**Linux Questions:**

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

The chmod command changes the permission of the file

* Original permissions: -**rw-r--r--**

→ Owner: read + write

→ Group: read

→ Others: read

* After chmod +x:

→ New permissions: -**rwxr-xr-x**

→ Owner: read, write, execute

→ Group: read, execute

→ Others: read, execute

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

Both commands achieve the same result, just written in different formats:

* chmod 744 file.txt is numeric mode, where:
  + 7 = read (4) + write (2) + execute (1) → for user
  + 4 = read (4) → for group
  + 4 = read (4) → for others
* chmod u=rwx,go=r file.txt is symbolic mode, explicitly setting:
  + u=rwx → user gets read, write, and execute
  + g=r → group gets read-only
  + o=r → others get read-only

Resulting permissions for both: **-rwxr--r--**

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

The sticky bit is a special permission used on directories. When set, it allows only the file's owner or root to delete or rename files within the directory, even if others have write access to that directory.

**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 file**

* u=rwx → user (owner): read, write, execute
* g=x → group: execute only
* o= → others: no permissions

Final permissions**: -rwx--x---**

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

umask (user file creation mask) defines the default permission restrictions for newly created files and directories.

* It subtracts permissions from the system’s defaults:
* Files: default max = 666 (read & write for all)
* Directories: default max = 777

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

File: 666 - 022 = 644 → rw-r--r--

Directory: 777 - 022 = 755 → rwxr-xr-x

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

* 002 in development:

Allows files/directories to be group-writable.

Ideal for collaboration, where multiple users are part of the same group.

* 27 or 007 in production:

Used to restrict access for security.

8. useradd vs adduser

**useradd**:

* A **low-level** command available on all Unix-like systems.
* Does **not** prompt for user details (like password) unless options are explicitly passed.

**adduser**:

* A **higher-level** command (often a shell script).
* **Interactive**: prompts for password, full name, etc.
* Uses useradd internally.