**Task- 1**

| **Symbol** | **Meaning** | **Example** |
| --- | --- | --- |
| **.** | **Any character except newline** | **a.b matches acb, a1b** |
| **^** | **Start of string** | **^abc matches abc at beginning** |
| **$** | **End of string** | **abc$ matches abc at end** |
| **\*** | **0 or more of the previous token** | **ab\*c matches ac, abc, abbc** |
| **+** | **1 or more of the previous token** | **ab+c matches abc, abbc** |
| **?** | **0 or 1 of the previous token (optional)** | **ab?c matches ac, abc** |
| **`** | **`** | **Alternation (or)** |
| **()** | **Grouping** | **(abc)+ matches abc, abcabc** |
| **[]** | **Character class** | **[abc] matches a, b, or c** |
| **[^]** | **Negated character class** | **[^abc] matches anything except a, b, c** |
| **\** | **Escape character** | **\. matches a literal dot .** |

**Task- 2**

## **Key Features of Linux:**

### **1. Open Source**

* In this Source code is freely available.
* Anyone can view, modify, and distribute it.
* This encourages community-driven development and customization**.**

### **2. Multiuser**

* Multiple users can use the system at the same time.
* Each user has their own permissions, files, and processes.

### **3. Multitasking**

* Linux can run multiple tasks simultaneously without affecting performance (e.g., downloading files while editing documents).

### **4. Security**

* Strong file permission system (read, write, execute).
* Supports firewalls, encryption, and user authentication.
* Commonly used in secure environments like servers and government systems.

### **5. Stability and Reliability**

* Linux is known for its uptime and crash resistance.
* Many servers run Linux for years without a reboot.

### **6. Portability**

* Linux runs on a wide variety of hardware platforms: PCs, servers, smartphones (Android), routers, Raspberry Pi, supercomputers, etc.

### **7. Command Line Interface (CLI) & GUI**

* Powerful CLI with tools like bash, grep, awk, sed.
* Multiple desktop environments (GUI) like GNOME, KDE, XFCE.

### **8. Modular & Customizable**

* Kernel is modular — you can load/unload modules as needed.
* Highly customizable OS; you can build your own version (e.g., using LFS - Linux From Scratch).

### **9. Package Management**

* Software is distributed via packages using tools like:  
  + apt (Debian/Ubuntu)
  + yum or dnf (Fedora/RHEL)
  + pacman (Arch Linux)

### **10. Support for Networking**

* Robust built-in networking tools.
* Supports advanced networking features like routing, DNS, FTP, SSH, NFS, and VPNs.

### **11. Virtualization & Containers**

* Supports Docker, LXC, KVM, and other virtualization/container tools.
* Great for cloud and DevOps environments.

### **12. Large Ecosystem**

* Used in Android, Chrome OS, web servers (Apache, Nginx), cloud platforms, IoT, etc.

**Task - 3**

The kernel is the core part of a linux operating system.

It acts like a bridge between your hardware (CPU, memory, etc.) and your software (apps, browsers, games, etc.).

Let’s think of it as the brain of the linux operating system.

**Task- 4**

BASH is a command-line shell and scripting language used in many Linux and Unix-based systems.

It is:

* A program that takes your commands (typed in the terminal) and tells the computer what to do.
* A successor to the original Bourne Shell (sh), with more features and flexibility — hence the name "Bourne Again" Shell (a pun on "born again").

You can also run Linux commands and In the Linux terminal (default shell in many distributions like Ubuntu)

For server maintenance and DevOps tasks

In shell scripts and CI/CD pipelines

On macOS, WSL (Windows Subsystem for Linux), and Docker containers.

**Task- 5**

Linux and Windows are both operating systems, but they work in different ways and are made for different kinds of users.

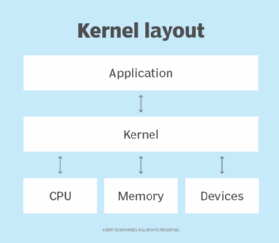
* Linux is free, open-source, and more customizable. It’s mostly used by developers, tech professionals, and servers. You can change almost anything in it — even how it looks and works. It’s also more secure and lightweight, which is great for older computers or learning how computers work.
* Windows, on the other hand, is a paid operating system made by Microsoft. It’s easier for beginners, and most people use it at home or in offices. It supports more commercial software like Microsoft Office and many games, but it’s not as customizable or open as Linux.

**Task- 6**

## **Basic Components of Linux are as follows:**

## 1. Kernel – The Core

### What it does:

* It's the heart of Linux.
* Manages the system's resources: CPU, memory, hardware devices.
* Acts as a bridge between hardware and software.
* 

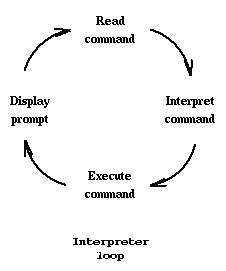
### Functions:

* Process Management: Handles multitasking.
* Memory Management: Allocates and deallocates RAM.
* Device Drivers: Talks to hardware like keyboard, mouse, etc.
* System Calls: Provides an interface for programs to request services.

## 2. Shell – The Command Interpreter

### What it does:

* It’s the interface between the user and the kernel.
* Translates commands typed by the user into actions.



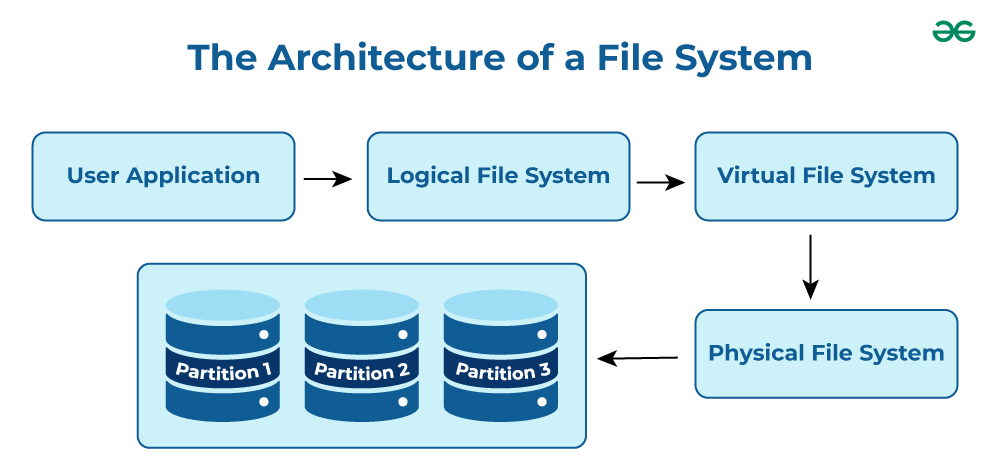
### Types:

* BASH (Bourne Again Shell) – most common.
* Others: ZSH, KSH, CSH

## 3. File System – The Storage Organizer

### What it does:

* Organizes data and files in a hierarchy.
* Linux treats everything as a file, even hardware.



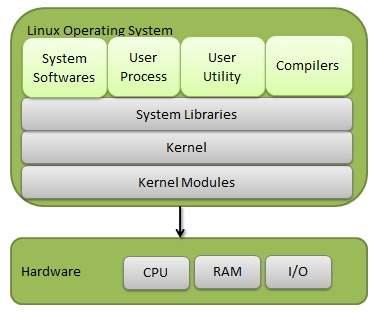
### Common directories:

* /bin – essential binaries
* /home – user files
* /etc – configuration files
* /var – log files
* /dev – device files

## 4. System Libraries – Support Functions

### What it does:

* These are pre-written functions programs use to perform tasks.
* Provide access to system resources without talking to the kernel directly.



### Example:

* The GNU C Library (glibc) is used by many Linux programs.

## 5. System Utilities – Essential Tools

### What it does:

* Tools for system management like file handling, user management, disk management.

### Examples:

* cp, mv, top, ps, chmod, df, etc.

## 6. User Applications – Software Programs

### What it does:

* Programs run by users: text editors, browsers, games, office tools.



### Examples:

* Firefox, LibreOffice, VS Code, GIMP, etc.

**Task- 7**

### Yes it is legal to edit the Linux kernel.

That’s because the Linux kernel is open-source software released under the GNU General Public License (GPL). This license allows you to:

* View the source code
* Modify it
* Use it for personal or professional projects
* Distribute your modified version (as long as you also share your changes under the same license)

**Task- 8**

LILO stands for Linux Loader. It is a boot loader program used in Linux systems. It helps start Linux on your computer.  
It can also manage dual-boot systems, letting you pick between Linux and other operating systems like Windows.

When you turn on your computer, the hardware starts up but doesn’t know how to run the operating system yet. That’s where LILO comes in:

* LILO’s job is to load the Linux operating system into the computer’s memory and start it running.
* It sits in the Master Boot Record (MBR) or a partition’s boot sector.
* When the computer boots, LILO presents you with a menu to choose which OS or kernel to boot if you have multiple installed.
* Once you select, LILO loads the selected Linux kernel into memory and passes control to it.

**Task- 9**

A shell is like a middleman between you and the computer’s operating system. It’s a program that listens to the commands you type and then tells the computer what to do.

Think of it as a translator — you speak commands, and the shell translates those into actions the computer can understand

There are several types of shells, each with its own way of understanding commands and its own features. Here are the most common ones:

* BASH (Bourne Again Shell): The most popular shell used in Linux. It’s powerful and easy to use.
* SH (Bourne Shell): The original shell in Unix. Basic and simple.
* CSH (C Shell): Has a syntax similar to the C programming language.
* KSH (Korn Shell): Combines features of BASH and CSH, good for scripting.
* ZSH (Z Shell): Very customizable and powerful, with lots of extra features.
* TCSH (TENEX C Shell): An improved version of C Shell with more user-friendly features.

**Task- 10**

Swap space is like extra memory for your computer, but instead of being in the fast RAM, it’s a special area on your hard drive or SSD.

When our computer’s RAM gets full because you’re running many programs at once, the system moves some of the data that isn’t being used right now from the RAM to the swap space. This frees up RAM for the programs that need it immediately.

It helps your computer keep running smoothly even when the RAM is full.  
It acts as a backup memory so your system doesn’t crash when there’s not enough RAM.  
However, since hard drives and SSDs are slower than RAM, using swap space can make the system slower if it’s used a lot.

**Task- 11**

Mount means connecting a storage device (like a USB drive, hard disk, or partition) to our Linux system so you can use its files. It’s like plugging in a USB stick and telling the computer, “Show me what’s inside this device in this folder.”

When we mount a device, you attach it to a folder (called a mount point) and then you can access the files inside that folder.

To mount a device:

1. Choose or create a folder where you want to see the files (for example, /mnt/usb).
2. Use the command to attach the device (for example, /dev/sdb1) to that folder:

sudo mount /dev/sdb1 /mnt/usb

After this, you can open /mnt/usb and access the files on the device.

To Unmount a File System:

When you’re done, you should safely disconnect the device by unmounting it. This tells the system to stop using the device, so it’s safe to remove:

sudo umount /mnt/usb

### In short:

* Mounting means making a device’s files available by linking it to a folder.
* Unmounting means safely disconnecting the device so no data is lost.

**Task- 12**

chmod stands for "change mode". It is used in Linux to change the permissions of a file or folder.

Permissions control who can read, write, or run a file. With chmod, you can decide:

* Who can view the file (read)
* Who can edit it (write)
* Who can execute/run it (execute)

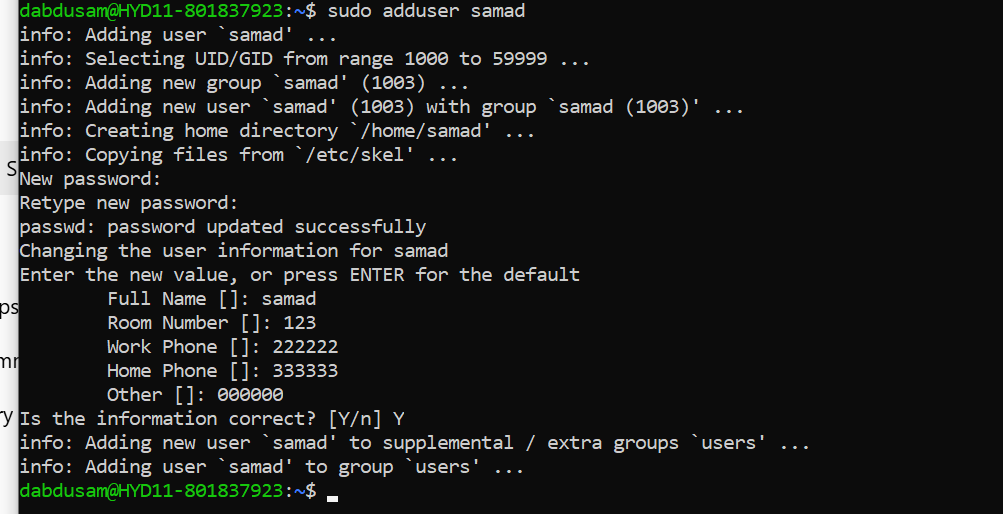
How to use it:

#### Symbolic Method (easy to read)

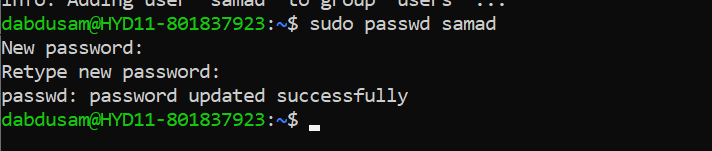
chmod u+x filename

This means: give the owner permission to execute the file.

**Task- 13:**



**Task- 14**:



**Task- 15**:

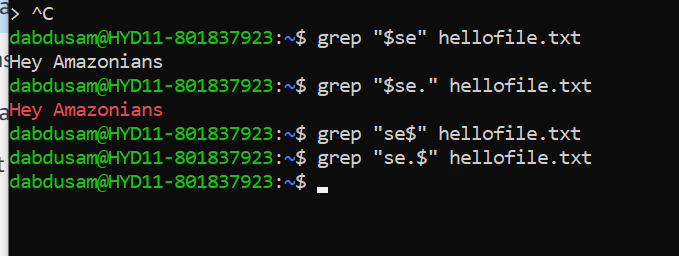
A process is like a complete program that is running on your computer.

* It has its own memory and resources.
* One process can’t easily access another process’s data.
* Example: When you open Google Chrome and Spotify, each one runs as a separate process.

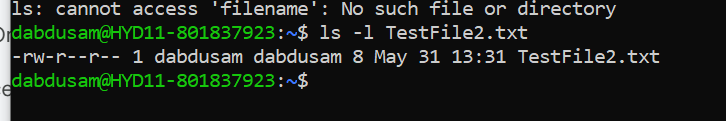
A thread is like a small task running inside a process.

* Threads share the same memory and resources of the process they belong to.
* A process can have multiple threads working at the same time.
* Example: In Chrome, one thread may handle loading the page, another handles user input, and another plays video.

**Task-16**



**Task- 18**

****

**Task- 19**

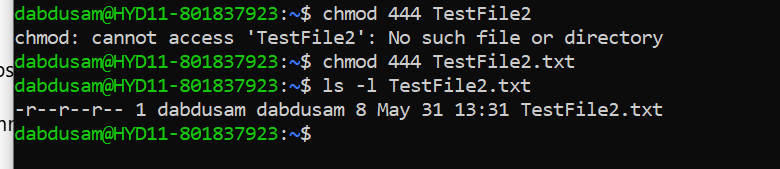
Default base permission for files = 666 (rw-rw-rw-)  
Default umask (usually 022) subtracts write (w) from group and others

Owner → rw- → Read & Write

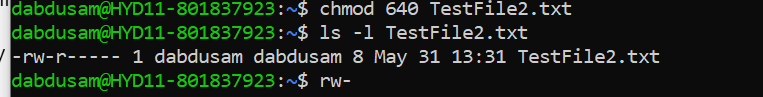
Group → r-- → Read only

Others → r-- → Read only

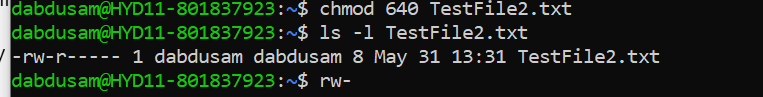
**Task- 20**

****

**Task- 21**

****

**Task- 22**

****

**Task- 26**

A process is just a program that is running on your computer.

When you open or run anything — like a text editor, a browser, or even a command in the terminal — Linux starts a process for it.

Each process:

* Has its own ID (called PID)
* Uses some memory and CPU
* Runs independently from other processes

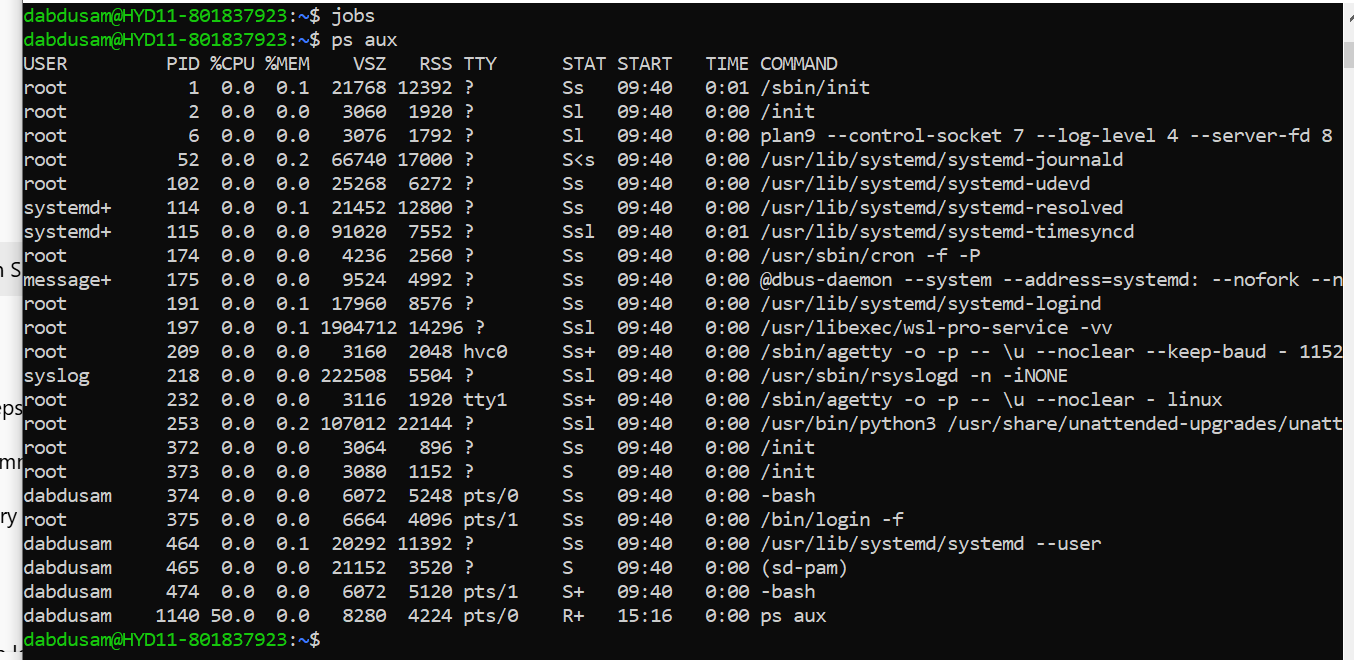
**Task- 27**

A foreground process is running directly in your terminal — it blocks input until it finishes.

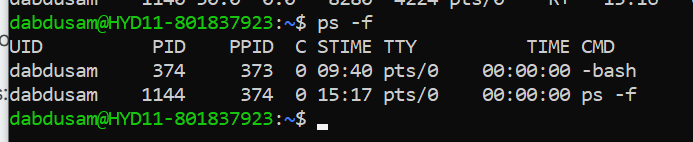
./myscript.sh & # This runs in background

A background process is started with & at the end

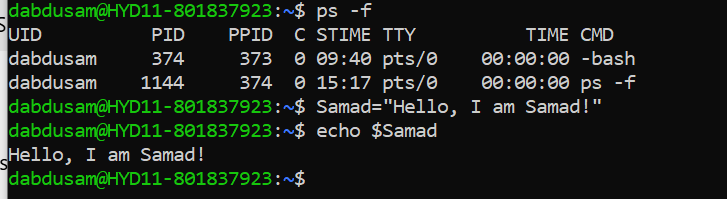
**Task- 28**

****

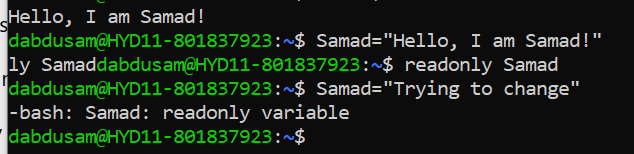
**Task- 29**

****

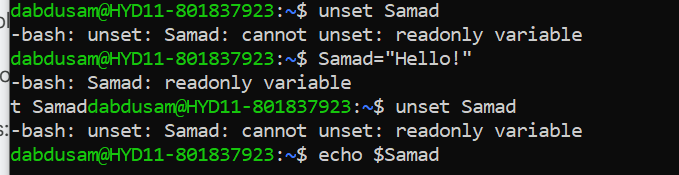
**Task- 30**

****

**Task- 31**

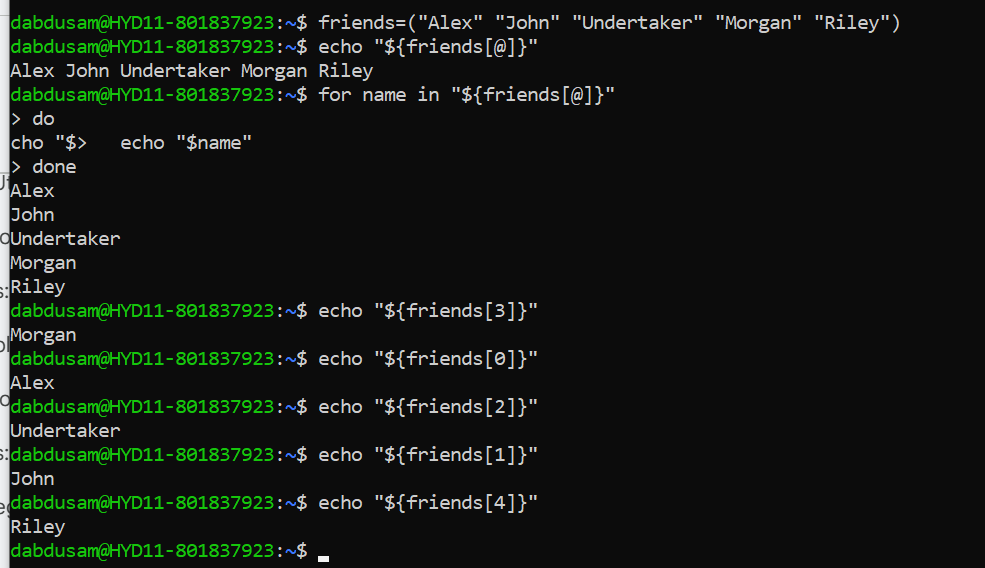
****

**Task- 32**

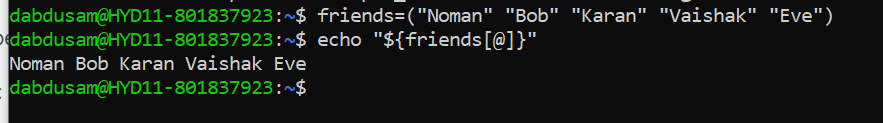
****

# This will print nothing because it's unset

**Task- 33**

****

**Task- 34**

****

**Task- 35**

The Output is:

0

1 0

2 1 0

3 2 1 0

4 3 2 1 0

5 4 3 2 1 0

6 5 4 3 2 1 0

7 6 5 4 3 2 1 0

8 7 6 5 4 3 2 1 0

9 8 7 6 5 4 3 2 1 0