

# Linux Programming: Assignment 5

**Name: Nanditha M**

**USN:ENG24CY0038**

**ROLL NO:7**

**SECTION:B**

## 1. What is a shell in Linux OS? How many categories of shell currently exist in Linux? Why is bash shell very popular in Linux distribution?

- **Shell:** A shell is a command-line interpreter that allows users to interact with the operating system by executing commands. It acts as a bridge between the **user** and the **kernel**.
- **Categories of Shells** in Linux:
  1. **Bourne Shell (sh)**
  2. **Bourne Again Shell (bash)**
  3. **C Shell (csh)**
  4. **Korn Shell (ksh)**
  5. **Z Shell (zsh)**
  6. **Fish Shell (friendly interactive shell)**
- **Why bash is popular:**
  - Default in most Linux distributions.
  - Backward-compatible with sh.
  - Supports scripting, command history, aliases, job control.
  - Large community support and portability.

## 2. What does the ls -Z command display?

- ls -Z is used to **display SELinux (Security-Enhanced Linux) security contexts** of files.
- It shows additional attributes like **user, role, type, and level**.

- Example: `ls -Z file.txt`  
Output: `-rw-r--r--. user1 user1 unconfined_u:object_r:user_home_t:s0 file.txt`

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

- Hidden files start with `.` in Linux.
- Command:  
`ls -a`  
to list only hidden files:  
`ls -d .*`

**4. Difference between hard links and soft links in Linux.**

Feature	Hard Link	Soft Link (Symbolic Link)
Definition	Another directory entry pointing to the same inode	A separate file that points to the original file's path
Inode	Shares same inode as original file	Has a different inode
File existence	Remains even if original is deleted	Becomes broken if original is deleted
Across file systems	Cannot span multiple file systems	Can span across file systems
Directory linking	Not allowed	Allowed
Example	<code>ln file1 file2</code>	<code>ln -s file1 link1</code>

**5. A file has permissions `-rwxr-x--x`. Explain who can read, write, and execute it.**

- Format: `-rwxr-x--x`
  - **Owner (user):** `rwx` → can read, write, execute
  - **Group:** `r-x` → can read, execute, but **not write**
  - **Others:** `--x` → can only execute

**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.**

- 777 means **read, write, and execute for everyone** (owner, group, others).
- Dangerous because:
  - Anyone can modify or delete the file.
  - Anyone can execute it (security risk if it's a script or binary).
- Example:
- chmod 777 script.sh
  - Any user could **inject malicious code** into script.sh and execute it, compromising the system.

## **8. Difference between apropos (man -k) and whatis (man -f).**

- **apropos / man -k**: Searches the manual page names and descriptions for a keyword.  
Example:
  - apropos copy  
→ Lists all commands related to "copy".
- **whatis / man -f**: Displays a **short one-line description** of a command.  
Example:
  - whatis cp  
→ Output: cp (1) - copy files and directories.

## **9. Write a command to redirect the error output of a command to a file named error.log.**

- Using 2> to redirect **stderr**:
- command 2>error.log
- Example:

- `ls /nonexistent 2> error.log`

## **10. How can you use the tee command to append output to a file instead of overwriting it?**

- By default, tee overwrites files. Use -a option to append.
- Example:

`ls | tee -a output.txt`