srushti

Assignment-5

1. What is a shell in Linux OS? How many categories of shell is currently exists in Linux? Why bash shell is very popular in Linux distribution? (CO2)

   A shell in Linux is a command-line interpreter that acts as a bridge between the user and the kernel (the core of the operating system).

   Categories of shell:-

   1) Command-line shells (text-based)
   2) Graphical shell

   - Default shell in most Linux distributions (like Ubuntu, Fedora, Debian).
   - User-friendly — has command history, auto-completion, and shortcuts.
   - Backward compatible with sh.
   - Efficient and stable, used by both beginners and system administrators.
   - Open-source and customizable — can be modified for different needs.

2. What does the ls -Z command display?

   -rw-r--r--. user user 1024 Oct 12
   file1.txt
   unconfined_u:object_r:user_home_t:s0

3. Write a command to list all hidden files in the current directory.

   - ls -a

4. Explain the difference between hard links and soft links (symbolic links) in Linux.

   Hard Link:-

   - A new directory entry pointing to the same inode as the original file. Essentially, it's another name for the same file data.
   - Shares the same inode number as the original file.
   - Cannot span across different file systems. Must be on the same file system as the original file.
   - Can only link to files. Cannot link to directories.
   - If the original file is deleted, the hard link still works because it points directly to the data. The data is only removed when all hard links are deleted.
     Ex: ln file1.txt faile2.txt

   Soft Link:-

- A new file that contains the path to another file or directory. It acts as a pointer or shortcut.
- Has its own unique inode number.
- Can span across different file systems. Can link files or directories on different partitions or network drives.
- Can link to files or directories.
- If the original file or directory is deleted or moved, the soft link becomes broken (dangling) because its target path no longer exists.
- Ex: ln -s file.txt link1.txt

5. A file has permissions -rwxr-x--x. Explain who can read, write, and execute it.
   With -rwxr-x--x permissions, the owner can read, write, and execute the file. The group can only read and execute it, while others can only execute it.
   - Owner: The user who owns the file has read (r), write (w), and execute (x) permissions.
   - Group: Members of the file's group have read (r) and execute (x) permissions, but cannot write to it.
   - Others: Anyone else on the system has only execute (x) permission. They cannot read or write to the file.

6. Write the command to change the group ownership of a file data.txt to group staff.
   - ➢ chgrp staff data.txt

7. Why is it dangerous to give 777 permissions to a file? Explain with an example.
   a file 777 permissions (read, write, and execute for owner, group, and others) is dangerous because it grants unrestricted access to that file to anyone who can access the system. This violates the principle of least privilege, a fundamental security concept that dictates users and processes should only have the minimum permissions necessary to perform their functions.
   Ex:
   suppose if I have a script backup.sh that runs automatically

```
#!/bin/bash
cp /important/data /backup/location
chmod 777 backup.sh
echo "rm -rf /important/data" >
backup.sh
```

Now, every time backup.sh runs, it could delete or corrupt your important data, because the script is writable and executable by anyone.

8. What is the difference between apropos (i.e., man -k) and whatis (i.e., man -f)?

Feature	**apropos (man -k)**	**whatis (man -f)**
Search Scope	Anywhere in name and description	Exact command name only (before the dash)
Use Case	When the command is unknown; searching by keyword	When the command is known; quick description
Output	Many partial matches possible	Only exact matches
Typical Syntax	apropos keyword/man -k keyword	whatis cmd/man -f cmd

9. Write a command to redirect the error output of a command to a file named error.log.
    - ➢ command 2> error.log

10. How can you use the tee command to append output to a file instead of overwriting it?
    - ➢ command | tee -a filename