chmod u=rwx,g=x,o= bench.py
```

By using the command we gave owner full access(Read,Write,Execute) and the group executable access (x -> execute) and others no access.

5. What is umask, and why is it important?

When a file or directory is created it starts with default permissions. For the file it starts with permission 666 and for the directory is 777. If umask is provided then the umask value is subtracted from default permission value leads to required permission.

Eg:
umask 0077
Files: `666 - 077 = 600` (rw-------)
Dirs: `777 - 077 = 700` (rwx------)

6. If the umask is 022, what are the default permissions for a new file and a new directory?

Default Permission before Mask
File : 666
Directory : 777

If a umak with value 022 is provided, then the value will be subtracted from the file and directory default values.

File : 666-022 = 644
Directory : 777 - 022 = 755

File:
- Owner: `rw`
- Group: `r--`
- Others: `r--`

Directory:
- Owner: `rwx`
- Group: `r-x`
- Others: `r-x`

7. Why is umask often set to 002 in development environments but 027 or 077 in production?

Development: umask 002

Default file perms: 664 → rw-rw-r--

Default dir perms: 775 → rwxrwxr-x

Use case: Developers collaborate within the same group.

Group members can read/write/edit shared files.

Ideal for team projects where code is co-owned**.**

Production:

```
umask 027
```

Files: 640 → rw-r-----
Dirs: 750 → rwxr-x–

Group can read but not write.

Others have no access.

Suitable when limited team needs read-only access.

```
umask 077
```

Files: 600 → rw-------

Dirs: 700 → rwx------

Only the owner has access.

Ideal for private data like credentials, logs, or configs.

8. UserAdd vs AddUser

Useradd is a low-level system utility that is available on all Linux distributions and requires explicit options to be passed (e.g., to create a home directory or set a shell). It is suitable for scripting and automation because of its non-interactive nature

Adduser is a high-level command that is available by default in Debian-based systems like Ubuntu. It provides an interactive prompt that guides you through user creation, asking for a password, full name, and other optional details