**Q1. Report on Different Types of Linux Shells**

A **Linux shell** is a program that allows users to interact with the operating system. It takes the commands you type and tells the computer what to do. There are several types of shells, each with its own features and style. The most common ones are:

**1. sh (Bourne Shell)**

* **Full Name**: Bourne Shell
* **Creator**: Stephen Bourne
* **Introduced**: Late 1970s
* **File Name**: sh

**Key Features:**

* One of the earliest Unix shells.
* Simple and stable.
* Used in many shell scripts, especially on older systems.
* Does not support advanced features like command history.

**Use:**

* Great for writing basic shell scripts.
* Often used as a default shell in older systems.

**2. csh (C Shell)**

* **Full Name**: C Shell
* **Creator**: Bill Joy
* **Introduced**: Late 1970s
* **File Name**: csh

**Key Features:**

* Syntax similar to the C programming language.
* Supports features like history and job control.
* Easier for programmers who know C.

**Use:**

* Useful for interactive work.
* Less commonly used for scripting due to some quirks.

**3. ksh (Korn Shell)**

* **Full Name**: Korn Shell
* **Creator**: David Korn
* **Introduced**: 1980s
* **File Name**: ksh

**Key Features:**

* Combines features of both sh and csh.
* Supports scripting, command history, and job control.
* Faster than many other shells.

**Use:**

* Good for both scripting and interactive use.
* Popular in enterprise and commercial Unix systems.

**4. bash (Bourne Again Shell)**

* **Full Name**: Bourne Again Shell
* **Creator**: Brian Fox (for GNU Project)
* **Introduced**: Late 1980s
* **File Name**: bash

**Key Features:**

* Most popular shell today.
* Backward compatible with sh.
* Has many useful features: command history, auto-completion, scripting tools, and more.
* Default shell on most Linux distributions.

**Use:**

* Great for beginners and experts alike.
* Widely used for writing scripts and daily command-line tasks.

**Summary Table:**

| **Shell** | **Based On** | **Good For** | **Key Features** |
| --- | --- | --- | --- |
| sh | Original Unix shell | Simple scripts | Basic, stable |
| csh | C language style | Interactive use | History, C-like syntax |
| ksh | sh + extras | Scripts and performance | Fast, powerful |
| bash | Improved sh | General use | Most features, user-friendly |

**Conclusion**

Each shell has its own strengths. If you're just starting out, **bash** is the best choice because it is powerful and user-friendly. For advanced users or specific tasks, **sh**, **csh**, and **ksh** can be useful too. Understanding these shells helps you work better in Linux environments.

Q2. Copy, move, and remove files using cp, mv, and rm commands.

**Working with Files in Linux**

In Linux, you can manage files using simple commands in the **terminal**. Here are three basic ones:

**1. Copy Files – cp Command**

The cp command is used to **copy files or directories**.

**Syntax:**

cp [source] [destination]

**Examples:**

* Copy a file:
* cp file1.txt file2.txt

This creates a copy of file1.txt named file2.txt.

* Copy a file to a folder:
* cp file1.txt /home/user/Documents/
* Copy a folder and its contents:
* cp -r myfolder/ /home/user/Backup/

Use -r (recursive) to copy directories.

**2. Move or Rename Files – mv Command**

The mv command is used to **move or rename** files and directories.

**Syntax:**

mv [source] [destination]

**Examples:**

* Rename a file:
* mv oldname.txt newname.txt
* Move a file to another folder:
* mv file.txt /home/user/Desktop/
* Move and rename at the same time:
* mv file.txt /home/user/Desktop/newfile.txt

**3. Remove Files – rm Command**

The rm command is used to **delete files or directories**.

**Syntax:**

rm [file\_or\_directory]

**Examples:**

* Delete a file:
* rm file.txt
* Delete multiple files:
* rm file1.txt file2.txt
* Delete a folder and all its contents:
* rm -r myfolder/
* Force delete without asking:
* rm -rf myfolder/

**Be careful** with -rf — it deletes everything without confirmation.