

**CSE/IT**  
**2025** Batch

**3<sup>rd</sup>**  
**Semester**

# **Operating System**

**Polytechnic**

**ONE SHOT**



**Semester**





## Syllabus

1. Overview of Operating System.

2. Process Management.

3. Memory Management.

4. I/O System.

5. File management.



## Chapter No 1:

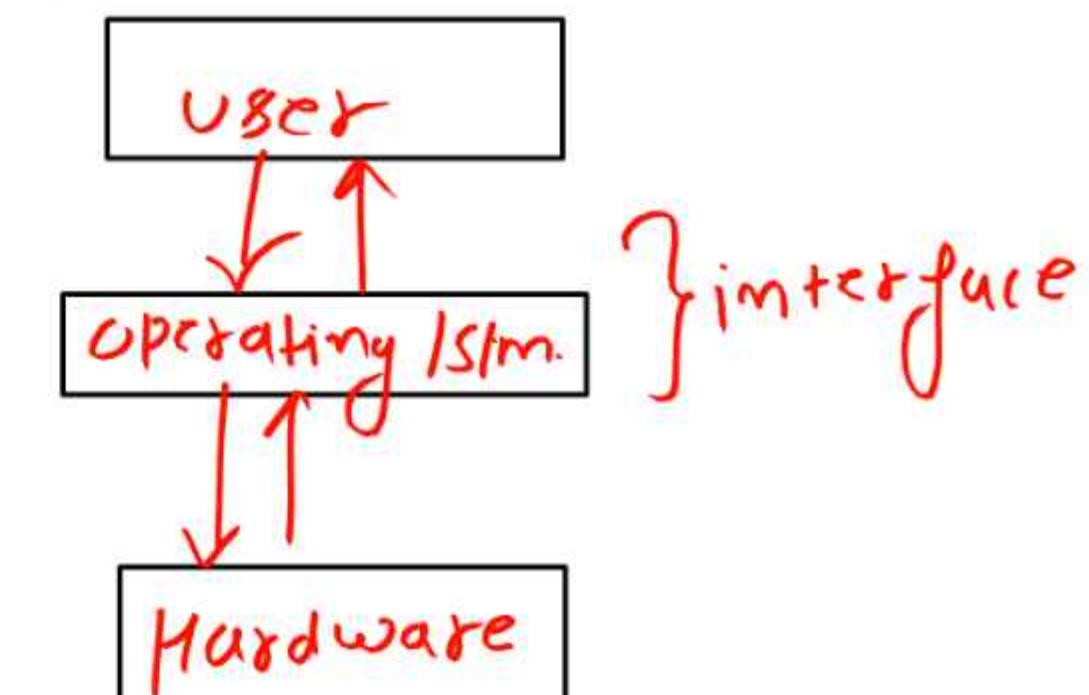
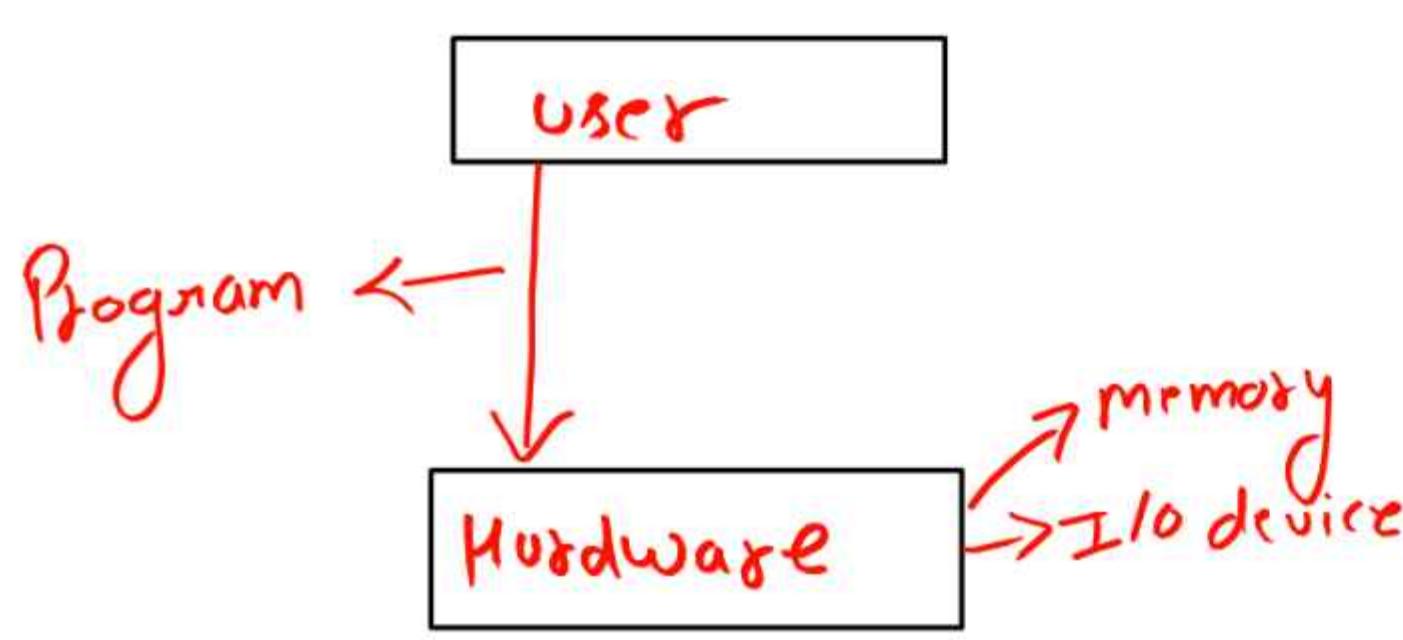
## Chapter No :1 Overview of Operating System.

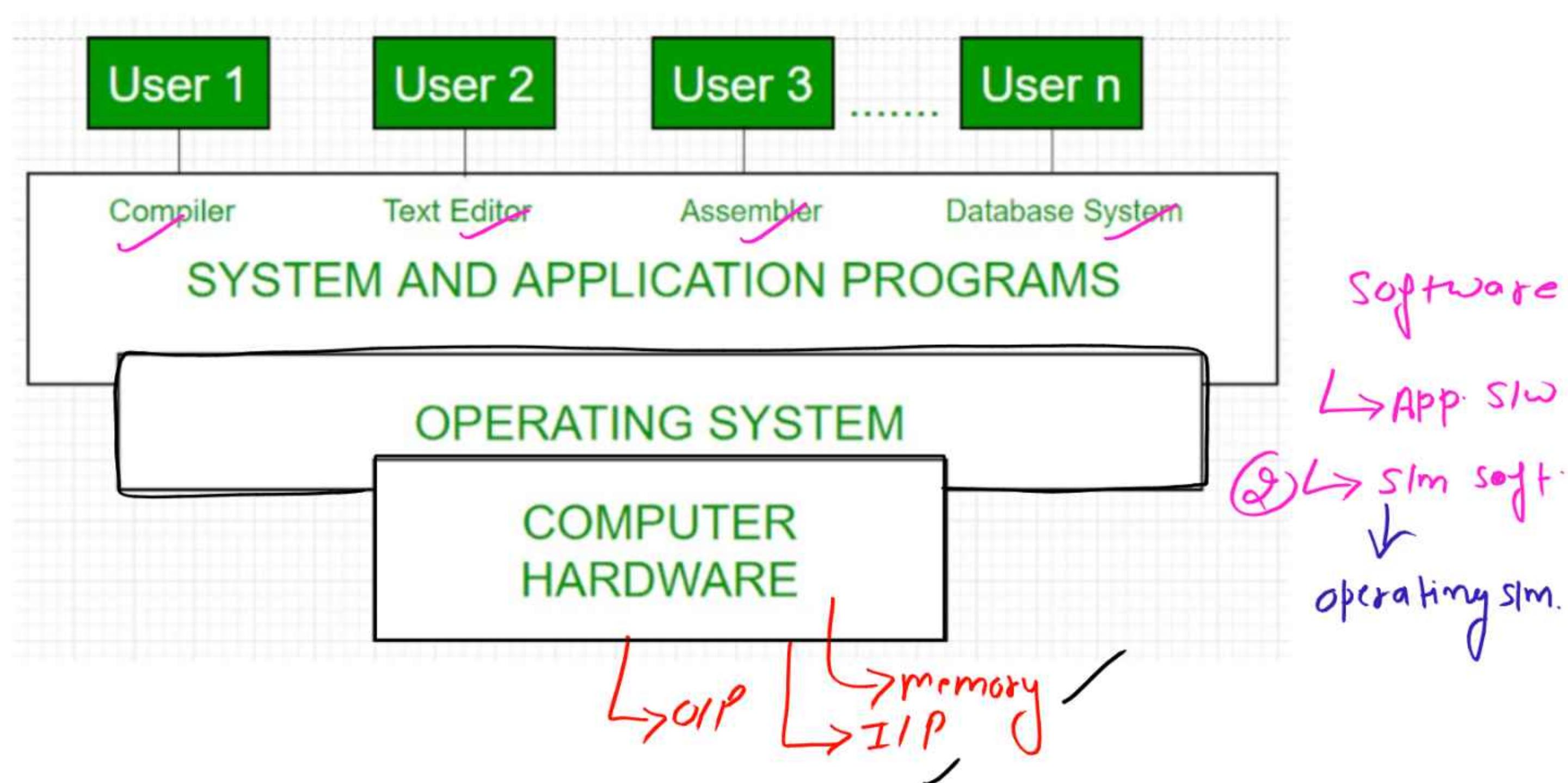
- Introduction
- Functions & Services of OS
- Different types of OS
- UNIX/LINUX Architecture
- Kernel
- Services and systems calls
- System programs

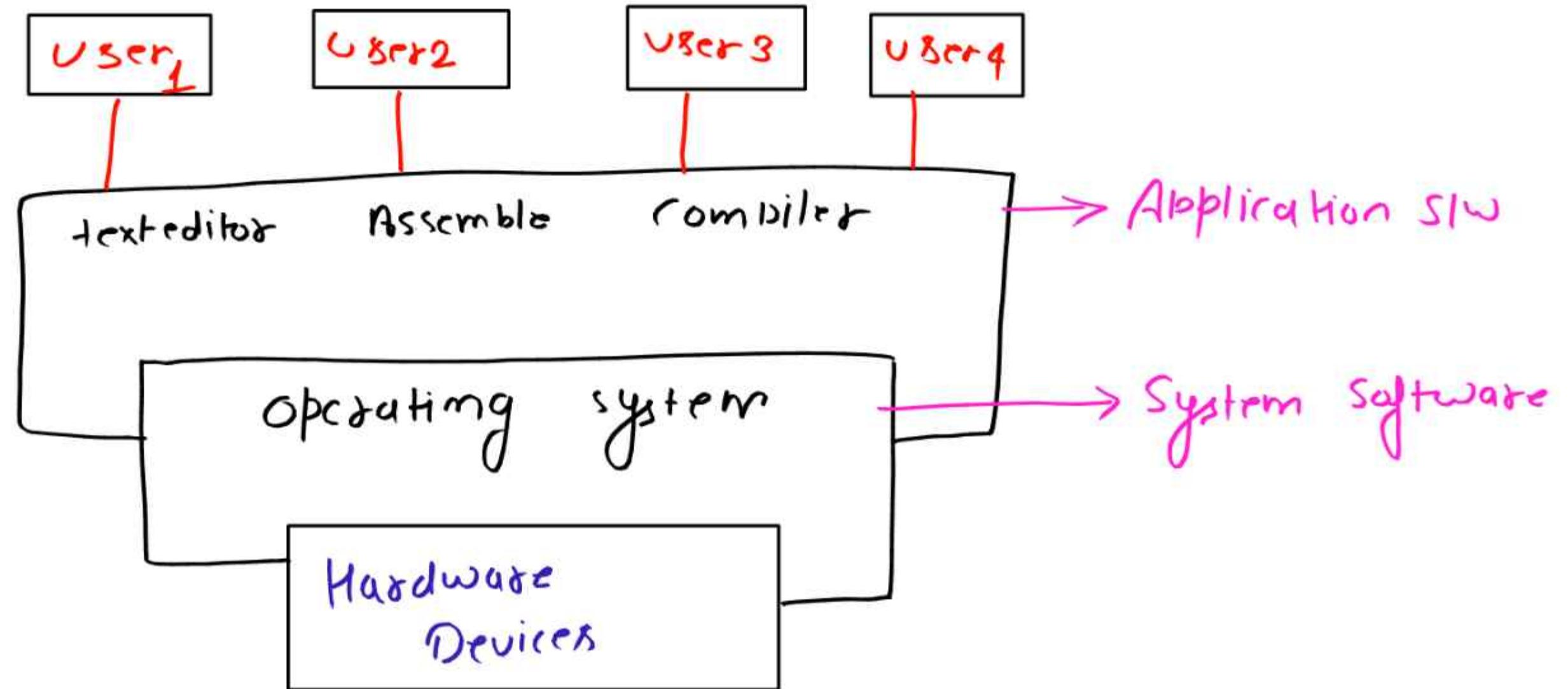


## Operating System:

- ऑपरेटिंग सिस्टम (Operating System) एक ऐसा सॉफ्टवेयर होता है जो कंप्यूटर और उपयोगकर्ता (User) के बीच मध्यस्थ (Mediator) का काम करता है। यह कंप्यूटर के सभी हार्डवेयर और सॉफ्टवेयर संसाधनों को नियंत्रित (Manage) करता है।
- Operating System (Operating System) is a software that acts as an intermediary (mediator) between the computer and the user (user). It controls (manages) all the documents and software of the computer.









Window



Linux



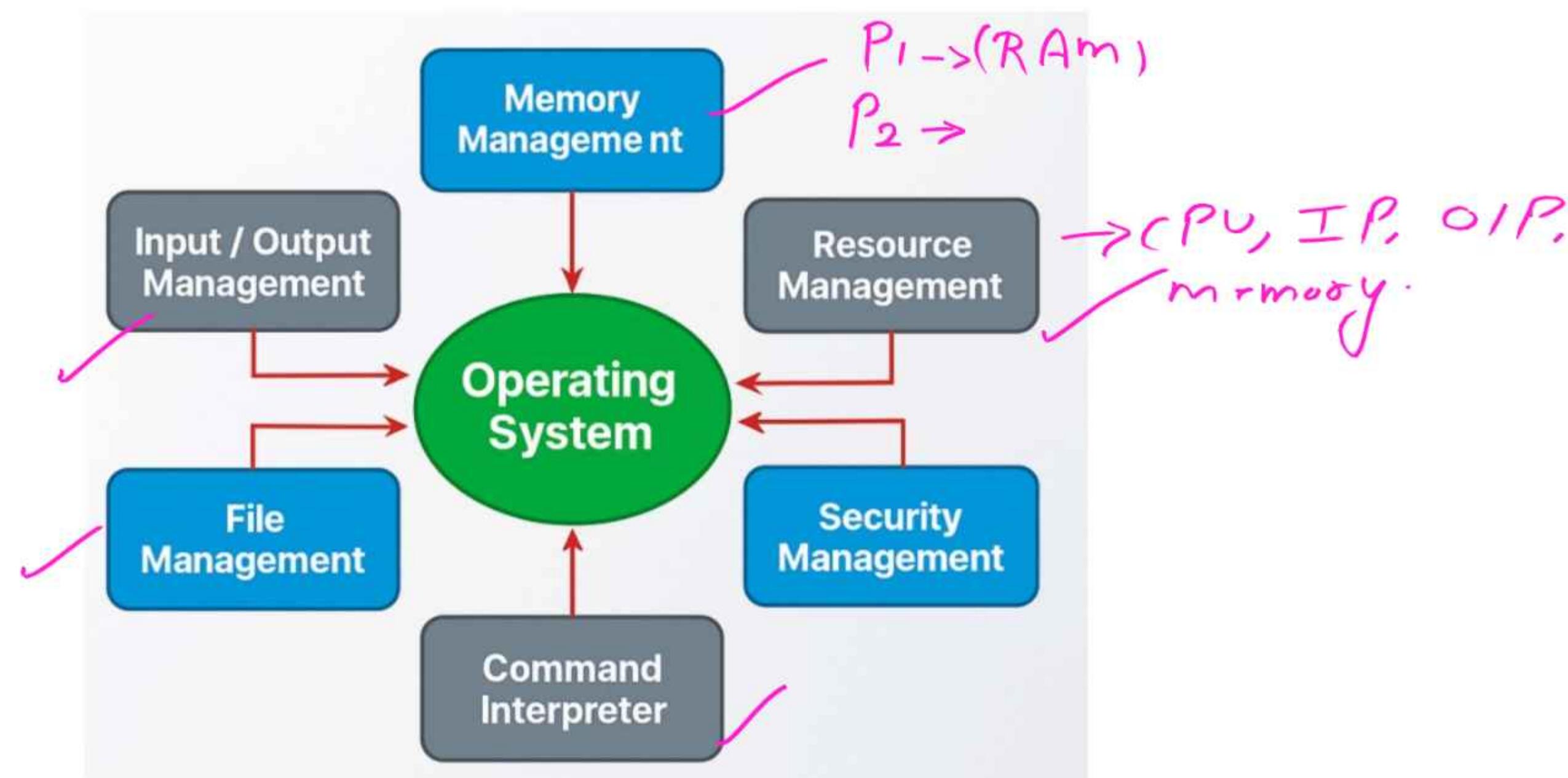
MAC



Android



## Function of Operating System (ऑपरेटिंग सिस्टम का कार्य):





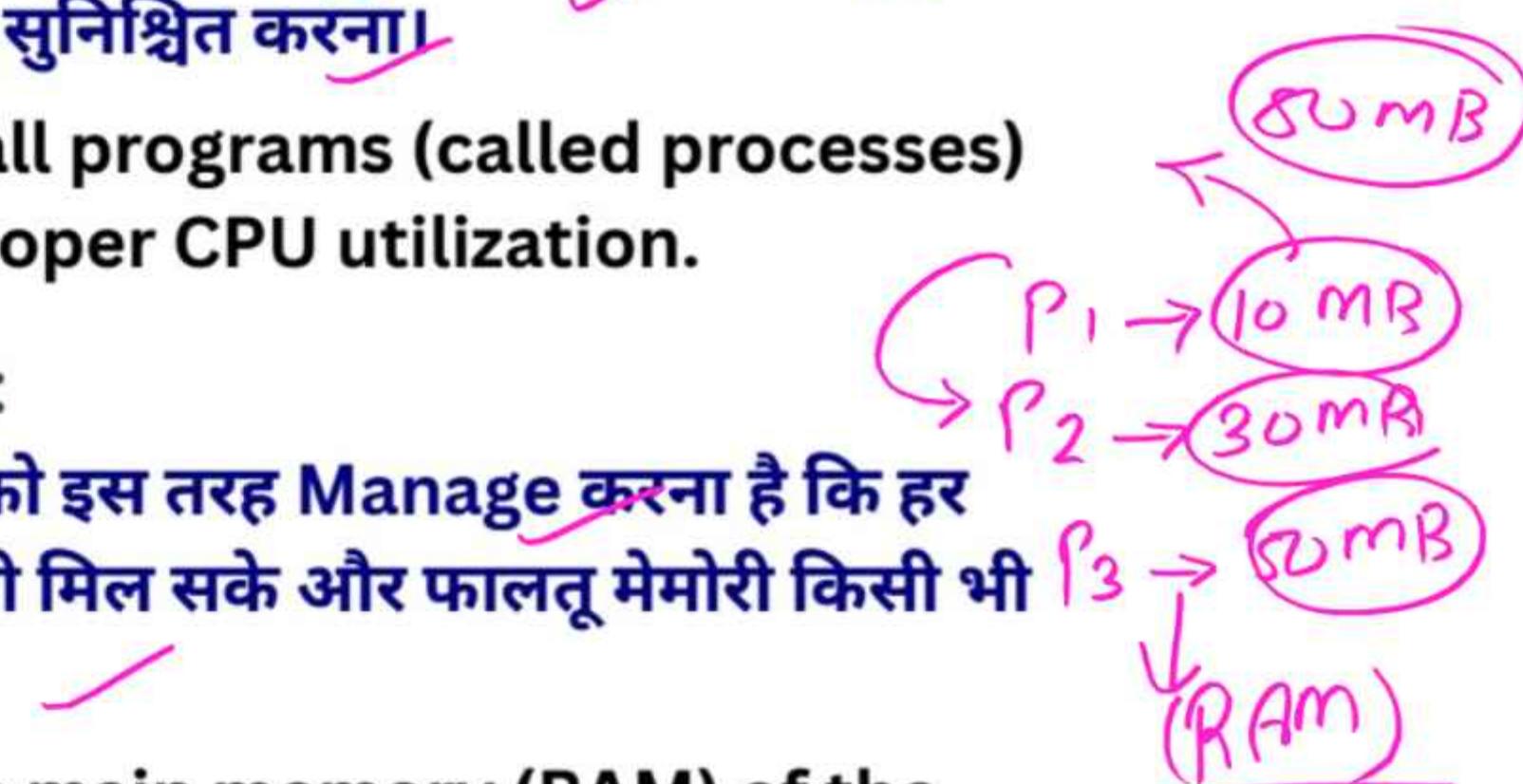
## Function of Operating System (ऑपरेटिंग सिस्टम का कार्य):

### 1. Process Management (प्रोसेस प्रबंधन):

- सिस्टम में चल रहे सभी Programs (जिन्हें हम Process कहते हैं) को शुरू करना, रोकना, एक से दूसरे में स्विच करना, और CPU का सही उपयोग सुनिश्चित करना।
- Starting, stopping, switching between all programs (called processes) running on the system, and ensuring proper CPU utilization.

### 2. Memory Management (मेमोरी का प्रबंधन):

- OS का यह कार्य सिस्टम की मुख्य मेमोरी (RAM) को इस तरह Manage करना है कि हर Running Program को उसकी जरूरत की मेमोरी मिल सके और फालतू मेमोरी किसी भी प्रोग्राम को न दी जाए।
- The function of the OS is to manage the main memory (RAM) of the system in such a way that every running program gets the memory it needs and extra memory is not given to any program.





### ◆ 3. File Management (फाइल प्रबंधन):

- OS कंप्यूटर में फाइलों और फोल्डरों को व्यवस्थित रखने का कार्य करता है। यह फाइल को बनाने, पढ़ने, संशोधित करने, हटाने और Access Control प्रदान करने का काम करता है।
- OS works to keep the files and folders organized in the computer. It works to create, read, modify, delete and provide access control to the file.

### ◆ 4. Device Management (डिवाइस प्रबंधन):

- OS इनपुट और आउटपुट डिवाइस जैसे Keyboard, Mouse, Printer, Hard Disk आदि को नियंत्रित और मैनेज करता है।
- OS controls and manages input and output devices such as keyboard, mouse, printer, hard disk etc.



#### ◆ 5. User Interface (यूजर इंटरफेस प्रदान करना):

- OS यूजर को कंप्यूटर के साथ इंटरैक्ट करने के लिए इंटरफेस प्रदान करता है, जिससे यूजर सिस्टम को आसानी से उपयोग कर सके। यह CLI (Command Line Interface) या GUI (Graphical User Interface) हो सकता है।
- OS provides the user with an interface to interact with the computer, so that the user can use the system easily. It can be CLI (Command Line Interface) or GUI (Graphical User Interface).

#### ◆ 6. Security and Access Control (सुरक्षा एवं पहुँच नियंत्रण):

- Operating System सिस्टम और उसमें store डेटा को Unauthorised Access से बचाता है। यह User Authentication, Passwords, File Permissions आदि तकनीकों का उपयोग करता है।
- Operating System protects the system and the data stored in it from unauthorized access. It uses techniques like User Authentication, Passwords, File Permissions etc.

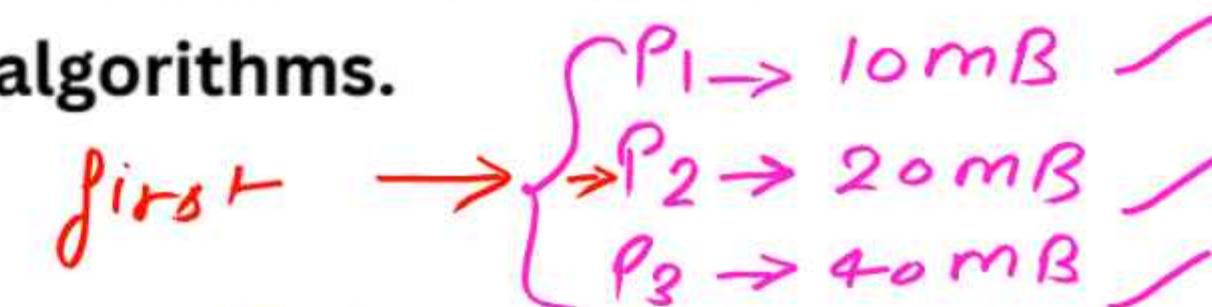


#### ◆ 7. Job Scheduling (कार्यक्रम निर्धारण / Scheduling):

- Operating System यह तय करता है कि कौन-सा प्रोग्राम कब CPU का उपयोग करेगा और कितनी देर तक करेगा। यह काम Scheduling Algorithms के जरिए किया जाता है।
- The operating system decides which program will use the CPU and for how long. This work is done through scheduling algorithms.

#### 8. Communication Management (प्रोसेस संचार):

- अगर दो या दो से ज्यादा प्रोसेस आपस में डेटा का आदान-प्रदान करना चाहें, तो OS उन्हें एक माध्यम (Mechanism) देता है, जैसे Pipes, Message Passing, Shared Memory आदि।
- If two or more processes want to exchange data among themselves, then the OS provides them a mechanism, like Pipes, Message Passing, Shared Memory etc.



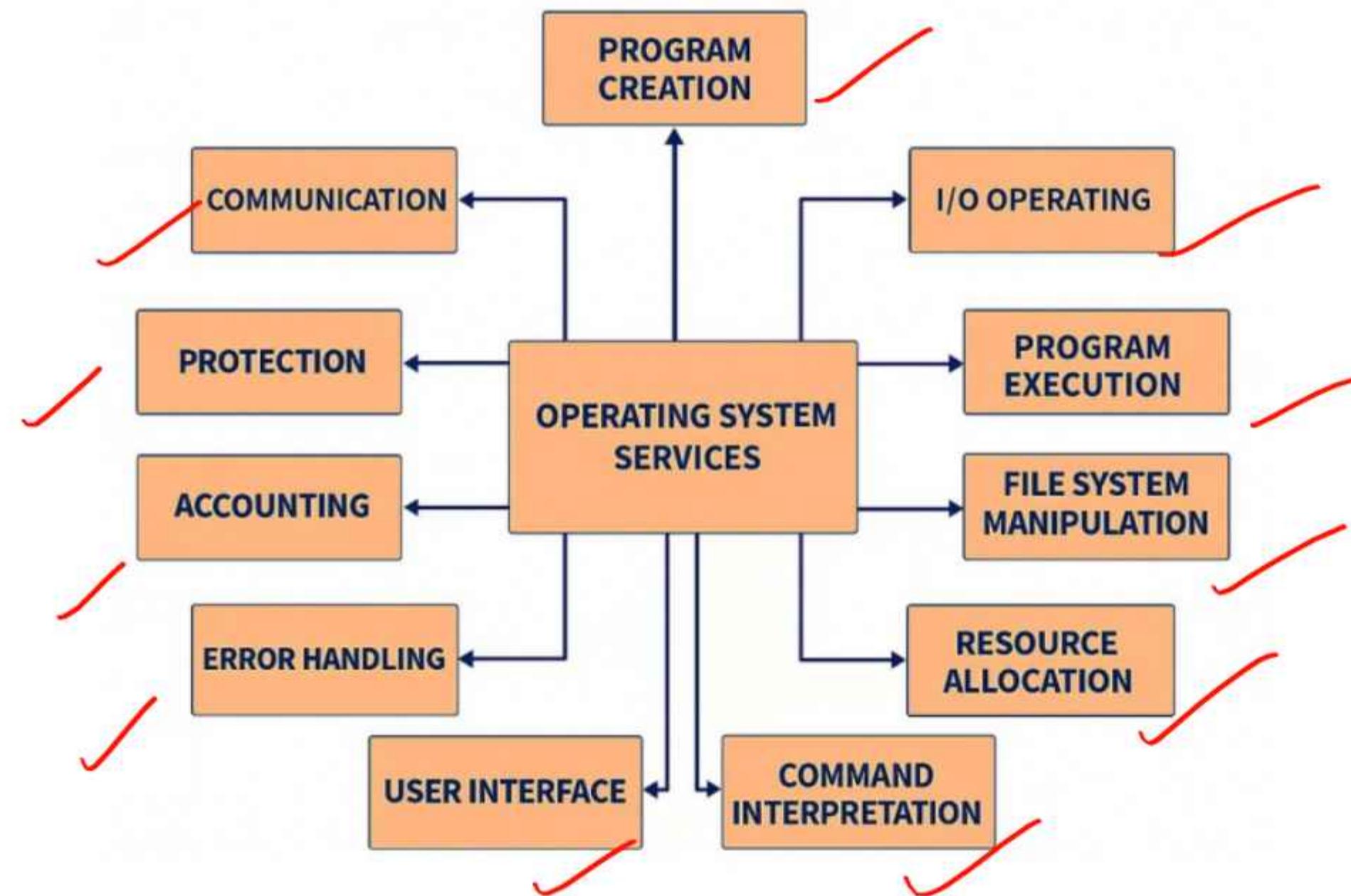


## Services of Operating System (ऑपरेटिंग सिस्टम की सेवाएँ):

- सर्विस (Service) का मतलब है – ऑपरेटिंग सिस्टम वह सुविधाएँ (Features) और कार्यप्रणालियाँ (Functionalities) जो वह यूज़र या प्रोग्राम को प्रदान करता है, ताकि वह आसानी से सिस्टम का उपयोग कर सके।
- Service means - the features and functionalities that the operating system provides to the user or program so that he can use the system easily.



## → Services of Operating System (ऑपरेटिंग सिस्टम की सेवाएँ):





## → Services of Operating System (ऑपरेटिंग सिस्टम की सेवाएँ):

### 1. Program Execution (प्रोग्राम चलाना):

- OS यूजर द्वारा दिए गए किसी भी प्रोग्राम को Load करता है, उसे रन करने के लिए CPU को सौंपता है और उसे Execute कराता है।
- The OS loads any program given by the user, assigns it to the CPU for running and executes it.



### 2. I/O Operations (इनपुट/आउटपुट संचालन)

- OS Input और Output Devices (जैसे कीबोर्ड, माउस, प्रिंटर) के जरिए डेटा को लेने और भेजने की सुविधा देता है। यह Input/Output Commands को संभालता है।
- OS allows data to be received and sent through input and output devices (such as keyboard, mouse, printer). It handles input/output commands.



### ◆ 3. File System Manipulation (फाइल सिस्टम का प्रबंधन):

- OS यूज़र को फाइलें बनाने, खोलने, पढ़ने, लिखने, हटाने और व्यवस्थित करने की सेवाएँ देता है।
- The OS provides services to the user to create, open, read, write, delete and organize files.

### ◆ 4. Communication Services (प्रोसेस कम्युनिकेशन सेवा):

- अगर दो या दो से ज्यादा प्रोग्राम या प्रोसेस को आपस में डेटा शेयर करना है, तो OS उन्हें एक Communication Mechanism (जैसे Message Passing, Shared Memory) प्रदान करता है।



- If two or more programs or processes need to share data among themselves, then the OS provides them with a communication mechanism (eg. message passing, shared memory).



#### ◆ 5. Error Detection and Handling (त्रुटि पहचान और सुधार):

- OS यह सेवा देता है कि सिस्टम में कोई Error (त्रुटि) या Unexpected Behavior हो तो उसे Detect किया जाए और सही तरीके से हैंडल किया जाए।
- OS provides this service so that if there is any error or unexpected behavior in the system, it is detected and handled properly.

#### ◆ 6. Resource Allocation (रिसोर्स का आवंटन):

- Operating System सभी रिसोर्स (CPU, RAM, Disk, Devices) को इस तरह बाँटता है कि सभी प्रोग्राम और प्रोसेस को उनका हिस्सा मिल सके।
- The Operating System distributes all the resources (CPU, RAM, Disk, Devices) in such a way that all programs and processes get their share.

✓  $P_1 \rightarrow (10\text{mB})$ , Keyboard, Printer  
✓  $P_2 \rightarrow (30\text{mB})$ , Printer  
 $P_3 \rightarrow (50\text{mB})$ , Mouse, Printer



#### ◆ 7. Security and Protection (सुरक्षा और संरक्षण सेवा):

- OS यह सेवा प्रदान करता है जिससे unauthorized users सिस्टम या फाइलों तक पहुँच न पा सकें। इसके लिए पासवर्ड, Permission, Firewall जैसी तकनीकें इस्तेमाल होती हैं।
- OS provides this service so that unauthorized users cannot access the system or files. For this, techniques like password, permission, firewall are used.

#### ◆ 8. Accounting (लेखा जोखा रखना):

- OS यह सेवा देता है कि सिस्टम कौन-कौन यूज़र कितनी देर चला रहा है, कौन-से रिसोर्स कितनी बार इस्तेमाल हुए – उसका पूरा रिकॉर्ड रखा जाता है।
- OS provides the service of keeping a complete record of which user is using the system for how long, which resources were used how many times.



## Objectives of Operating System (ऑपरेटिंग सिस्टम के उद्देश्य):

### 1. उपयोग में आसान(Easy to Use):

- ऑपरेटिंग सिस्टम कंप्यूटर को सभी के लिए उपयोग में सरल और सुविधाजनक बनाता है।
- The OS makes computers simple and convenient for everyone to use.

### 2. उपयोगकर्ता-अनुकूल (User-Friendly):

- यह उपयोगकर्ताओं के लिए एक इंटरैक्टिव और समझने में आसान इंटरफ़ेस प्रदान करता है।
- It provides an interactive and easy-to-understand interface for users.

### 3. संसाधनों तक आसान पहुंच (Easy Access to Resources):

- यह उपयोगकर्ताओं को बिना किसी कठिनाई के कंप्यूटर के हार्डवेयर (जैसे मेमोरी, स्टोरेज और प्रोसेसर) तक पहुंचने में मदद करता है।
- It helps users access the computer's hardware (like memory, storage, and processor) without difficulty.



#### 4. संसाधनों का प्रबंधन(Manages Resources):

- यह मेमोरी, सीपीयू और स्टोरेज जैसे कंप्यूटर संसाधनों का कुशल उपयोग सुनिश्चित करता है।
- It ensures the efficient use of computer resources like memory, CPU, and storage.

#### 5. नियंत्रण और मॉनिटर (Controls and Monitors):

- ओएस यह ट्रैक करता है कि कौन सा प्रोग्राम या उपयोगकर्ता किस संसाधन का उपयोग कर रहा है और टकराव को रोकता है।
- The OS tracks which program or user is using which resource and prevents conflicts.



## ऑपरेटिंग सिस्टम के प्रकार (Types of Operating System):

- ऑपरेटिंग सिस्टम के कई प्रकार होते हैं और हर प्रकार का उपयोग अलग-अलग कामों के लिए किया जाता है।
- There are many types of operating systems and each type is used for different purposes.



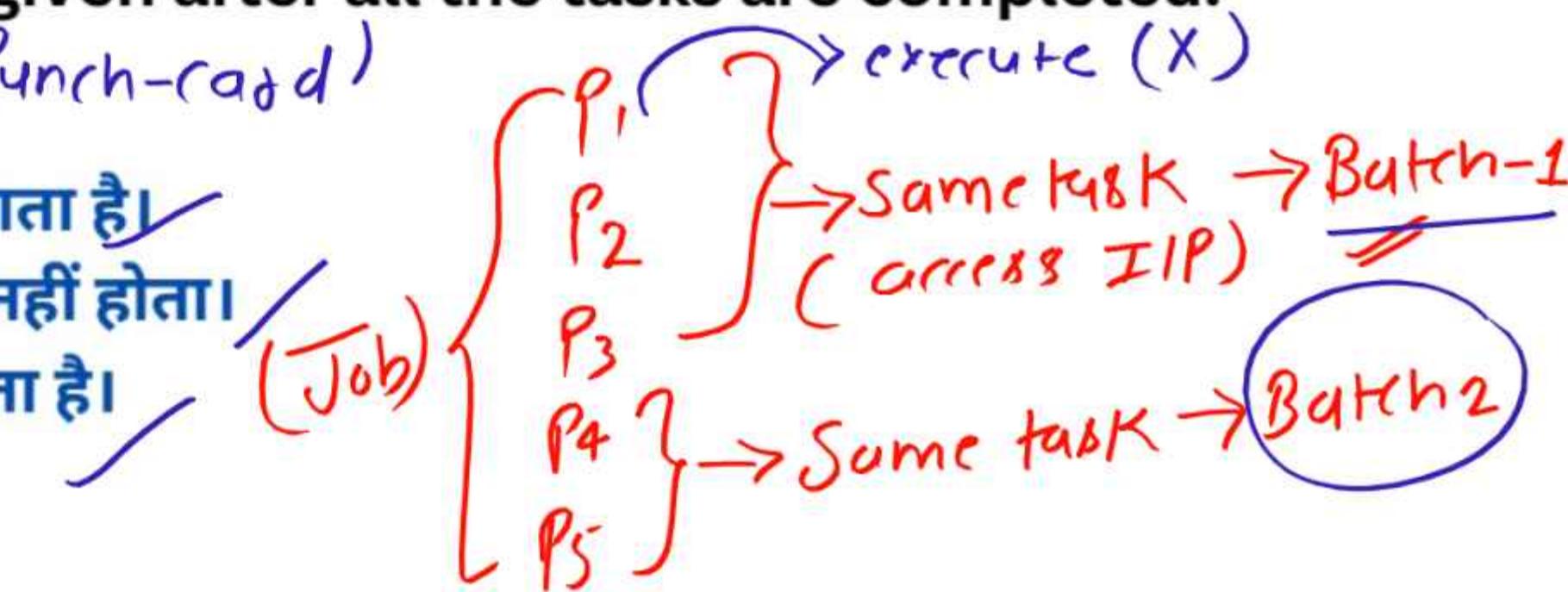
## 1. बैच ऑपरेटिंग सिस्टम (Batch Operating System):

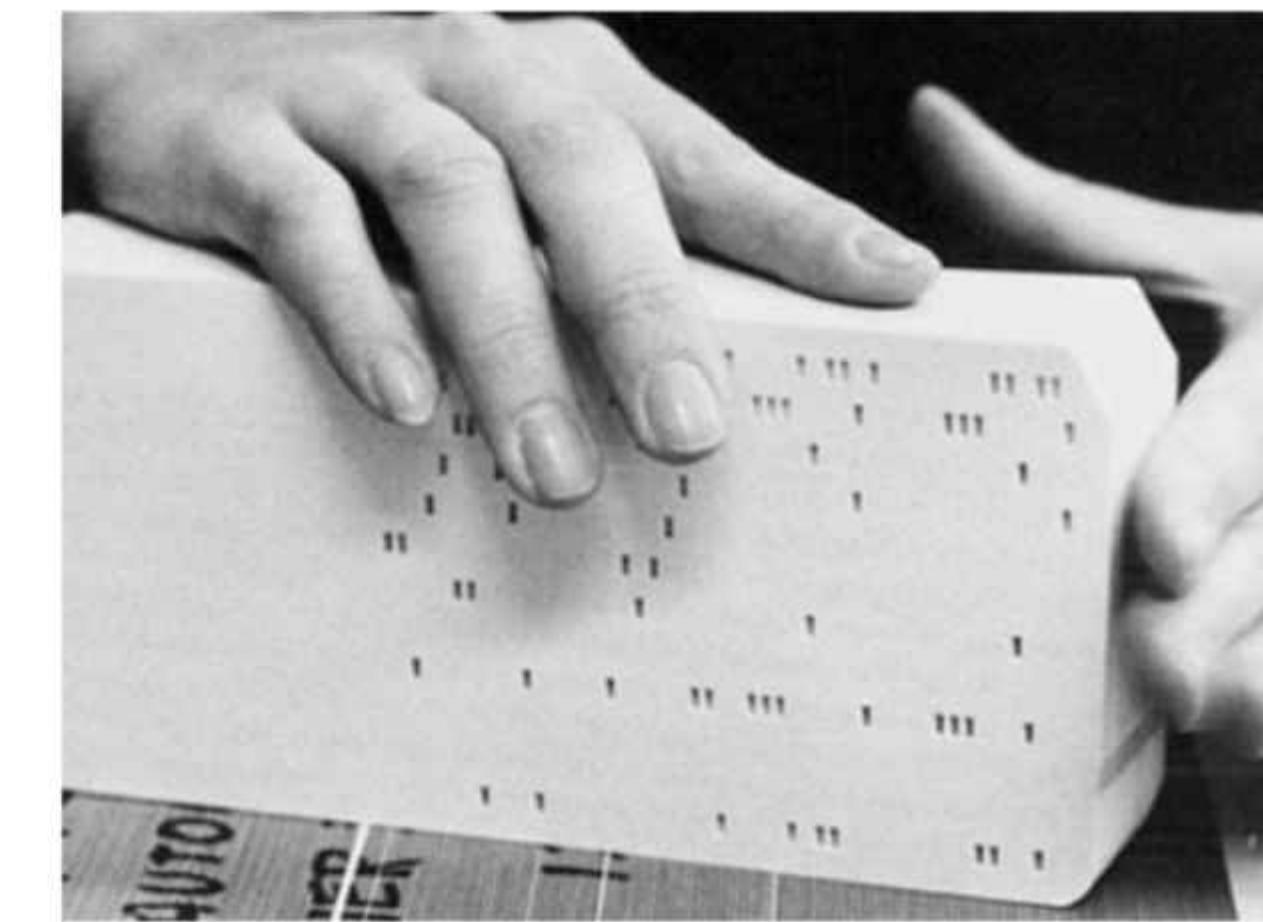
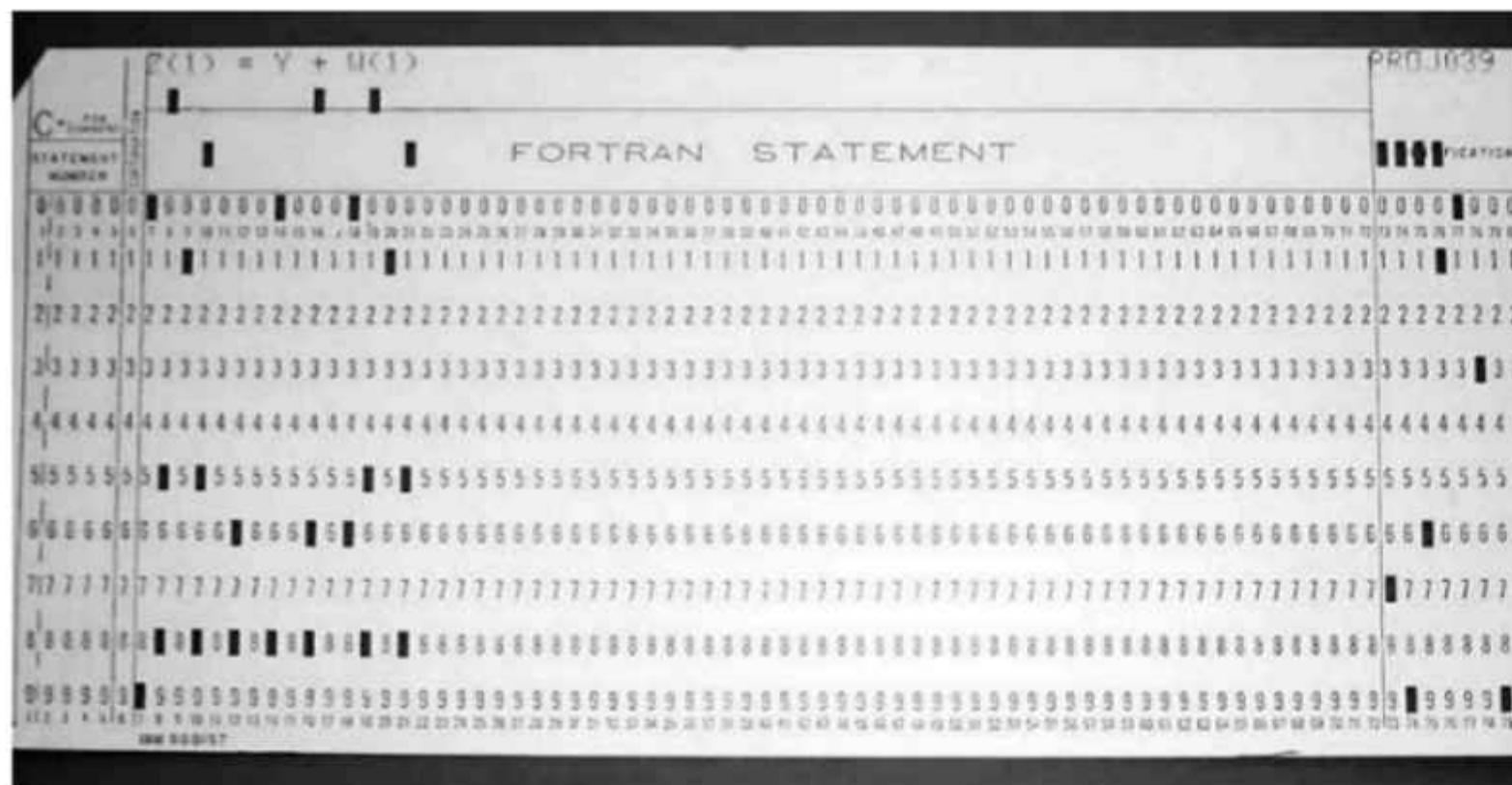
- इसमें समान प्रकार के कार्यों (जैसे डेटा प्रोसेसिंग, प्रिंटिंग) को एक साथ ग्रुप बनाकर प्रोसेस किया जाता है। इसमें user को तुरंत आउटपुट नहीं मिलता, बल्कि सभी कार्य पूरे होने के बाद रिजल्ट दिया जाता है।
- In this, similar tasks (such as data processing, printing) are grouped together and processed. In this, the user does not get the output immediately, but the result is given after all the tasks are completed.

विशेषताएँ:

- एक जैसे कार्यों को बैच (समूह) में किया जाता है।
- उपयोगकर्ता और कंप्यूटर का सीधा संपर्क नहीं होता।
- यह बड़े डाटा प्रोसेसिंग के लिए उपयोगी होता है।

(IIP, OP → Punch-card)

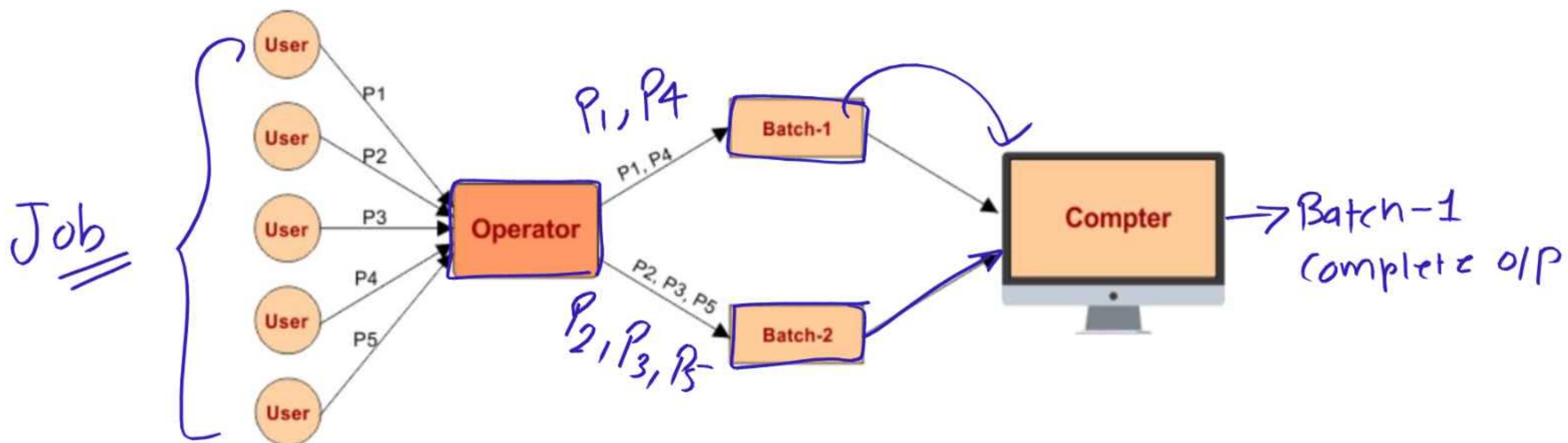






### Features:

- Similar tasks are performed in batches.
- There is no direct contact between the user and the computer.
- It is useful for large data processing.





## Advantages of Batch Operating System (फायदे):

### Automatic Execution(स्वचालित तरीके से प्रोग्राम चलना):

- Jobs को manually चलाने की ज़रूरत नहीं होती, सब कुछ अपने आप होता है।
- Jobs do not need to be executed manually, everything happens automatically.

### No User Interaction Needed(प्रोग्राम के दौरान यूज़र की ज़रूरत नहीं पड़ती):

- Jobs को एक बार देने के बाद user को कुछ करने की ज़रूरत नहीं होती।
- Once the jobs are given the user does not need to do anything.

### Time-Saving for Users(यूज़र का समय बचता है):

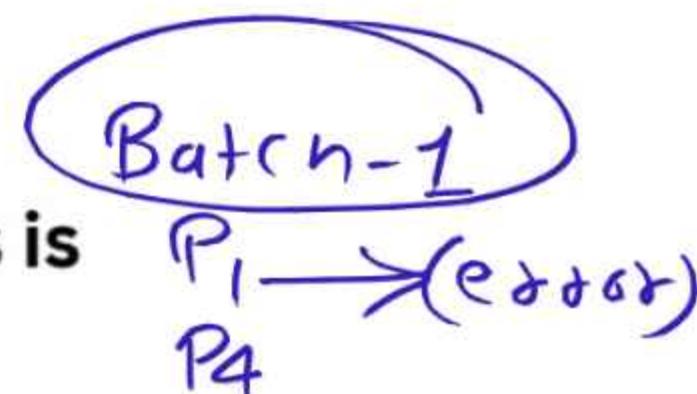
- यूज़र jobs देकर दूसरा काम कर सकता है, सिस्टम अपना काम करता रहता है।
- The user can do other work by giving jobs, the system continues doing its work.



## ~~X~~ Disadvantages of Batch Operating System (नुकसान):

### Error Detection is Delayed (गलतियाँ समय पर पकड़ में नहीं आतीं)

- Error तब ही पता चलता है जब पूरा batch process हो चुका होता है।
- The error becomes known only when the entire batch process is completed.



### Job Waiting Time Can Be Long (नई job को लंबा इंतज़ार करना पड़ सकता है):

- पहले से queued jobs के कारण नई jobs को delay हो सकता है।
- New jobs may be delayed due to already queued jobs.

P<sub>1</sub> → 10 m/s  
P<sub>4</sub> → 2 m/s



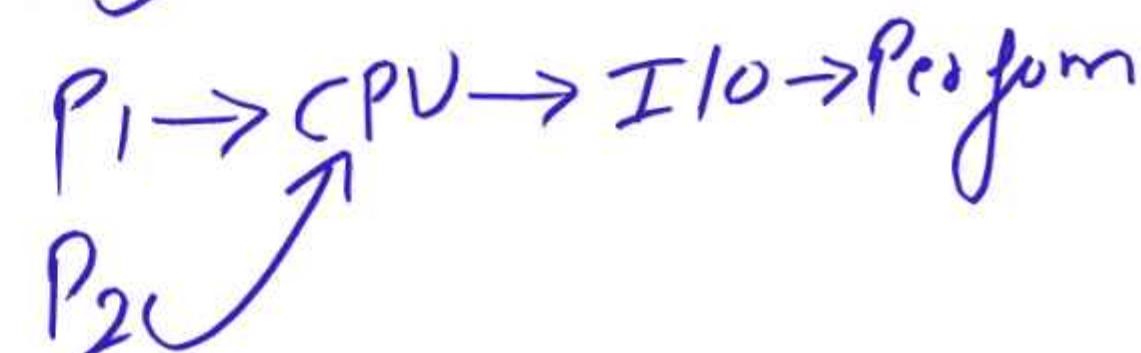
## 2. मल्टीप्रोग्रामिंग ऑपरेटिंग सिस्टम

### (Multiprogramming Operating System):

- इसमें एक समय में कई प्रोग्राम एक साथ रन कर सकते हैं। जब एक प्रोग्राम CPU का उपयोग नहीं कर रहा होता, तब दूसरा प्रोग्राम CPU का उपयोग कर सकता है।
- In this, many programs can run simultaneously at a time. When one program is not using the CPU, then another program can use the CPU.

#### विशेषताएँ:

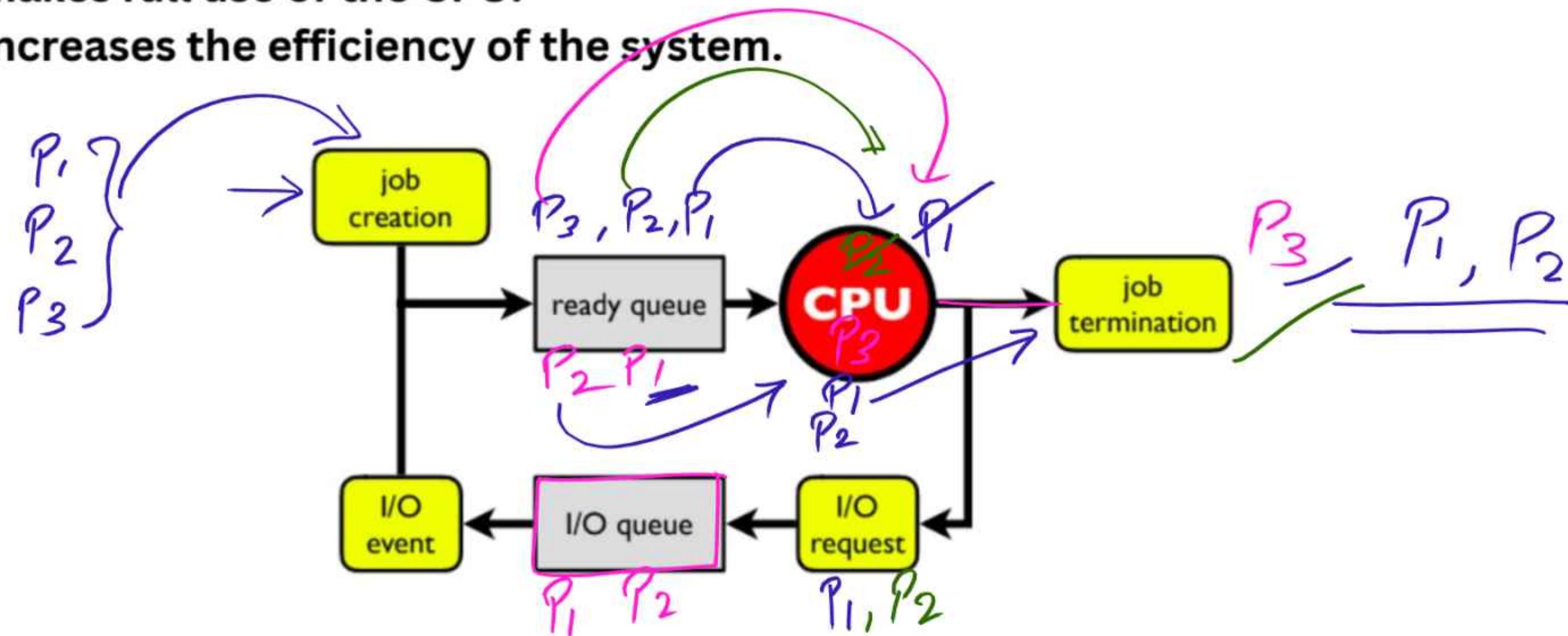
- कंप्यूटर की मेमोरी में एक समय में कई प्रोग्राम लोड हो सकते हैं।
- यह CPU का पूरा उपयोग करता है।
- इससे सिस्टम की कार्यक्षमता बढ़ती है।





### Features:

- Many programs can be loaded in the computer's memory at one time.
- It makes full use of the CPU.
- It increases the efficiency of the system.





### Advantages of Multiprogramming Operating System (फायदे):

#### Maximum CPU Utilization (CPU का पूरा उपयोग होता है)

- जब एक प्रोग्राम I/O का इंतजार करता है, दूसरा प्रोग्राम CPU का उपयोग करता है। इससे CPU idle नहीं रहता।
- While one program waits for I/O, another program uses the CPU. This keeps the CPU from remaining idle.

#### Reduced Waiting Time (प्रोग्राम्स को कम इंतज़ार करना पड़ता है):

- Jobs एक के बाद एक execute होती हैं, जिससे सभी tasks जल्दी पूरे होते हैं।
- Jobs are executed one after the other, so that all tasks are completed quickly.

#### Supports Multi-user Environment (कई यूज़र्स एक साथ काम कर सकते हैं):

- एक ही समय पर कई users अपनी jobs system में चला सकते हैं।
- Multiple users can run their jobs on the system at the same time.



## Disadvantage of Multiprogramming Operating System (नुकसान):

- ज्यादा load होने पर system crash या hang हो सकता है। ✓
- The system may crash or hang if there is excessive load.
- कई programs एक साथ चलने पर errors trace करना मुश्किल होता है। ✓
- It is difficult to trace errors when many programs are running simultaneously.
- अगर proper protection नहीं हो तो programs एक-दूसरे को interfere कर सकते हैं। ✓
- If there is no proper protection, programs can interfere with each other.

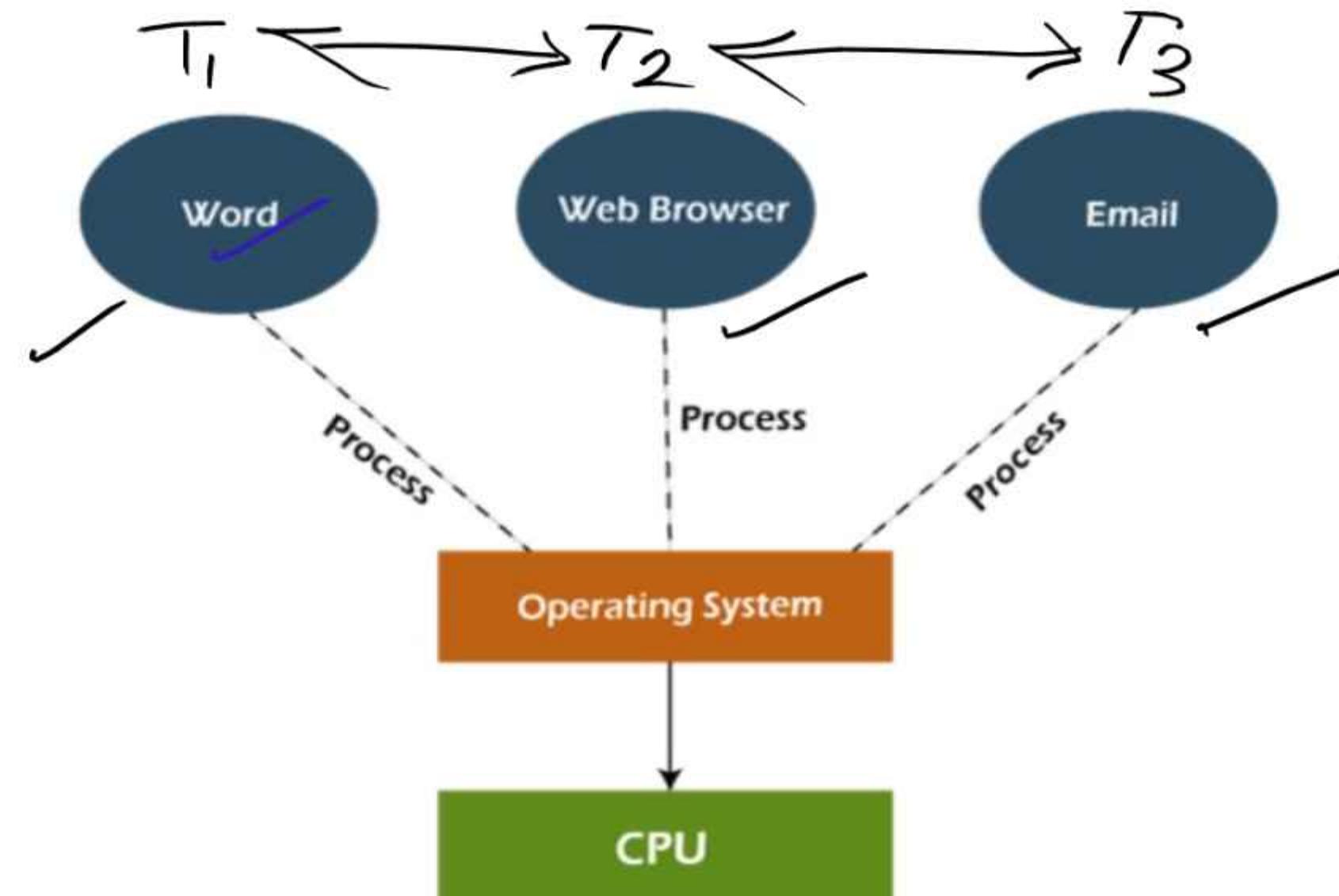


### 3. मल्टीटास्किंग ऑपरेटिंग सिस्टम (Multitasking Operating System)

- **मल्टीटास्किंग ऑपरेटिंग सिस्टम (Multitasking Operating System)** वह ऑपरेटिंग सिस्टम होता है जो एक समय में कई कार्य (टास्क) को एक साथ करने की सुविधा देता है। इसका मतलब यह है कि उपयोगकर्ता एक ही समय में कई एप्लिकेशन (Programs) चला सकता है, जैसे - म्यूजिक सुनना, इंटरनेट ब्राउज़िंग करना और डॉक्यूमेंट टाइप करना, सब कुछ एक साथ।
- A multitasking operating system is an operating system that allows multiple tasks to be performed at the same time. This means that the user can run multiple applications at the same time, such as listening to music, browsing the Internet and typing documents, all at the same time.



## 3. मल्टीटास्किंग ऑपरेटिंग सिस्टम (Multitasking Operating System)





## Disadvantage of Multiprogramming Operating System (नुकसान):

- Allows user to perform multiple tasks at the same time.
- यूजर एक साथ कई काम (जैसे typing, music सुनना, file download) कर सकता है।
- Saves time by switching between tasks quickly.
- कामों के बीच जल्दी-जल्दी switching होने से समय की बचत होती है।
- Increases productivity as user doesn't have to wait.
- यूजर को इंतज़ार नहीं करना पड़ता, जिससे काम जल्दी होता है।
- Provides better system utilization.
- सिस्टम के सभी resources का बेहतर उपयोग होता है



# IMP

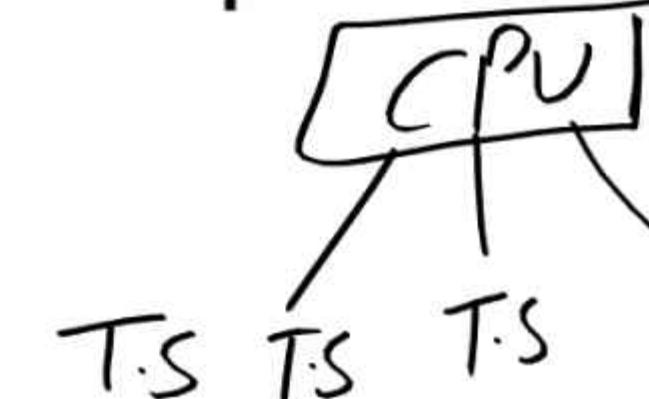
## 4. टाइम-शेयरिंग ऑपरेटिंग सिस्टम

(Time-Sharing Operating System)

T.S  $\Rightarrow$  2 ms

	$P_1$	$P_2$	$P_3$	$P_4$
	1 ms	3 ms	8 ms	10 ms
	2 ms	2 ms	2 ms	2 ms
	8 ms	6 ms	6 ms	8 ms

- टाइम-शेयरिंग ऑपरेटिंग सिस्टम (Time-Sharing Operating System) वह ऑपरेटिंग सिस्टम होता है जो कई उपयोगकर्ताओं (Users) को एक ही समय पर एक ही कंप्यूटर सिस्टम का उपयोग करने की सुविधा देता है।
- इसमें CPU का समय (Time Slice) छोटे-छोटे हिस्सों में बांटा जाता है और हर उपयोगकर्ता को एक निश्चित समय के लिए CPU की सेवा दी जाती है।
- Time-Sharing Operating System is an operating system that allows multiple users to use the same computer system at the same time.
- In this, the CPU time (Time Slice) is divided into small parts and each user is given CPU service for a fixed time.

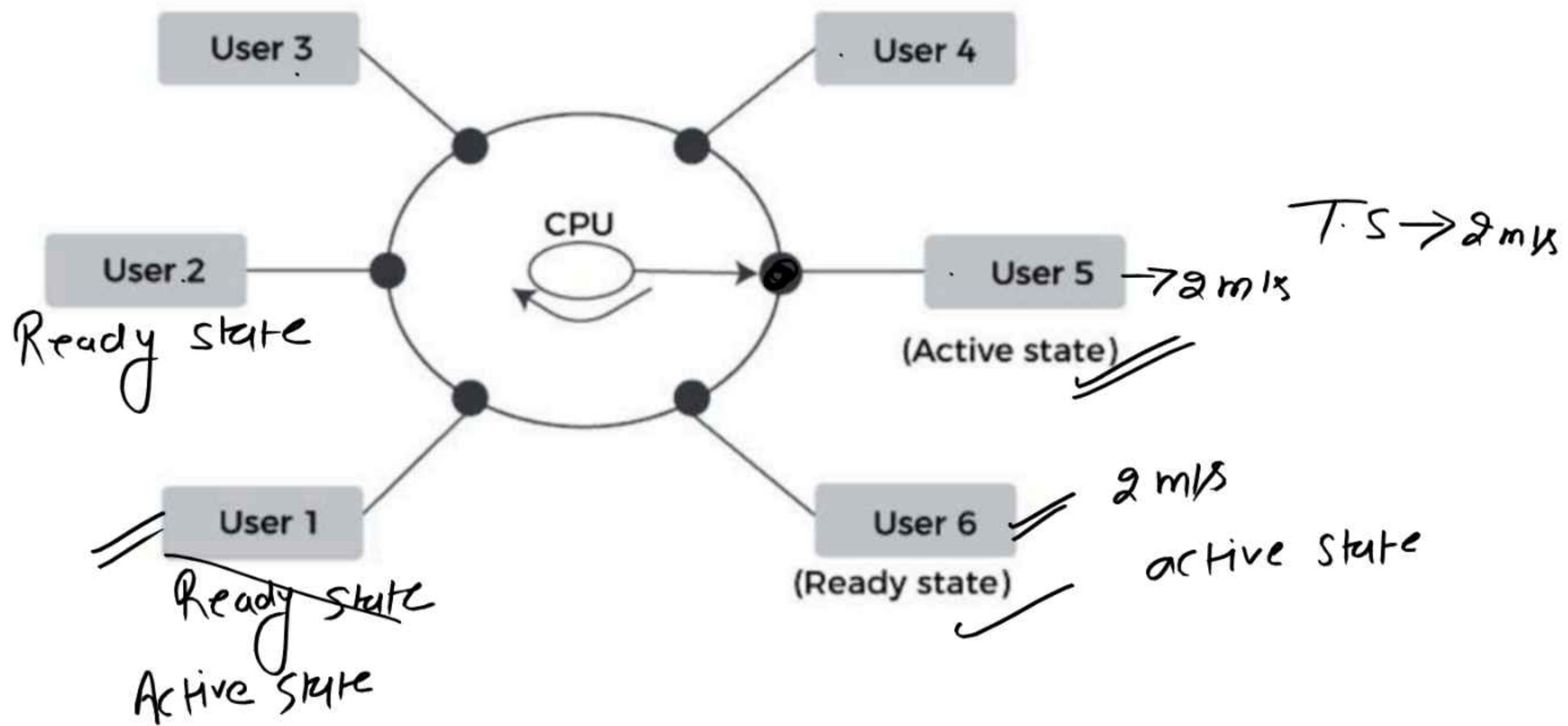




## Example:

- (1): जब हम गूगल ड्राइव (Google Drive) या क्लाउड स्टोरेज (Cloud Storage) का उपयोग करते हैं, तो एक ही सर्वर पर लाखों लोग अपने फाइल्स को एक्सेस कर सकते हैं।
- When we use Google Drive or Cloud Storage, millions of people can access their files on a single server.
- (2): रेलवे रिजर्वेशन सिस्टम में कई उपयोगकर्ता एक साथ टिकट बुक कर सकते हैं।
- In railway reservation system many users can book tickets simultaneously.

11: AM → P<sub>1</sub>  
↓





✓ Advantages (फायदे):

- CPU का पूरा उपयोग होता है। ✓
- CPU is fully utilized.
- एक साथ कई यूजर काम कर सकते हैं ✓
- Multiple users can work simultaneously (multi-user system).
- जल्दी प्रतिक्रिया (Fast response) मिलती है। ✓
- Quick response.
- सभी यूजर्स को बराबर का समय मिलता है। ✓  
$$\begin{array}{l} 2ms \rightarrow P_1 \rightarrow 10ms \\ 2ms \rightarrow P_2 \rightarrow 5ms \\ 2ms \rightarrow P_3 \rightarrow 3ms \end{array} \left. \begin{array}{l} 10ms \\ 5ms \\ 3ms \end{array} \right\} X$$
- All users get equal time.
- रिसोर्स (जैसे प्रिंटर, सॉफ्टवेयर) को शेयर किया जा सकता है। ✓
- Resources (such as printer, software) can be shared. ✓



## ✖ Disadvantages (नुकसान):

- सिस्टम डिजाइन करना जटिल होता है।
- Designing the system is complex.
- डेटा की सुरक्षा में खतरा हो सकता है।
- Data security may be at risk.
- ज्यादा ओवरहेड (CPU बार-बार स्विच करता है)।
- High overhead (CPU switches frequently).
- ज्यादा यूजर्स होने पर सिस्टम धीमा हो सकता है।
- The system may slow down if there are more users.

Imp

## 5. रियल-टाइम ऑपरेटिंग सिस्टम

### (Real-Time Operating System - RTOS):

- रियल-टाइम ऑपरेटिंग सिस्टम (Real-Time Operating System - RTOS) एक विशेष प्रकार का ऑपरेटिंग सिस्टम होता है, जो बहुत तेज़ी से कार्य करता है और किसी कार्य को निर्धारित समय सीमा (Time Constraint) के भीतर पूरा करता है।
- Real-Time Operating System (RTOS) is a special type of operating system, which works very fast and completes a task within the prescribed time limit.
- Such as – aerospace, medical systems, automation, robotics and defense related equipment.

$$\begin{aligned} &= P_1 \rightarrow 20 \text{ ms} \\ &\quad \downarrow 19 \text{ ms} \text{ (fail)} \\ &\quad \downarrow 21 \text{ ms} \text{ (fail)} \end{aligned}$$



## रियल-टाइम ऑपरेटिंग सिस्टम के प्रकार (Types of RTOS):

### 1. हार्ड रियल-टाइम ऑपरेटिंग सिस्टम (Hard Real-Time Operating System):

- इसमें कोई भी देरी (Delay) स्वीकार्य नहीं होती। यदि कार्य निर्धारित समय सीमा में पूरा नहीं हुआ, तो सिस्टम पूरी तरह से फेल हो सकता है।
- Any delay is not acceptable in this. If the work is not completed within the stipulated time frame, the system may fail completely.

### उदाहरण:

- एयरोस्पेस सिस्टम (Aerospace Systems)
- मिसाइल कंट्रोल सिस्टम (Missile Control Systems)



## रियल-टाइम ऑपरेटिंग सिस्टम के प्रकार (Types of RTOS):

### 2. सॉफ्ट रियल-टाइम ऑपरेटिंग सिस्टम (Soft Real-Time Operating System)

- इसमें थोड़ी बहुत देरी स्वीकार्य होती है, लेकिन फिर भी यह तेज़ी से कार्य करता है। अगर कोई कार्य समय सीमा से थोड़ा लेट होता है, तो सिस्टम पर बहुत अधिक प्रभाव नहीं पड़ता।
- A little delay is acceptable but it still works fast. If a task is slightly late from its deadline, there is not much impact on the system.

#### उदाहरण:

- वीडियो स्ट्रीमिंग (YouTube, Netflix, Live Streaming)
- ऑनलाइन ट्रांजैक्शन सिस्टम (Stock Market Trading, Banking Systems)

→ 10: AM  
→ 10: 08 AM  
→ 10 : 00 //



## रियल-टाइम ऑपरेटिंग सिस्टम के प्रकार (Types of RTOS):

### 3. फर्म रियल-टाइम ऑपरेटिंग सिस्टम (Firm Real-Time Operating System):

- इसमें थोड़ी सी देरी सहन की जा सकती है, लेकिन बार-बार होने वाली देरी सिस्टम को बेकार कर सकती है।
- A small delay can be tolerated, but repeated delays can render the system useless.

#### उदाहरण:

- इंडस्ट्रियल रोबोटिक्स (Factory Automation Robots)
- ऑटोमेटिक एयर ट्रैफिक कंट्रोल (Automatic Air Traffic Control System)

Machine → execute → time  
Stop  
↓ start.



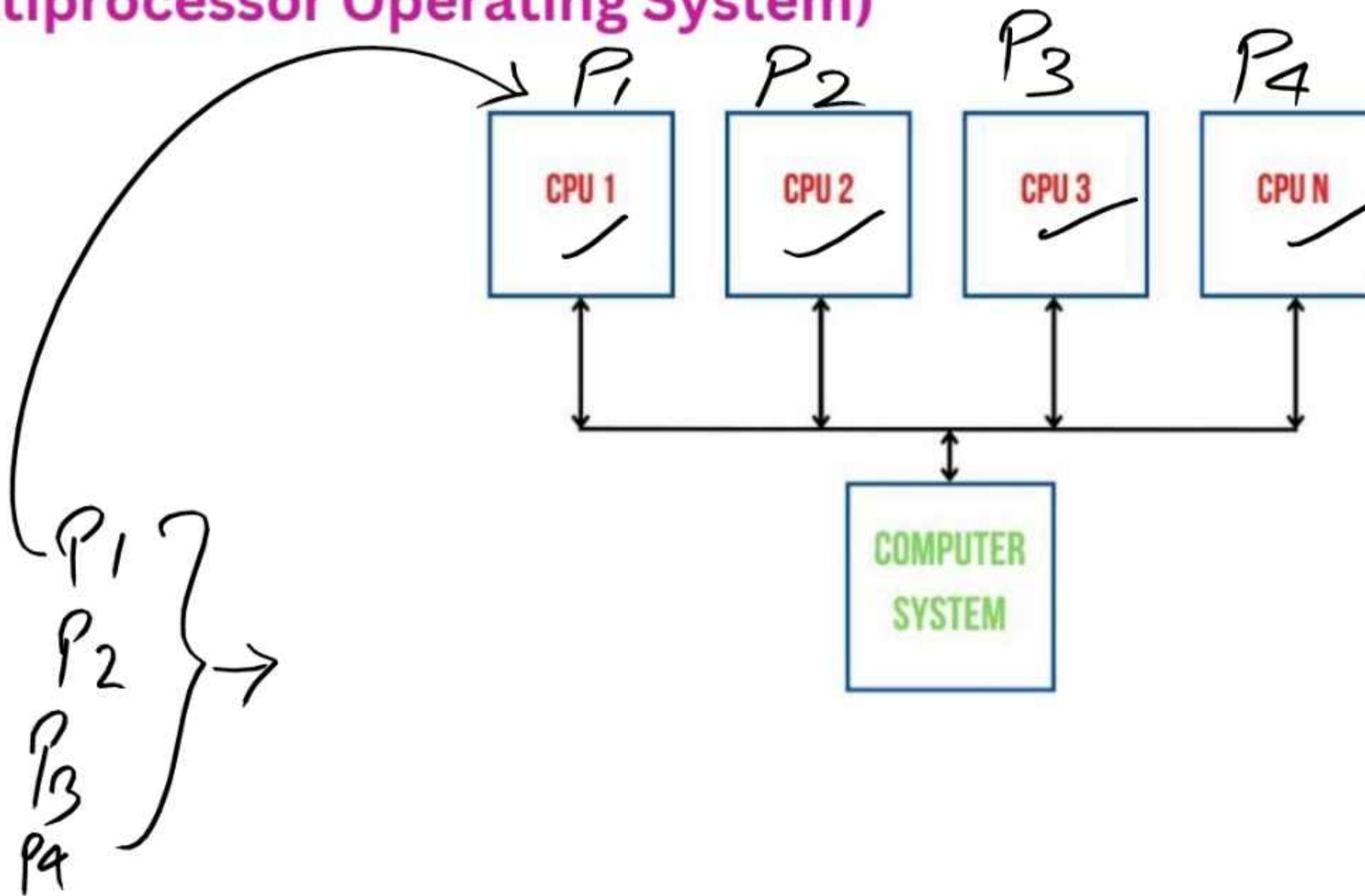
## 6: मल्टीप्रोसेसर ऑपरेटिंग सिस्टम

### (Multiprocessor Operating System)

- मल्टीप्रोसेसर ऑपरेटिंग सिस्टम वह ऑपरेटिंग सिस्टम है जो एक से अधिक प्रोसेसर (CPU) वाले कंप्यूटर सिस्टम को मैनेज और कंट्रोल करता है। इसमें दो या अधिक प्रोसेसर एक साथ काम करते हैं और समान मेमोरी, बस (Bus) और I/O डिवाइसेस को साझा करते हैं।
- A multiprocessor operating system is an operating system that manages and controls a computer system that has more than one processor (CPU). In this, two or more processors work simultaneously and share the same memory, bus and I/O devices.



## 6: मल्टीप्रोसेसर ऑपरेटिंग सिस्टम (Multiprocessor Operating System)

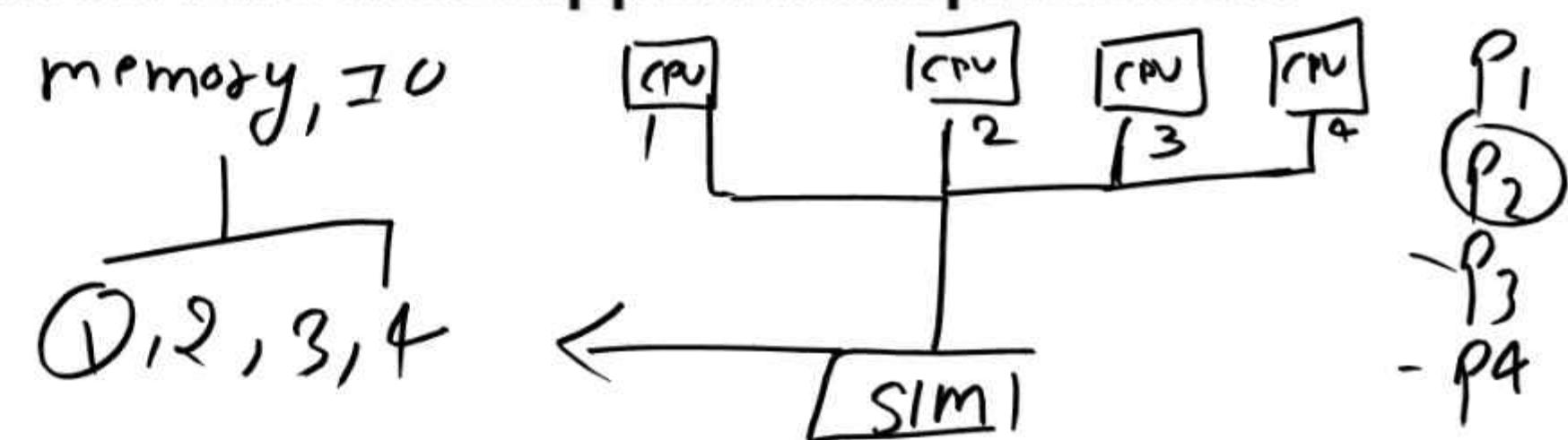




(मल्टीप्रोसेसर सिस्टम के प्रकार (Types of multiprocessor systems):

### 1. सिमेट्रिक मल्टीप्रोसेसिंग (SMP - Symmetric Multiprocessing):

- सभी प्रोसेसर एक समान होते हैं और एक ही ऑपरेटिंग सिस्टम का इस्तेमाल करते हैं।
- हर प्रोसेसर स्वतंत्र रूप से किसी भी कार्य को कर सकता है।
- उदाहरण: Windows, Linux, UNIX सर्वर जो मल्टीप्रोसेसर को सपोर्ट करते हैं।
- All processors are identical and use the same operating system
- Each processor can perform any task independently
- Example: Windows, Linux, UNIX servers that support multiprocessor.

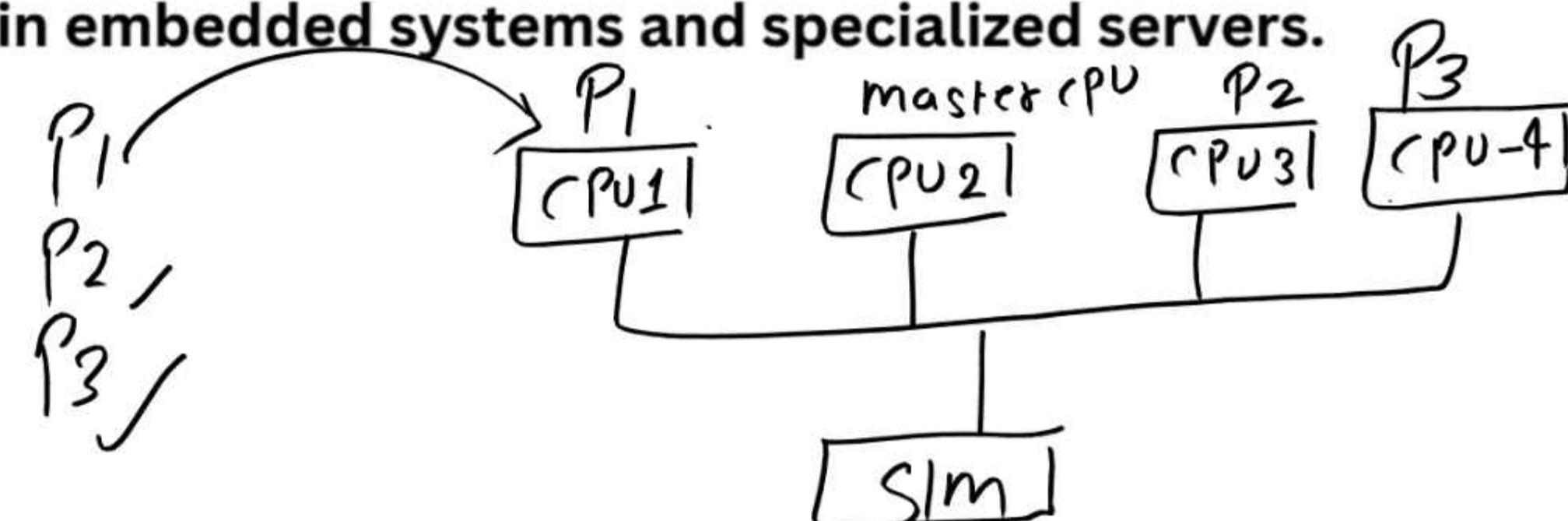




(मल्टीप्रोसेसर सिस्टम के प्रकार (Types of multiprocessor systems):

## 2. असिमेट्रिक मल्टीप्रोसेसिंग (AMP - Asymmetric Multiprocessing):

- एक मुख्य प्रोसेसर (Master Processor) होता है जो अन्य प्रोसेसर (Slave Processors) को काम सौंपता है।
- यह सिस्टम अधिकतर एंबेडेड सिस्टम और स्पेशलाइज्ड सर्वर्स में उपयोग होता है।
- There is a main processor (Master Processor) which assigns work to other processors (Slave Processors).
- This system is mostly used in embedded systems and specialized servers.





40/1  
7

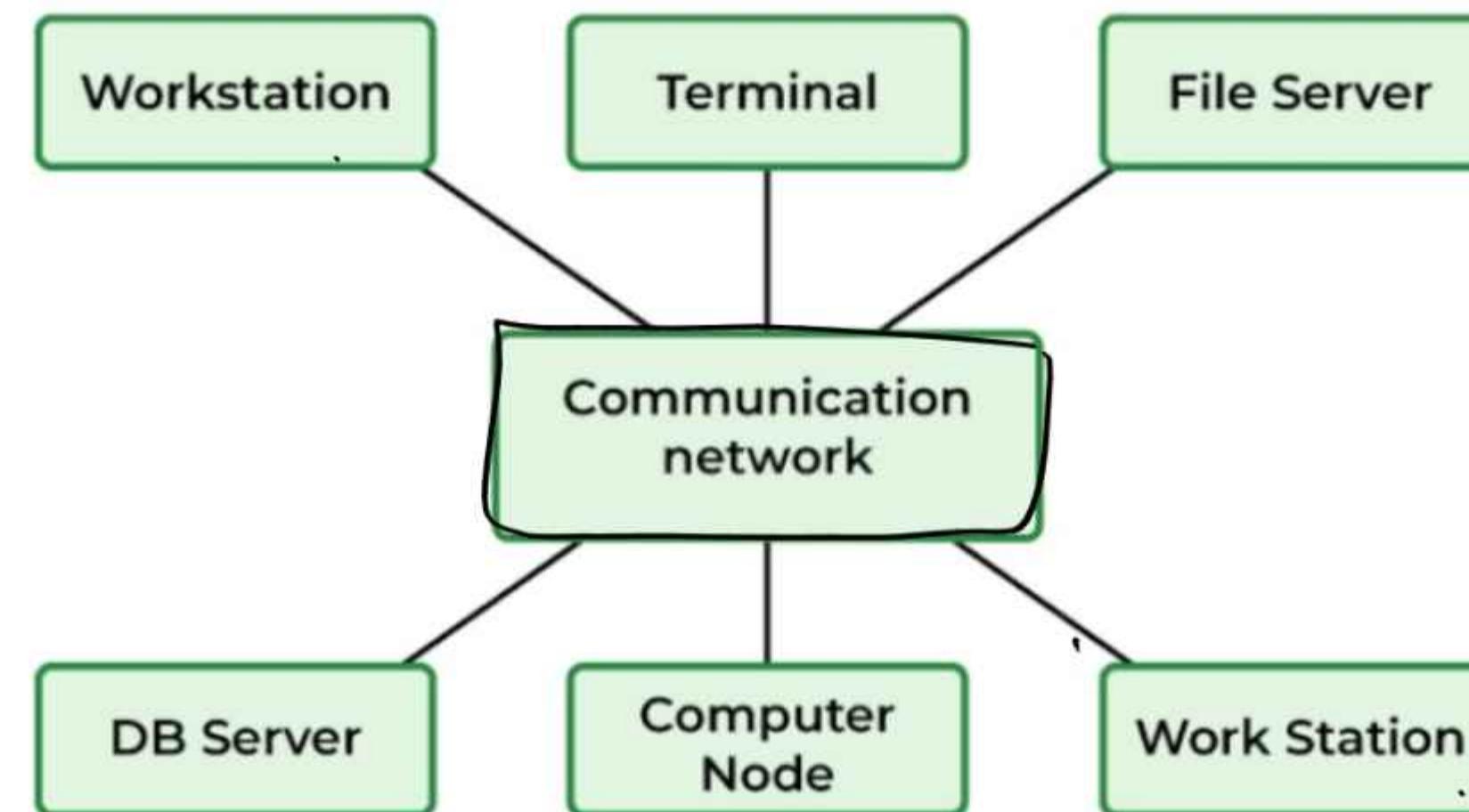
## 7 डिस्ट्रिब्यूटेड ऑपरेटिंग सिस्टम

(Distributed Operating System) :

- डिस्ट्रिब्यूटेड ऑपरेटिंग सिस्टम (Distributed Operating System) एक ऐसा ऑपरेटिंग सिस्टम होता है, जिसमें एक से अधिक कंप्यूटर आपस में जुड़े होते हैं और मिलकर एक ही सिस्टम की तरह कार्य करते हैं।
- Distributed Operating System is an operating system in which more than one computer is interconnected and work together as a single system.
- इसमें कई कंप्यूटर या सर्वर एक नेटवर्क के माध्यम से जुड़े रहते हैं और आपस में संसाधन (Resources) साझा करते हैं।
- In this, many computers or servers are connected through a network and share resources among themselves.



## 6. डिस्ट्रिब्यूटेड ऑपरेटिंग सिस्टम (Distributed Operating System) :





#

## UNIX Operating System:

- UNIX एक शक्तिशाली, मल्टीयूजर और मल्टीटास्किंग ऑपरेटिंग सिस्टम है, जिसे मूल रूप से 1970 के दशक में AT&T की Bell Labs में Ken Thompson और Dennis Ritchie द्वारा विकसित किया गया था।
- UNIX is a powerful, multiuser, multitasking operating system originally developed in the 1970s at AT&T's Bell Labs by Ken Thompson and Dennis Ritchie.
- UNIX को बड़े servers, mainframes और enterprise systems के लिए design किया गया था।
- UNIX was designed for large servers, mainframes, and enterprise systems.

① → Some feature of GUI

② → costly

③ Not open-source

④ Commercial Purpose

⑤ multi-tasking or multi-user

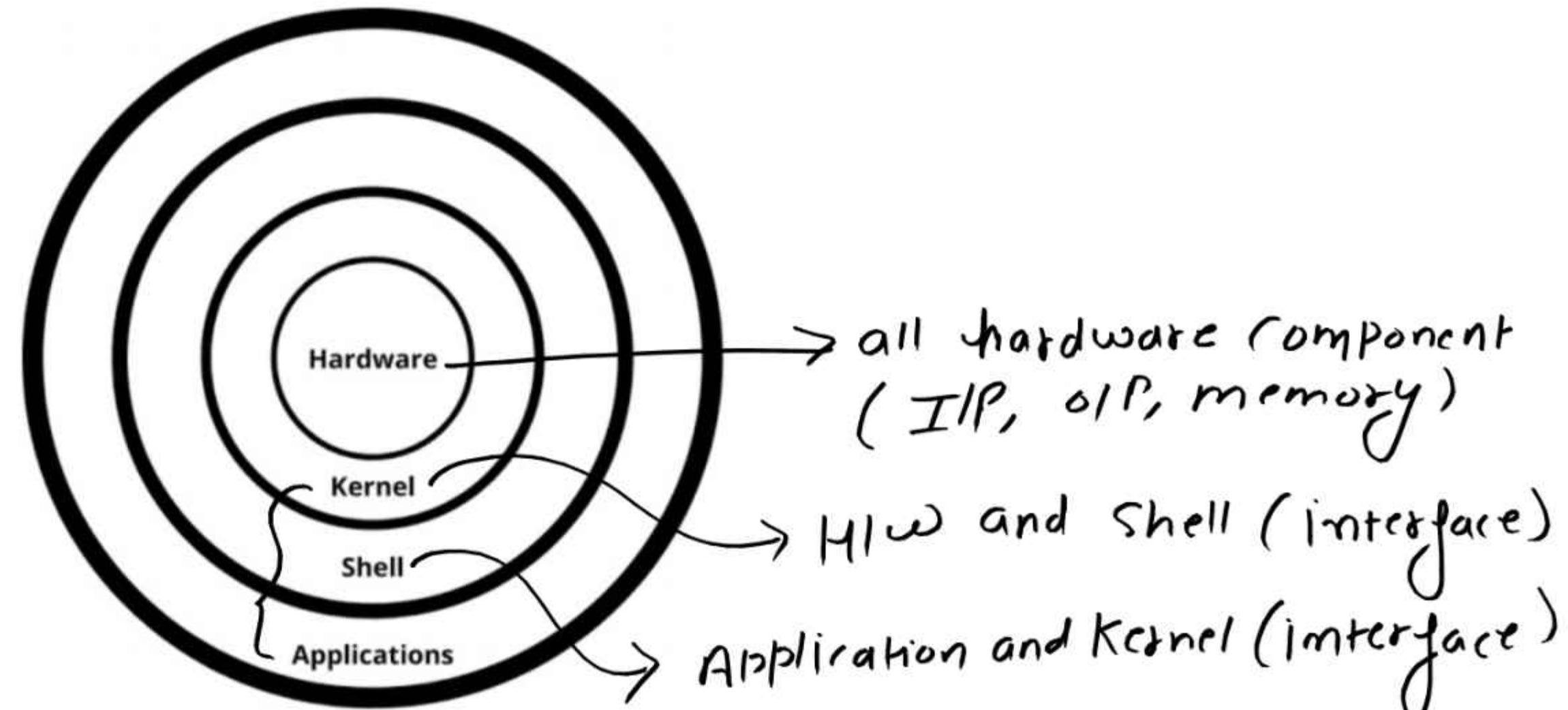


## → **Architecture of UNIX Operating System:**

- UNIX का आर्किटेक्चर एक लेयर आधारित (Layered Architecture) होता है, जिसमें चार मुख्य स्तर (Layers) होते हैं:
- UNIX's architecture is a layered architecture, consisting of four main layers:



## Architecture of UNIX Operating System:





## 1: Hardware (हार्डवेयर)

- **Hardware** UNIX सिस्टम का सबसे अंदर वाला हिस्सा (Inner Core) होता है।
- इसमें वो सभी **Physical Components** आते हैं जो किसी कंप्यूटर सिस्टम को चलाने के लिए ज़रूरी होते हैं।
- **Hardware** is the innermost part (Inner Core) of the UNIX system.
- It contains all the physical components that are required to run a computer system.



## 2. Kernel (कर्नेल):

- Kernel UNIX ऑपरेटिंग सिस्टम का मस्तिष्क (Brain) है।
- यह Hardware और Software के बीच में पुल (Bridge) की तरह काम करता है।
- Kernel is the brain of UNIX operating system.
- It acts as a bridge between hardware and software.



### 3. Shell (शेल):

- Shell एक Command Line Interpreter (CLI) होता है जो User और Kernel के बीच Translator का काम करता है।
- Shell is a Command Line Interpreter (CLI) which works as a translator between the user and the kernel.
- User जो भी command देता है (ls, pwd, cd) – Shell उसे system call में बदलकर Kernel को भेजता है।
- Whatever command the user gives (ls, pwd, cd) – Shell converts it into a system call and sends it to the kernel.



#### 4. Applications (एप्लिकेशन प्रोग्राम्स):

- Applications वो Programs होते हैं जो user को किसी specific काम में मदद करते हैं।
- Applications are programs that help the user in some specific work.

##### Example:

- **Text Editors** – vi, nano
- **Compilers** – gcc, g++
- **File tools** – cp, mv, rm, mkdir
- **Networking tools** – ssh, ftp, ping
- **Monitoring tools** – top, ps, kill



## → Linux Operating System:

- ♦ Linux एक फ्री और ओपन-सोर्स ऑपरेटिंग सिस्टम है, जिसे सबसे पहले 1991 में Linus Torvalds ने विकसित किया था।
- ♦ Linux is a free and open-source operating system, first developed by Linus Torvalds in 1991.
- ♦ यह UNIX के जैसा (Unix-like) है और उसी की तरह एक multiuser, multitasking, और secure सिस्टम है।
- ♦ It is Unix-like and is a multiuser, multitasking, and secure system.
- ♦ आज के समय में Linux का उपयोग servers, supercomputers, mobile phones (Android), और embedded systems में बहुत बड़े पैमाने पर किया जाता है।
- ♦ Today, Linux is used extensively in servers, supercomputers, mobile phones (Android), and embedded systems.



## → ◆ Linux की विशेषताएँ (Features of Linux):

### 1: Multiuser

- Linux ऐसा सिस्टम है जिसे एक ही समय में कई यूज़र इस्तेमाल कर सकते हैं। हर यूज़र का अपना यूज़रनेम, फोल्डर और फाइल्स होती हैं। इससे सिस्टम पर किसी एक यूज़र का काम, दूसरे को प्रभावित नहीं करता।
- Linux is a system that can be used by multiple people at the same time. Everyone has their own appliances, laptops and files. On this system, the work of one person does not affect the work of the other.

### 2: Multitasking (मल्टीटास्किंग)

- Linux एक साथ कई programs या processes को चलाने की क्षमता रखता है।
- Linux has the ability to run multiple programs or processes simultaneously.



### → ◆ Linux की विशेषताएँ (Features of Linux):

#### 3: Open Source (ओपन सोर्स)

- Linux का source code publicly उपलब्ध होता है, जिसे कोई भी देख सकता है, modify कर सकता है, और अपने अनुसार इस्तेमाल कर सकता है
- The source code of Linux is publicly available, which anyone can view, modify, and use as per their requirement

#### 4:: Security (सुरक्षा)

- Linux में बहुत मजबूत security system होता है। इसमें हर फाइल और प्रोसेस के लिए अलग permissions होती हैं
- Linux has a very strong security system. It has different permissions for every file and process.



## → ◆ Linux की विशेषताएँ (Features of Linux):

### 5: Portability :

- Linux को आप किसी भी हार्डवेयर (जैसे- Laptop, Server, Mobile, या Supercomputer) पर चला सकते हैं। यही कारण है कि Android phones भी Linux Kernel पर ही चलते हैं।
- You can run Linux on any hardware (such as Laptop, Server, Mobile, or Supercomputer). This is the reason why Android phones also run on Linux Kernel.

### 6: Customizability

- Linux को आप अपने मनचाहे तरीके से बदल सकते हैं – Desktop, themes, applications सब कुछ आप control में रख सकते हैं।
- You can change Linux the way you want – you can control everything - Desktop, themes, applications.



### → ◆ Linux की विशेषताएँ (Features of Linux):

#### 7: Free of Cost (मुफ्त में उपलब्ध):

- Linux के ज्यादातर versions पूरी तरह से मुफ्त (free) होते हैं। कोई भी इसे डाउनलोड करके install कर सकता है, बिना किसी license की जरूरत के।
- Most versions of Linux are completely free. Anyone can download and install it, without the need for a license.

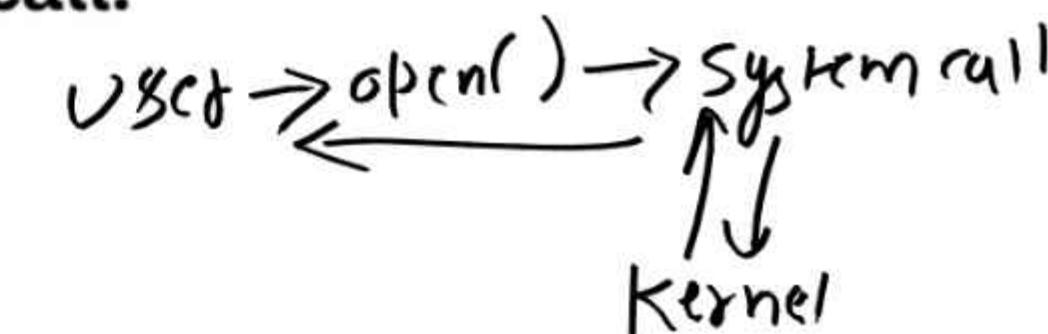
#### 8: Shell और Command Line Support:

- Linux में एक powerful shell होता है (जैसे Bash shell), जिसकी मदद से आप text-based commands से पूरा सिस्टम चला सकते हैं।
- Linux has a powerful shell (like Bash shell), with the help of which you can run the entire system with text-based commands.



## System Call :

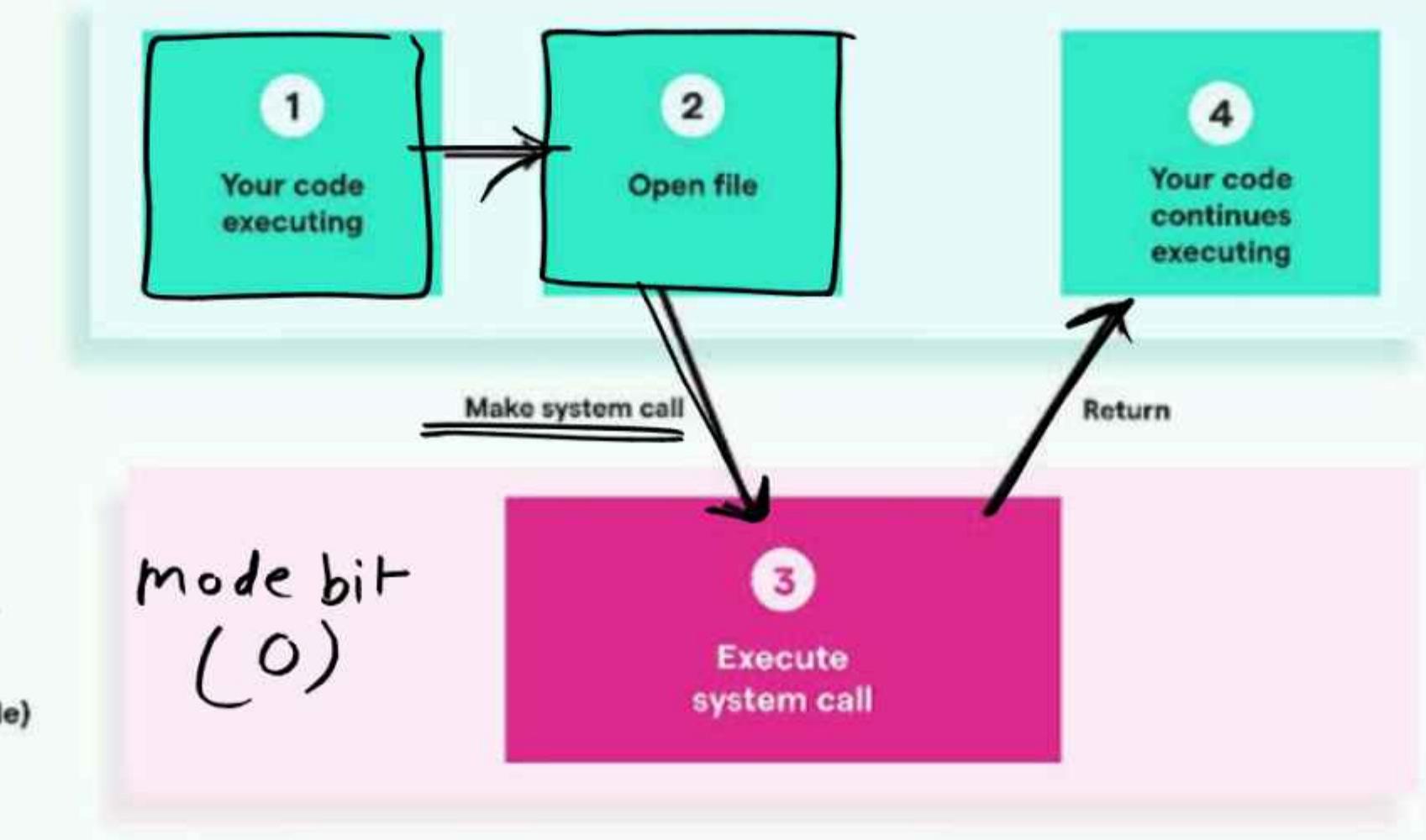
- System Call वह संचार माध्यम (interface) है जिसके द्वारा कोई उपयोगकर्ता प्रोग्राम (user program),ऑपरेटिंग सिस्टम (Operating System) से सेवाएँ (services) प्राप्त करता है।
- System Call is the communication medium (interface) through which a user program receives services from the Operating System.
- यानि जब कोई प्रोग्राम किसी काम के लिए – जैसे फाइल पढ़ना, लिखना, नई प्रोसेस बनाना, या डिवाइस से डेटा लेना –ऑपरेटिंग सिस्टम से मदद मांगता है, तो यह काम System Call के माध्यम से होता है।
- That is, when a program asks the operating system for help with some task—such as reading or writing a file, creating a new process, or getting data from a device—this work is done through a system call.





## System Calls:

$M_B \rightarrow 1$  {  
  USER MODE  
  {  
    KERNEL MODE  
    (privileged mode)



mode bit  $\rightarrow 1$

mode bit  
 $0 \rightarrow$   
 $1 \rightarrow$



## सिस्टम कॉल के फायदे (Advantages of System Calls):

### (1): सुरक्षा (Security):

- सिस्टम कॉल हार्डवेयर को सीधे एक्सेस करने से रोकता है, जिससे सिस्टम सुरक्षित रहता है।
- System calls prevent direct access to hardware, keeping the system secure.

### (2): प्रयोग में सरलता (Ease of Use):

- प्रोग्रामर को हार्डवेयर से सीधे संपर्क नहीं करना पड़ता, वे केवल सिस्टम कॉल का उपयोग करते हैं।
- Programmers do not have to interact with the hardware directly, they only use system calls.

### (3): अधिक कुशलता (Efficiency):

- सिस्टम कॉल का उपयोग करके ऑपरेटिंग सिस्टम संसाधनों का बेहतर प्रबंधन करता है।
- The operating system manages resources better by using system calls.



## → Types of System Call (सिस्टम कॉल के प्रकार):

### 1. प्रोसेस कंट्रोल (Process Control) सिस्टम कॉल:

- प्रोसेस कंट्रोल सिस्टम कॉल का उपयोग नए प्रोसेस बनाने, प्रोसेस को नियंत्रित करने और समाप्त करने के लिए किया जाता है।
- Process control system calls are used to create new processes, control and terminate processes.

### मुख्य कार्य (Main functions):

- नए प्रोसेस बनाना और मैनेज करना (Creating and managing new processes)
- किसी प्रोसेस को समाप्त करना (Terminating a process)
- किसी प्रोसेस की जानकारी प्राप्त करना. (Getting information about a process)
- किसी प्रोसेस को रोकना या फिर से शुरू करना. (Stopping or restarting a process)



## सिस्टम कॉल,कार्य (System Call, Functions):

1. `fork()`: नया प्रोसेस बनाता है। (Creates a new process.)
2. `exec()`: नए प्रोग्राम को लोड और रन करता है। (Loads and runs a new program)
3. `exit()`: वर्तमान प्रोसेस को समाप्त करता है। (Terminates the current process)
4. `wait()`: किसी प्रोसेस के खत्म होने तक इंतजार करता है। (Waits for a process to finish)
5. `kill()`: किसी प्रोसेस को समाप्त करता है। (Terminates a process)
6. `getpid()`: प्रोसेस की ID प्राप्त करता है। (Gets the ID of a process)

$P_1 \xrightarrow{\text{exe}} S/\alpha$   
 $P_2 \downarrow$



## 2. फाइल मैनेजमेंट (File Management) सिस्टम कॉल

- फाइल मैनेजमेंट सिस्टम कॉल का उपयोग फाइल्स और डायरेक्टरीज़ को मैनेज करने के लिए किया जाता है।
- File management system calls are used to manage files and directories.

### मुख्य कार्य(Main functions):

- फाइल को खोलना और बंद करना (Opening and closing a file)
- फाइल से डेटा पढ़ना और लिखना (Reading and writing data from a file)
- फाइल को डिलीट करना या उसकी जानकारी प्राप्त करना
- Deleting or getting information about a file

abc.txt → Delete  
↳ detail



## सिस्टम कॉल,कार्य

1. **open():** फाइल को खोलता है। (**Opens a file.**)
2. **read():** फाइल से डेटा पढ़ता है। (**Reads data from a file:**)
3. **write():** फाइल में डेटा लिखता है। (**Writes data to a file**)
4. **close():** फाइल को बंद करता है। (**Closes a file.**)
5. **unlink():** फाइल को डिलीट करता है। (**Deletes a file.**)



### 3. डिवाइस मैनेजमेंट (Device Management) सिस्टम कॉल:

- डिवाइस मैनेजमेंट सिस्टम कॉल का उपयोग हार्डवेयर डिवाइसेस (जैसे कीबोर्ड, प्रिंटर, डिस्क, मॉनिटर) को नियंत्रित करने के लिए किया जाता है।
- Device management system calls are used to control hardware devices (such as keyboard, printer, disk, monitor).

#### मुख्य कार्य (Main functions):

- इनपुट/आउटपुट डिवाइस को एक्सेस करना (Accessing input/output devices)
- डिवाइस से डेटा पढ़ना और भेजना. (Reading and sending data from a device)
- डिवाइस को बंद करना. (Turning off a device)



### सिस्टम कॉल,कार्य (System call,functions):

1. **ioctl()**: डिवाइस से विशेष जानकारी प्राप्त करता है।
  - retrieves specific information from a device.
2. **read()**: इनपुट डिवाइस (जैसे कीबोर्ड) से डेटा पढ़ता है।
  - reads data from an input device (such as a keyboard)
3. **write()**: आउटपुट डिवाइस (जैसे प्रिंटर) पर डेटा भेजता है।
  - sends data to an output device (such as a printer).
4. **close()**: किसी डिवाइस को बंद करता है।
  - closes a device.



#### 4. सूचना प्रबंधन (Information Management) सिस्टम कॉल:

- सूचना प्रबंधन सिस्टम कॉल का उपयोग सिस्टम और प्रोसेस से जुड़ी जानकारी प्राप्त करने और सेट करने के लिए किया जाता है।
- Information Management System calls are used to get and set system and process information.

#### मुख्य कार्य (Main functions):

- सिस्टम टाइम और डेट प्राप्त करना. (Getting system time and date)
- प्रोसेस की जानकारी प्राप्त करना. (Getting process information)
- सिस्टम स्टेटस देखना. (Viewing system status)



### सिस्टम कॉल, कार्य

1. **getpid()**: वर्तमान प्रोसेस की ID प्राप्त करता है।
2. **alarm()**: टाइमर सेट करता है।
3. **sleep()**: प्रोसेस को कुछ समय के लिए रोकता है।
4. **gettimeofday()**: सिस्टम का वर्तमान समय प्राप्त करता है।

### System Calls, Functions

1. **getpid()**: Gets the ID of the current process.
2. **alarm()**: Sets a timer.
3. **sleep()**: Pauses the process for some time.
4. **gettimeofday()**: Gets the current system time.

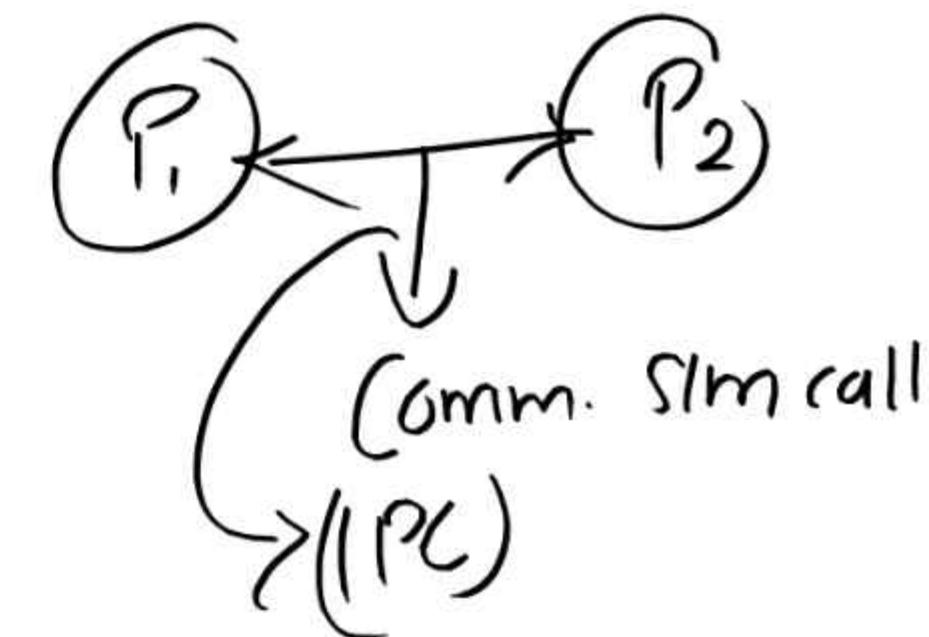


## 5. संचार (Communication) सिस्टम कॉल:

- संचार सिस्टम कॉल का उपयोग अलग-अलग प्रोसेस के बीच डेटा का आदान-प्रदान (Inter-Process Communication - IPC) करने के लिए किया जाता है।
- Communication system calls are used to exchange data between different processes (Inter-Process Communication - IPC).

### मुख्य कार्य:

- प्रोसेस के बीच डेटा भेजना और प्राप्त करना
- Sending and receiving data between processes
- नेटवर्क पर डेटा ट्रांसफर करना
- Transferring data over the network
- शेयर की गई मेमोरी का उपयोग करना
- Using shared memory





## सिस्टम कॉल, कार्य

1. **pipe()**: दो प्रोसेस के बीच डेटा भेजता है।
2. **shmget()**: साझा मेमोरी (Shared Memory) प्राप्त करता है।
3. **msgsnd()**: संदेश (Message) भेजता है।
4. **msgrcv()**: संदेश प्राप्त करता है।
5. **socket()**: नेटवर्क पर डेटा भेजता और प्राप्त करता है।

## System Calls, Functions

1. **pipe()**: Sends data between two processes.
2. **shmget()**: Acquires shared memory.
3. **msgsnd()**: Sends messages.
4. **msgrcv()**: Receives messages.
5. **socket()**: Sends and receives data over the network.

~~Imp~~

### Kernel (कर्नेल):

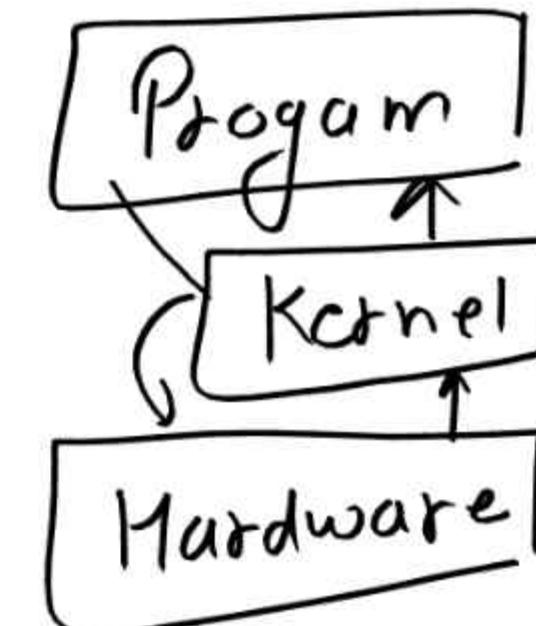
- Kernel (कर्नेल) ऑपरेटिंग सिस्टम का सबसे मुख्य हिस्सा होता है। यह सॉफ्टवेयर (Programs) और हार्डवेयर (Hardware) के बीच एक सेतु (bridge) की तरह काम करता है।
- Kernel is the most important part of the operating system. It acts like a bridge between software and hardware.

कर्नेल का मुख्य कार्य होता है:

- कंप्यूटर के संसाधनों (CPU, RAM, I/O devices) को मैनेज करना,
- प्रोग्राम्स और हार्डवेयर के बीच संपर्क स्थापित करना।

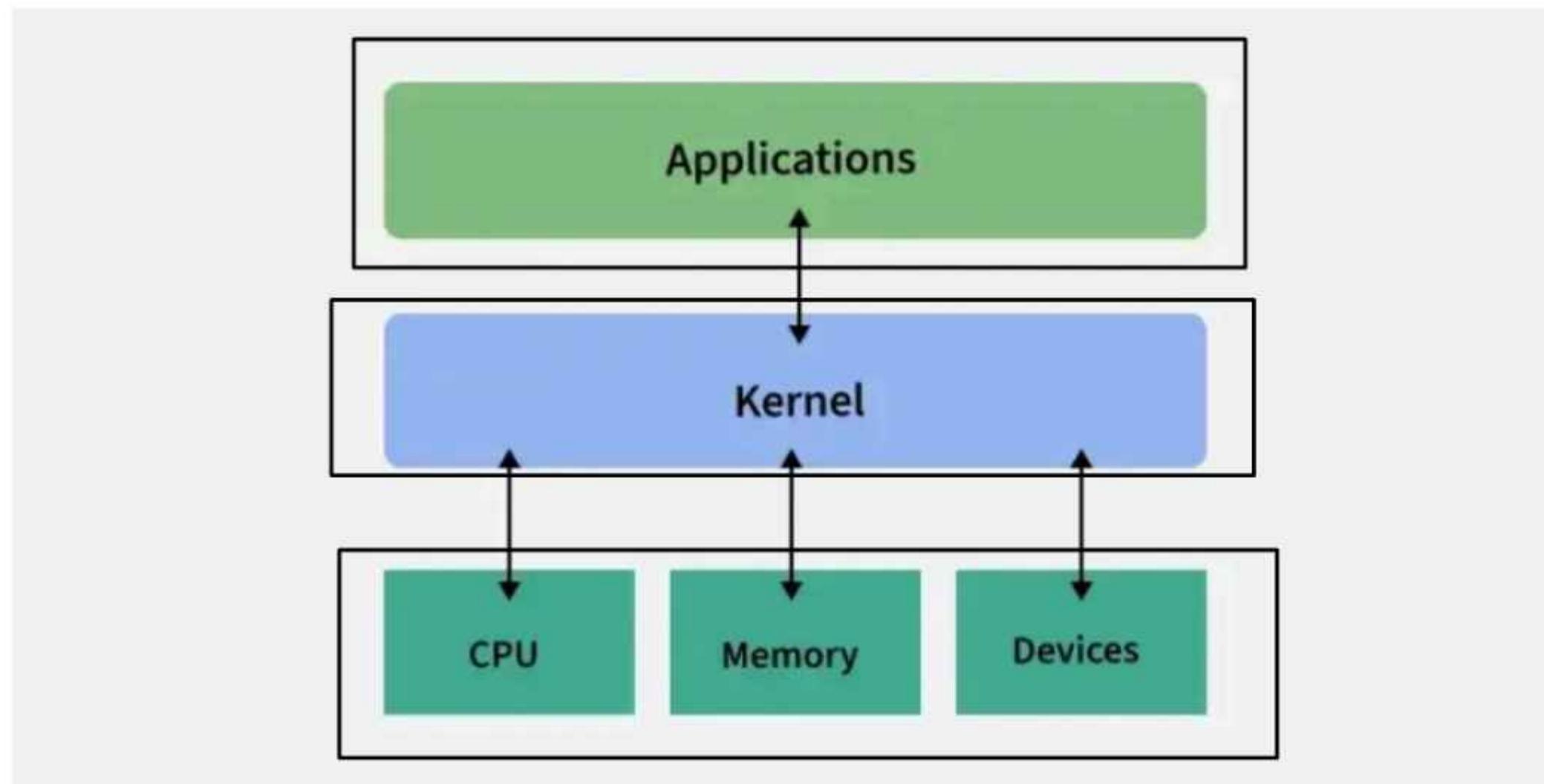
The main functions of the kernel are:

- Managing the computer's resources (CPU, RAM, I/O devices),
- Establishing communication between programs and hardware.





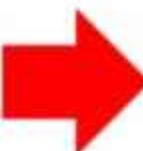
## Kernel (कर्नेल):





## कर्नेल के मुख्य कार्य (Main Functions of Kernel):

क्र.सं.	कार्य का नाम (Function Name)	विवरण (Discription)
1	Process Management (प्रोसेस प्रबंधन)	कर्नेल यह तय करता है कि कौन-सा प्रोग्राम कब और कैसे चलेगा, उसे CPU कब मिलेगा, और उसे कब रोका या बंद किया जाएगा।The kernel manages the creation, scheduling, and termination of processes.
2	Memory Management (मेमोरी प्रबंधन)	कर्नेल RAM को सभी प्रोग्राम्स में सही तरीके से बांटता है और यह सुनिश्चित करता है कि एक प्रोग्राम दूसरे की मेमोरी में हस्तक्षेप न करे।It allocates and deallocates memory to processes and keeps track of memory usage.
3	Device Management (डिवाइस प्रबंधन)	सभी इनपुट/आउटपुट डिवाइसेस (जैसे कीबोर्ड, माउस, प्रिंटर) को कंट्रोल करने का काम कर्नेल करता है।The kernel uses device drivers to communicate with hardware components.



क्र.सं.	कार्य का नाम (Function Name)	विवरण (Discription)
	File System Management (फाइल सिस्टम प्रबंधन)	कर्नेल फाइलों को पढ़ने, लिखने और स्टोर करने की प्रक्रिया को मैनेज करता है।It handles file reading, writing, permissions, and storage.
	System Calls Handling (सिस्टम कॉल का संचालन)	जब कोई प्रोग्राम ऑपरेटिंग सिस्टम से कोई सेवा मांगता है (जैसे डेटा पढ़ना), तो कर्नेल उस अनुरोध को संभालता है।System calls are how applications interact with the OS, and the kernel handles those calls.
	Security and Access Control (सुरक्षा और पहुँच नियंत्रण)	कर्नेल यह तय करता है कि कौन-से प्रोग्राम और यूजर को कौन-सी जानकारी या संसाधन तक पहुँच मिलेगी।It enforces permissions and access controls to protect system resources.



### → कर्नेल के प्रकार (Types of Kernel):

#### 1. Monolithic Kernel (मोनोलिथिक कर्नेल):

- Monolithic kernel एक ऐसा कर्नेल होता है जिसमें सभी आवश्यक सेवाएँ – जैसे कि मेमोरी मैनेजमेंट, फाइल सिस्टम, डिवाइस ड्राइवर, इंटरप्रोसेस कम्युनिकेशन आदि – एक ही बड़े ब्लॉक (process) में रन करती हैं और सब कुछ kernel space में ही होता है।
- A monolithic kernel is a kernel in which all essential services – such as memory management, file system, device drivers, interprocess communication, etc. – run in a single large block (process), and everything happens in kernel space.

#### विशेषताएँ (Features):

- सभी सेवाएँ कर्नेल के अंदर होती हैं। / All services are inside the kernel.
- सिस्टम कॉल्स का जवाब बहुत तेज़ी से मिलता है। / System calls are responded to very quickly.
- हार्डवेयर से डायरेक्ट और तेज़ संपर्क। / Direct and fast connection to hardware.



## फायदे (Advantages):

- Performance बहुत अच्छी होती है। / Performance is very good.
- हार्डवेयर से तेजी से कम्युनिकेशन होता है। / Communication with the hardware is fast.

## नुकसान (Disadvantages):

- अगर एक सर्विस फेल हो जाए, तो पूरा सिस्टम क्रैश हो सकता है।  
• If one service fails, the whole system may crash.  
• डिबगिंग और मेंटेनेंस कठिन होती है।  
• Debugging and maintenance are difficult.

## उदाहरण (Example):

- Linux Kernel
- Unix Kernel
- MS-DOS



## 2. Microkernel (माइक्रो कर्नेल):

- Microkernel में केवल सबसे जरूरी सेवाएँ ही kernel space में होती हैं (जैसे कि CPU scheduling, memory management, IPC), जबकि बाकी सभी सेवाएँ (जैसे फाइल सिस्टम, डिवाइस ड्राइवर) user space में चलाई जाती हैं।
- In a microkernel, only the most essential services are run in kernel space (such as CPU scheduling, memory management, IPC), while all other services (such as file system, device drivers) are run in user space.

### विशेषताएँ (Features):

- Kernel बहुत छोटा और modular होता है।
- The kernel is very small and modular.
- User mode में ज्यादातर services चलती हैं।
- Most services run in user mode.

→ User mode (Non - important)

→ Kernel mode (important services)



## फायदे (Advantages):

- अधिक सुरक्षा और स्थिरता (Stability) मिलती है। ✓ More security and stability is provided.
- एक सर्विस क्रैश हो जाए तो सिस्टम क्रैश नहीं होता। ✓
- If a service crashes, the system does not crash.

## नुकसान (Disadvantages):

- Performance थोड़ी कम हो सकती है। ✓
- Performance may be slightly slower.
- सिस्टम कॉल्स और communication में समय ज्यादा लगता है। ✓
- System calls and communication take more time. ✓

## उदाहरण (Example):

- Minix ✓
- QNX ✓
- L4 Microkernel ✓
- Mach (macOS का base) ✓



### 3. Hybrid Kernel (हाइब्रिड कर्नेल):

- Hybrid kernel एक ऐसा कर्नेल होता है जो Monolithic और Microkernel दोनों की विशेषताओं को मिलाकर बनाया गया होता है। इसमें कुछ सेवाएँ kernel space में होती हैं और कुछ user space में।
- Hybrid kernel is a kernel that is built by combining the features of both Monolithic and Microkernel. In this, some services are in kernel space and some in user space.

#### विशेषताएँ (Features):

- Monolithic की तरह performance और Microkernel की तरह modularity देता है। / Provides performance like Monolithic and modularity like Microkernel.
- सिस्टम कॉल्स तेज़ होती हैं और सेवाएं सेफ रहती हैं। / System calls are fast and services are safe.



## फायदे (Advantages):

- बेहतर परफॉर्मेंस और बेहतर सुरक्षा का संतुलन।
- Balances better performance and better security.
- नए फीचर्स जोड़ना आसान होता है।
- Addition of new features is easy.

## नुकसान (Disadvantages):

- संरचना जटिल होती है।
- The structure is complex.
- ज्यादा मेमोरी इस्तेमाल हो सकती है।
- May use more memory.

## उदाहरण (Example):

- Windows NT kernel (Windows XP, 7, 10, 11)
- macOS (XNU kernel)



## System Program:

- System programs वे प्रोग्राम होते हैं जो ऑपरेटिंग सिस्टम (OS) को चलाने, मेंटेन करने और यूज़र को सुविधाएं देने का काम करते हैं। ये प्रोग्राम हार्डवेयर और यूज़र के बीच एक इंटरफ़ेस का काम करते हैं।
- System programs are those programs that run, maintain and provide facilities to the user of the operating system (OS). These programs act as an interface between the hardware and the user.



## → Type of System Program (सिस्टम प्रोग्राम का प्रकार):

### 1. File Management Programs (फाइल मैनेजमेंट प्रोग्राम):

- फाइल मैनेजमेंट प्रोग्राम वह सॉफ्टवेयर होता है, जो कंप्यूटर में फाइल्स और फोल्डर्स को बनाने, हटाने, कॉपी करने, मूव करने और एडिट करने में मदद करता है।
- A file program is software that helps create, delete, copy, move and edit files and machinery in a computer.

उदाहरण:

- Windows Explorer,
- Linux ls command, cp, mv, rm

↑  
cp → copy  
mv → move  
rm → remove



## (2): Process Management Programs (प्रोसेस मैनेजमेंट प्रोग्राम):

- Process Management Program वह सॉफ्टवेयर है, जो कंप्यूटर में चल रहे सभी प्रोग्राम्स (Processes) को कंट्रोल करता है। यह तय करता है कि कौन सा प्रोग्राम कब और कितना चलेगा, उसे कितनी मेमोरी (RAM) और प्रोसेसिंग पावर (CPU) मिलेगी, और जरूरत पड़ने पर उसे बंद भी कर सकता है।
- Process Management Program is the software that controls all the programs (processes) running in the computer. It decides which program will run when and how much, how much memory (RAM) and processing power (CPU) it will get, and can also close it if needed.

### Example:

- Window (CMD): (tasklist, taskkill /IM process-name, taskkill /PID process-id)



### 3. Device Management Programs (डिवाइस मैनेजमेंट प्रोग्राम):

- डिवाइस मैनेजमेंट प्रोग्राम वह सॉफ्टवेयर होते हैं, जो कंप्यूटर या मोबाइल के सभी हार्डवेयर डिवाइसेस (जैसे कीबोर्ड, माउस, प्रिंटर, USB, डिस्क ड्राइव, आदि) को कंट्रोल और मैनेज करते हैं।
- **Device management programs are software that controls and manages all the hardware devices (such as keyboard, mouse, printer, USB, disk drive, etc.) of a computer or mobile.**

### 4. Communication Programs (कम्युनिकेशन प्रोग्राम):

- कम्युनिकेशन प्रोग्राम वह सॉफ्टवेयर होते हैं, जो कंप्यूटर और अन्य डिवाइसों (जैसे स्मार्टफोन, सर्वर आदि) के बीच डेटा, संदेश, या सूचनाओं का आदान-प्रदान (communication) करने में मदद करते हैं।
- **Communication programs are software that helps exchange data, messages, or information between computers and other devices (such as smartphones, servers, etc.).**



Unit No 2:

## UNIT 2: Process Management:



Process concepts



Operations on processes



CPU Scheduling, Scheduling Algorithms,

• IPC

• Process synchronization

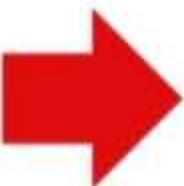
• critical section, Deadlock, Condition, Avoidance &amp; Prevention



⊗ 99.1%

(4)

10 → marks



## Process Concept:

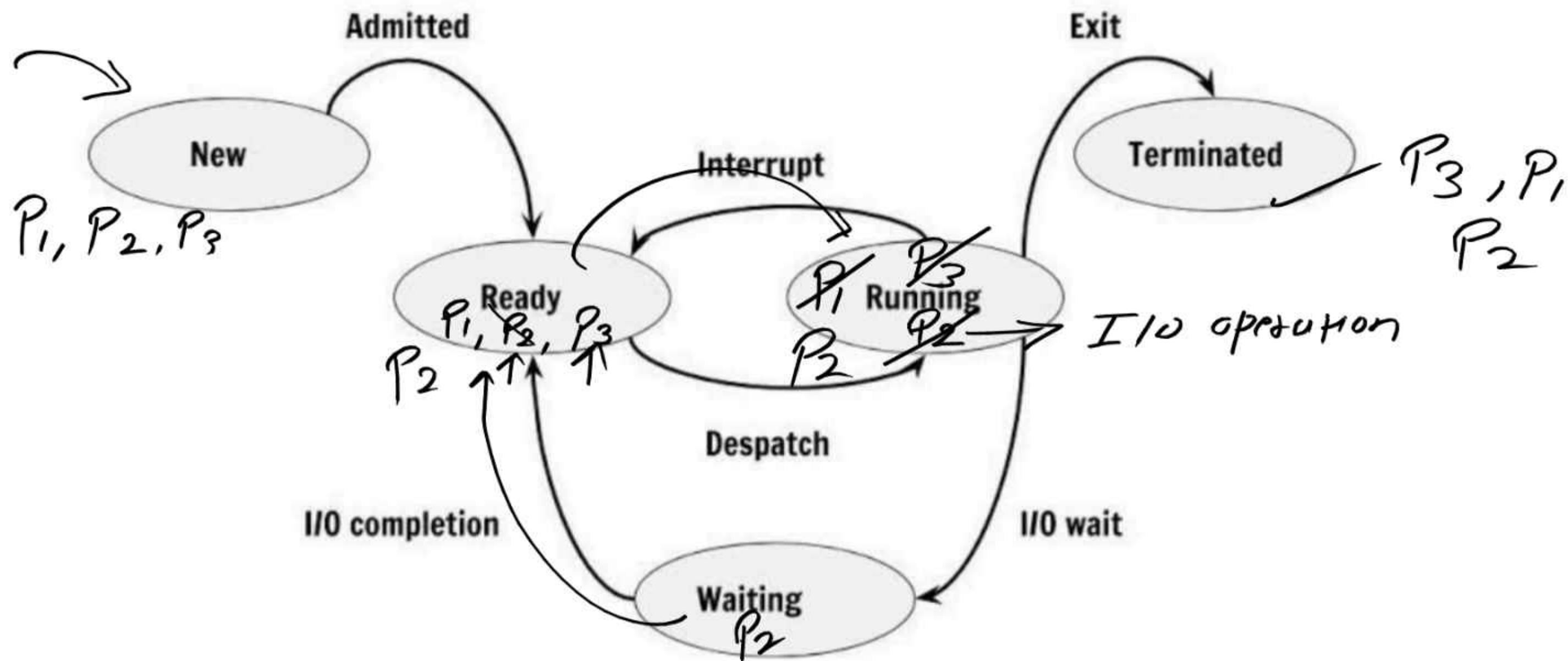
- इसे एक एक्जीक्यूशन यूनिट (Execution Unit) कहा जा सकता है,
- This can be called an execution unit.
- कंप्यूटर में जब हम कोई प्रोग्राम रन करते हैं, तो वह प्रोसेस (Process) बन जाता है। उदाहरण के लिए, अगर आप MS Word खोलते हैं, तो वह एक प्रोसेस बन जाता है।
- When we run a program in the computer, it becomes a process. For example, if you open MS Word, it becomes a process.
- ऑपरेटिंग सिस्टम प्रोसेस को बनाने (Create), शेड्यूल करने (Schedule) और खत्म करने (Terminate) में मदद करता है।
- The operating system helps to create, schedule and terminate processes.



## प्रोसेस स्टेट ( Process State):

- जब भी कोई प्रोग्राम रन करता है, तो वह एक प्रोसेस बन जाता है। यह प्रोसेस हमेशा एक ही स्थिति (State) में नहीं रहता, बल्कि अलग-अलग स्टेट्स (States) से होकर गुजरता है। इसे प्रोसेस स्टेट कहा जाता है।
- Whenever a program runs, it becomes a process. This process does not always remain in the same state, but passes through different states. This is called the process state.

- ① New State ( जब इस Process उन्होंने ही )
- ② Ready state ( Process ऑफ रीडी टॉ एक्सेक्यूटिव )
- ③ Running state ( Process is Executed )
- ④ Waiting ( Perform I/O operation )
- ⑤ → terminate  
( complete its execution )





### 1. न्यू (New State) :

- जब कोई नया प्रोग्राम शुरू होता है, तो पहले वह “न्यू स्टेट” में रहता है।
- उदाहरण:** जब आप MS Word खोलते हैं, तो वह पहले “नया प्रोसेस” बनता है।
- When a new program starts, it is first in “New State”.
- Example:** When you open MS Word, it first becomes a “New Process”.

### 2. रेडी (Ready State):

- जब प्रोसेस रन करने के लिए तैयार होता है लेकिन CPU उसे तुरंत उपलब्ध नहीं होता। ऑपरेटिंग सिस्टम इस प्रोसेस को रेडी क्यू (Ready Queue) में डाल देता है।
- उदाहरण:** आपने MS Word खोला, लेकिन CPU पहले किसी और प्रोसेस को चला रहा है, इसलिए यह प्रोसेस “रेडी” स्टेट में चला जाता है।
- When a process is ready to run but the CPU is not immediately available to it.
- The operating system puts this process in the Ready Queue.
- Example:** You opened MS Word, but the CPU is running another process first, so this process goes into “ready” state.



### 3. रनिंग (Running State):

- जब CPU किसी प्रोसेस को एक्सिक्यूट करने के लिए चुनता है, तो वह “रनिंग स्टेट” में आ जाता है।
- इस स्टेट में प्रोसेस CPU का उपयोग करके अपने इंस्ट्रक्शन्स को चलाता है।
- When the CPU selects a process to execute, it enters the “running state”.
- In this state the process executes its instructions using the CPU.

### 4. वेटिंग (Waiting State):

- जब किसी प्रोसेस को चलने के लिए किसी इनपुट या रिसोर्स का इंतजार करना पड़े, तो वह “वेटिंग स्टेट” में चला जाता है।
- When a process has to wait for some input or resource to run, it goes into a “waiting state”.





## 5. टर्मिनेटेड (Terminated State)

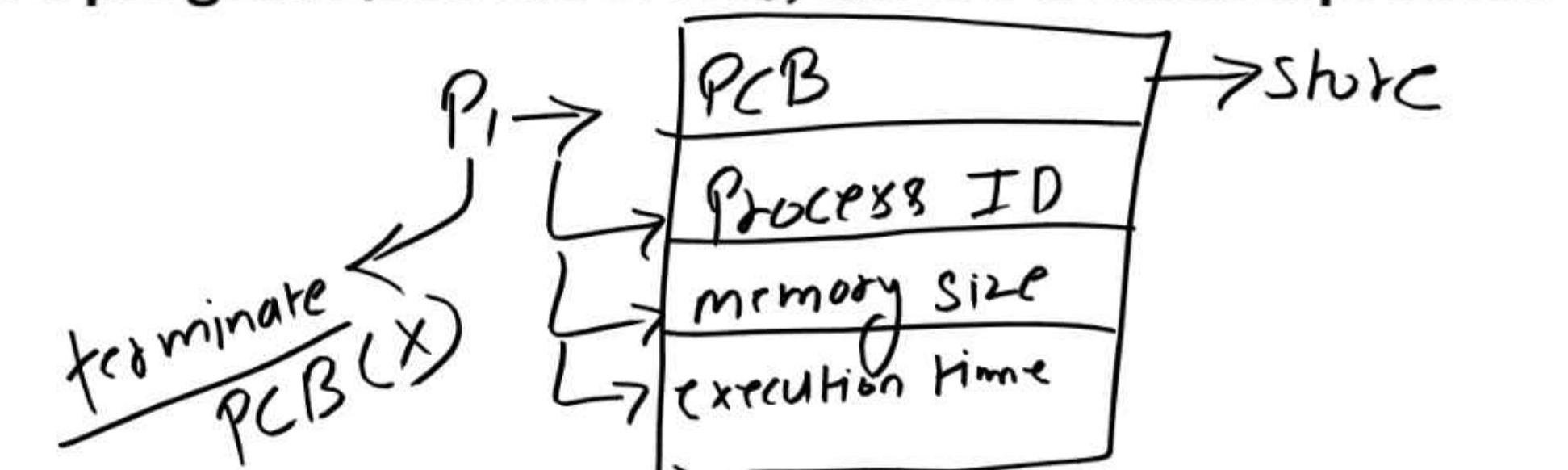
- जब जब प्रोसेस अपना काम पूरा कर लेता है या जब उसे जबरदस्ती बंद कर दिया जाता है, तो वह “टर्मिनेटेड स्टेट” में चला जाता है।
- When a process completes its work or when it is forcibly shut down, it goes into a “terminated state”.



## Operation of Process (प्रक्रिया का संचालन):

### (1): Process Creation (प्रोसेस का निर्माण):

- यह पहला स्टेप होता है जिसमें नया प्रोसेस बनाया जाता है।
- This is the first step in which a new process is created.
- OS (Operating System) एक नया PCB (Process Control Block) तैयार करता है।
- The OS (Operating System) creates a new PCB (Process Control Block).
- उदाहरण: जब आप कोई प्रोग्राम खोलते हैं (जैसे MS Word), तो OS उसके लिए प्रोसेस बनाता है।
- Example: When you open a program (like MS Word), the OS creates a process for it.





## (2): Process Scheduling (प्रोसेस अनुसूची बनाना):

- CPU को यह तय करना होता है कि किस प्रोसेस को कब CPU मिलना चाहिए।
- The CPU has to decide which process should get the CPU and when.
- इसके लिए OS Scheduling Algorithms का उपयोग करता है।
- For this the OS uses Scheduling Algorithms.
- उदाहरण: Round Robin, First Come First Serve आदि।

$P_1, P_2, P_3$   
Ready  $\rightarrow$

## (3): Process Execution (प्रोसेस का निष्पादन):

- अब प्रोसेस को CPU मिलता है और वह अपना कार्य करना शुरू करता है। इसमें instructions को execute किया जाता है।
- Now the process gets the CPU and starts doing its work. Instructions are executed in it.



#### (4): I/O Operation (इनपुट/आउटपुट कार्य):

- प्रोसेस को अगर किसी फाइल से data चाहिए या यूजर से input चाहिए, तो वो I/O सिस्टम से communicate करता है। प्रोसेस को CPU से हटा दिया जाता है और waiting queue में रखा जाता है।
- If the process needs data from a file or input from the user, it communicates with the I/O system. The process is removed from the CPU and placed in the waiting queue.

#### (5): Inter-Process Communication (IPC):

- कई बार एक प्रोसेस को दूसरे प्रोसेस से communicate करना होता है। जैसे एक process result भेजता है और दूसरा उसे use करता है। इसके लिए Pipe, Message Queue, Shared Memory आदि का उपयोग किया जाता है।
- Many times a process has to communicate with another process. Like one process sends the result and the other uses it. Pipe, Message Queue, Shared Memory etc. are used for this.





### (6): Process Termination (प्रोसेस समाप्ति):

- जब प्रोसेस अपना कार्य पूरा कर लेता है, तो वह terminate हो जाता है। OS उस प्रोसेस का PCB delete कर देता है और resources वापस ले लेता है।
- When the process completes its work, it terminates. The OS deletes the PCB of that process and takes back the resources.

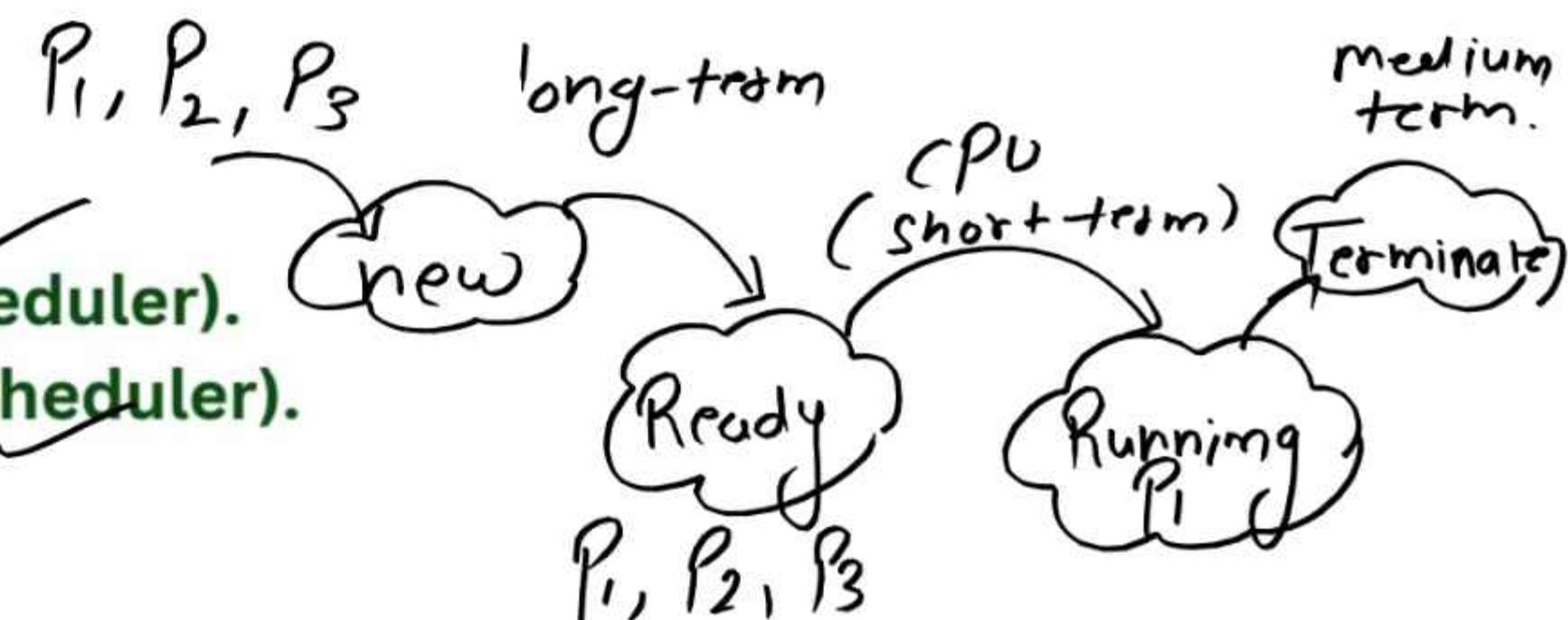


## Scheduler (शेड्यूलर):

- शेड्यूलर (Scheduler) ऑपरेटिंग सिस्टम का एक महत्वपूर्ण हिस्सा है जिसका काम यह तय करना होता है कि कौन सा प्रोसेस या कार्य CPU पर कब और कितने समय के लिए चलेगा।
- Scheduler is an important part of the operating system whose job is to decide which process or task will run on the CPU, when and for how long.

### Types of Scheduler:

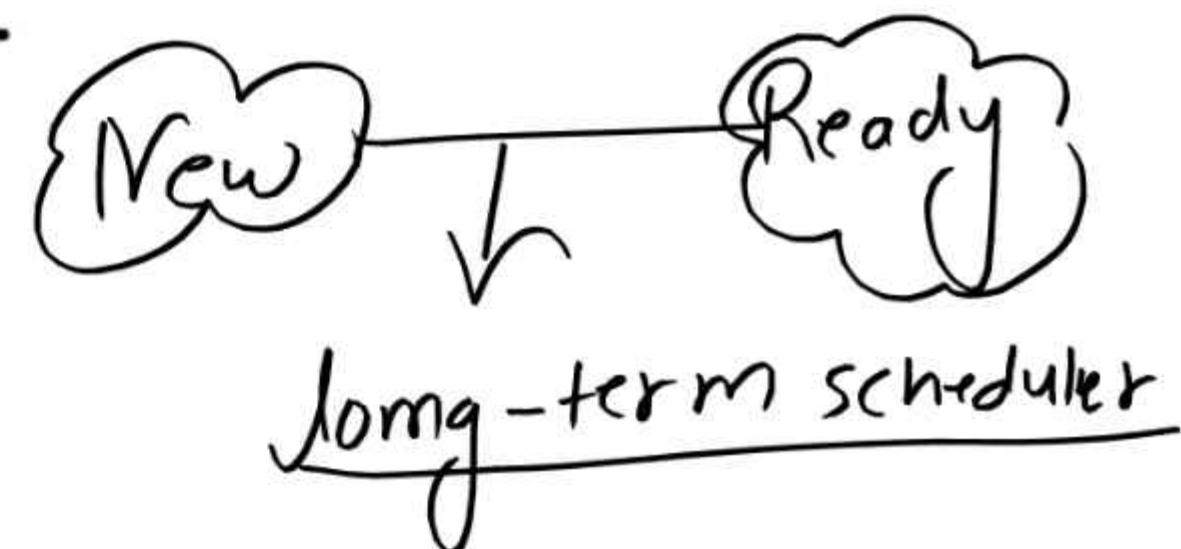
1. Long term Scheduler (job Scheduler).
2. Short term Scheduler (CPU Scheduler).
3. Medium term Scheduler.





## ① Long term Scheduler (job Scheduler):

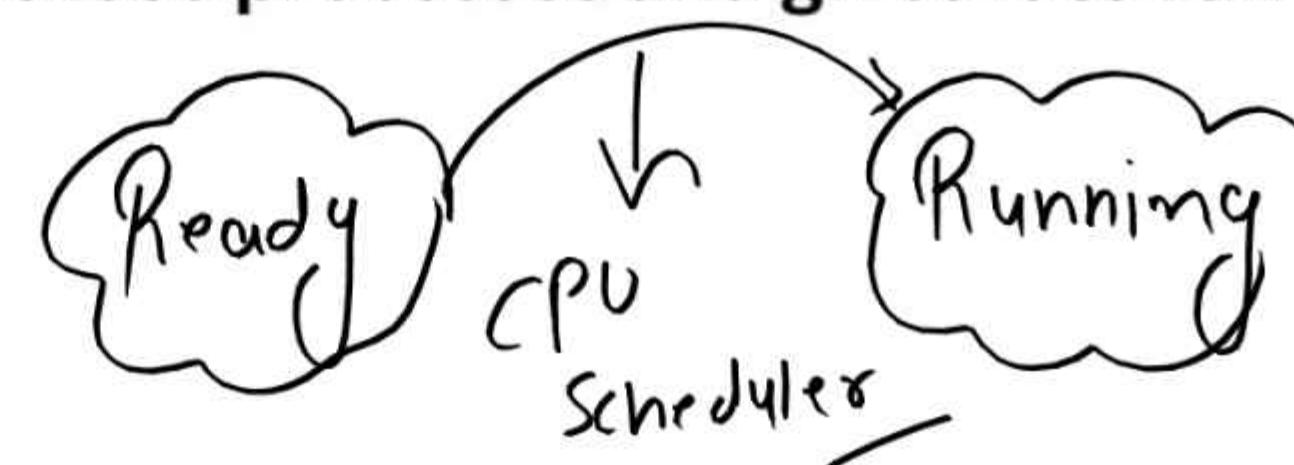
- लॉन्ग-टर्म शेड्यूलर, जिसे जॉब शेड्यूलर भी कहा जाता है, यह तय करता है कि कौन से प्रोग्राम सिस्टम में चलने के लिए आएंगे। यह कतार से कुछ प्रोग्राम चुनता है और उन्हें काम करने के लिए मेमोरी में डालता है। इसके बाद, ये प्रोसेस CPU द्वारा चलाने के लिए तैयार हो जाते हैं।
- The long-term scheduler, also called the job scheduler, decides which programs will come to run in the system. It selects some programs from the queue and puts them in memory to work on. After this, these processes become ready to be run by the CPU.





## 2. Short term Scheduler (CPU Scheduler).

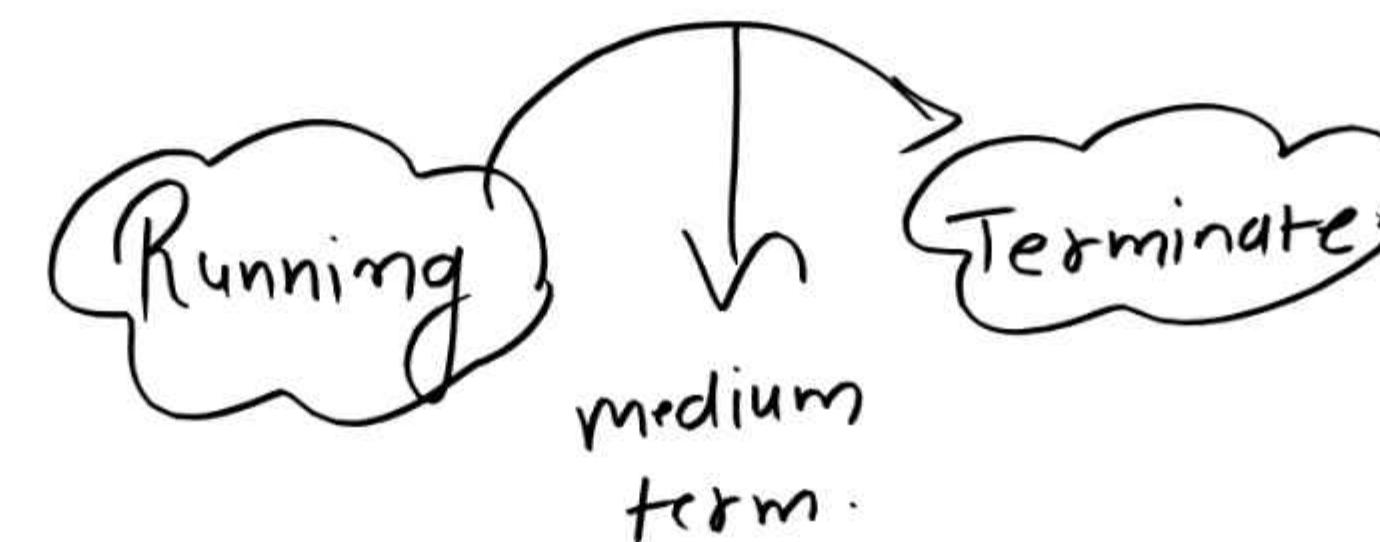
- शॉर्ट-टर्म शेड्यूलर, जिसे CPU शेड्यूलर भी कहते हैं, इसका काम सिस्टम की गति को बढ़ाना है। यह ready प्रोसेस को running प्रोसेस में बदलता है। जब कई प्रोसेस काम करने के लिए तैयार होते हैं, तो CPU शेड्यूलर उनमें से एक प्रोसेस को चुनता है और उसे CPU पर चलने के लिए दे देता है।
- Short-term scheduler, also known as CPU scheduler, its job is to increase the speed of the system. It converts ready processes into running processes. When many processes are ready to work, the CPU scheduler selects one of those processes and gives it to run on the CPU.





### 3. Medium term Scheduler.

- मीडियम-टर्म शेड्यूलर का मुख्य काम **प्रोसेस को मेमोरी से बाहर करना** और उनका प्रबंधन करना होता है, ताकि सिस्टम की परफॉर्मेंस बनी रहे और मेमोरी का उपयोग ठीक से किया जा सके।
- The main job of the medium-term scheduler is to manage and move processes in and out of memory so that system performance is maintained and memory is utilized properly.





## CPU Scheduling Criteria in OS (ऑपरेटिंग सिस्टम में शेड्यूलिंग मापदंड):

### 1. Arival Time (Aruval - टाइम):

- Arrival Time (AT) वह समय होता है जब कोई प्रक्रिया (process) सिस्टम में प्रवेश करती है और ready queue में शामिल होती है ताकि उसे CPU पर चलाया जा सके।
- Arrival Time is the time at which a process enters the ready queue – that is, the moment it becomes ready to be executed by the CPU.

### 2. Burst Time (बीटी - बर्स्ट टाइम): execution time

$$\begin{aligned}P_1 &\rightarrow 10 \text{ ms} \\P_2 &\rightarrow 2 \text{ ms}\end{aligned}$$

- Burst Time वो समय होता है जितने समय तक कोई प्रक्रिया (Process) CPU पर लगातार चलती है जब उसे CPU मिल जाता है।
- Burst Time is the total time required by a process to complete its execution on the CPU without interruption.



### 3. Completion Time (सीटी - कम्प्लीशन टाइम):

- Completion Time वह समय होता है जब कोई प्रक्रिया (Process) पूरी तरह से समाप्त हो जाती है।
- Completion Time is the time at which a process finishes its execution completely.

$$\begin{array}{l} P_1 \rightarrow 2 \text{ ms} \\ 3 \text{ ms} \rightarrow \text{wait} \end{array} \left. \begin{array}{l} \\ \} \end{array} \right\} 3+2 \Rightarrow 5 \text{ ms}$$

### 4. Turnaround Time (टीएटी - टर्नआराउंड टाइम):

- Turnaround Time (TAT) वह कुल समय होता है जो किसी प्रक्रिया (process) को ready queue में आने (arrival) से लेकर पूरी तरह से समाप्त होने (completion) तक लगता है।
- Turnaround Time is the total time taken by a process from the moment it enters the ready queue (arrives) until it completes its execution.

Formula:

$$TAT = \text{Completion Time} - \text{Arrival Time}$$



### 5. Waiting Time (डब्ल्यूटी - वेटिंग टाइम):

- Waiting Time वह समय होता है जब कोई प्रक्रिया Ready Queue में CPU का इंतजार कर रही होती है।
- Waiting Time is the total time a process has to wait in the ready queue without using the CPU.

Formula:

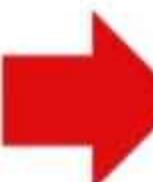
$$WT = TAT - BT$$

### 6: Response Time (RT)

- Response Time वह समय होता है जब कोई प्रक्रिया आई (Arrival Time) और जब उसने पहली बार CPU पर काम शुरू किया (Start Time) उस बीच का अंतर।
- Response Time is the time from arrival of a process to the first time it gets CPU (starts execution).

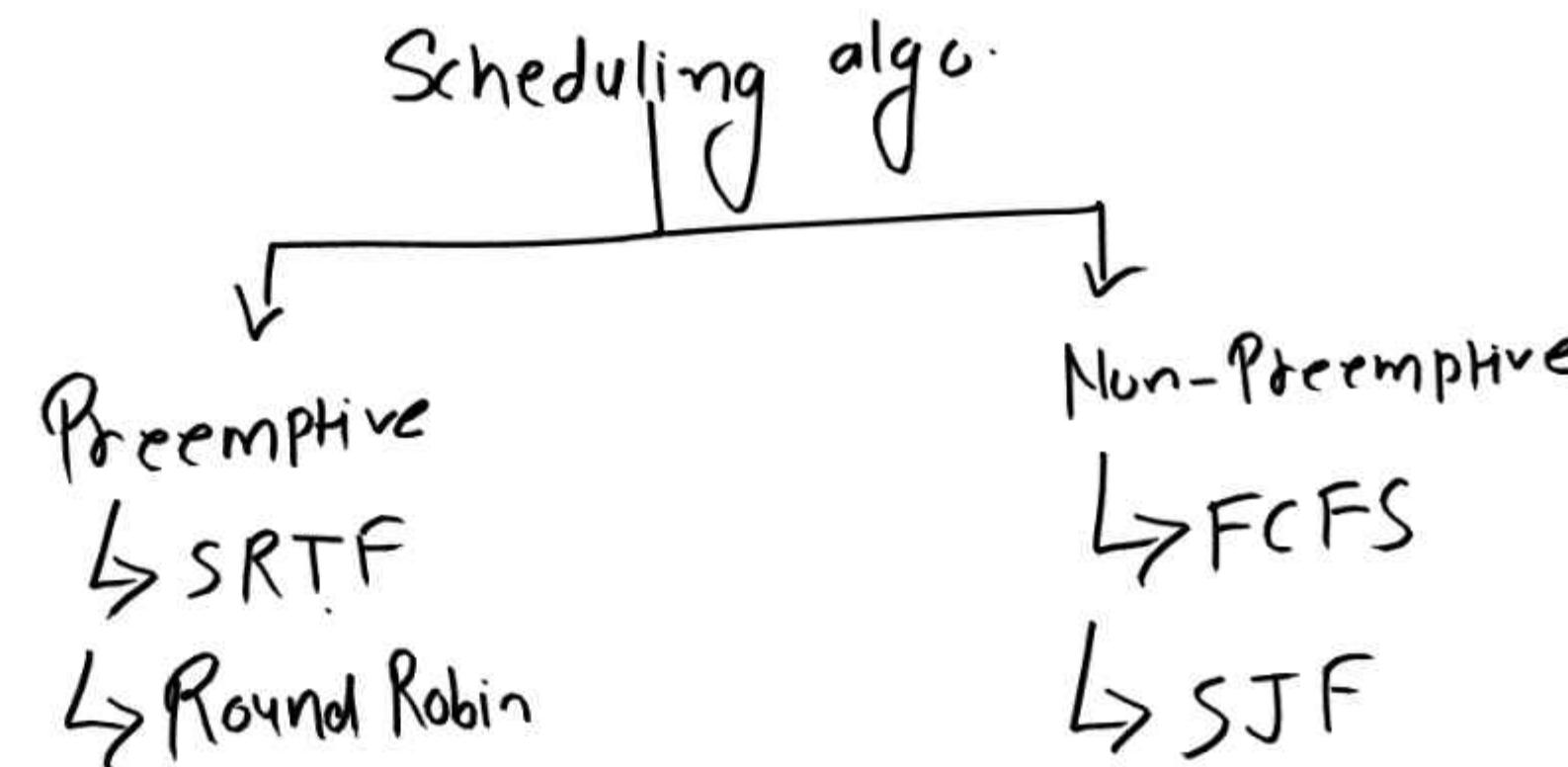
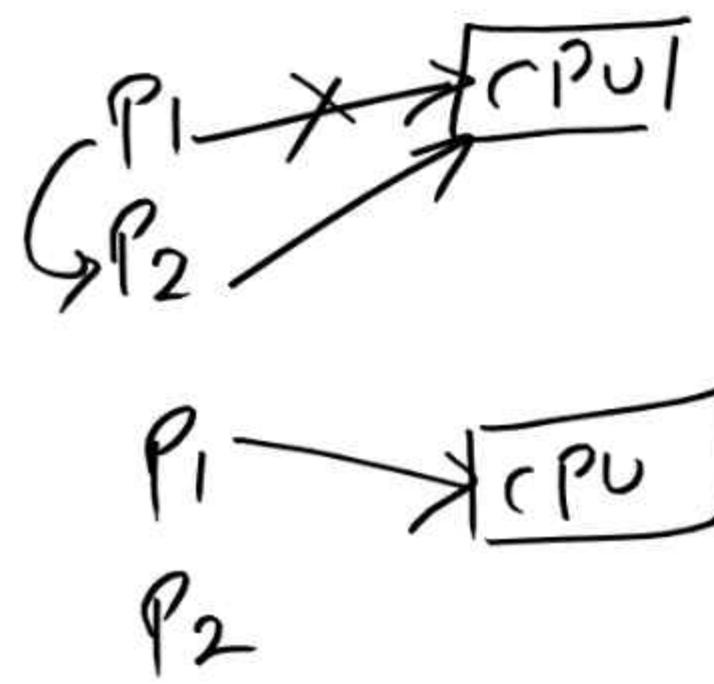
Formula:

$$RT = \text{first Time CPU allocation} - \text{Arrival Time}$$



## CPU Scheduling:

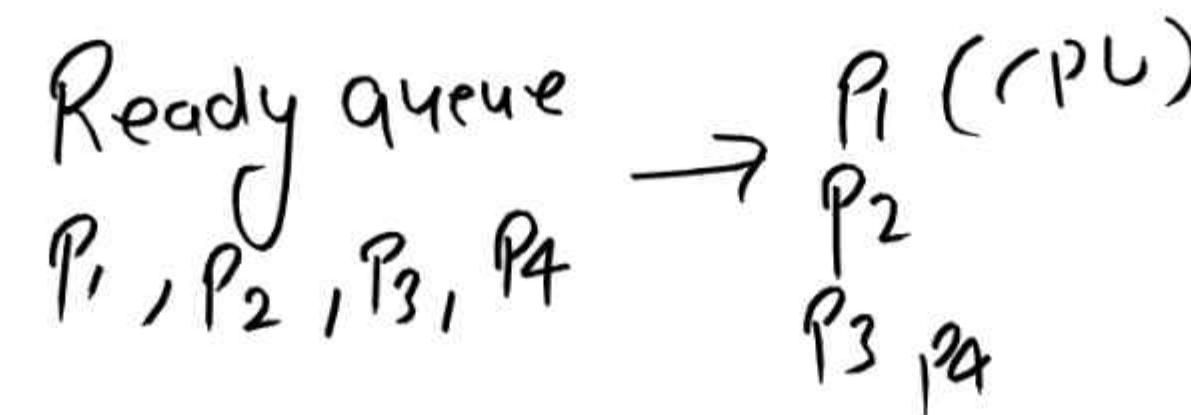
- CPU Scheduling एक प्रक्रिया है जिसके द्वारा Operating System यह तय करता है कि कौन सा Process (प्रक्रिया) CPU को कब और कितनी देर तक उपयोग करेगा।
- CPU scheduling is the process by which the operating system decides which process will use the CPU, when and for how long.





## → FCFS शेड्यूलिंग (First Come First Serve):

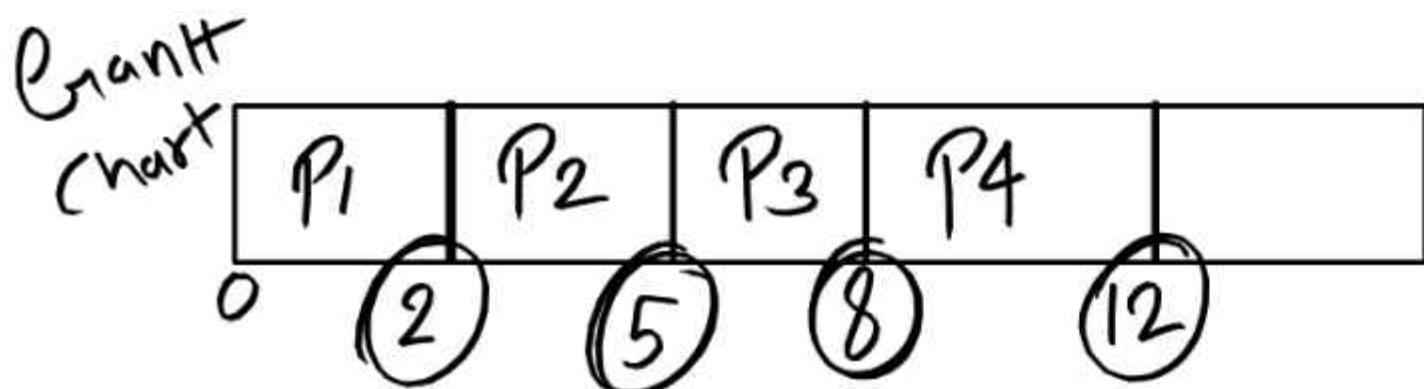
- FCFS (First Come, First Serve) एक CPU शेड्यूलिंग एल्गोरिदम है, जिसमें जो प्रोसेस सबसे पहले आता है, उसे सबसे पहले CPU मिलता है।
- FCFS (First Come, First Serve) is a CPU scheduling algorithm in which the process that comes first gets the CPU first.
- यह गैर-पूर्वग्रहणीय (Non-Preemptive) शेड्यूलिंग एल्गोरिदम है, जिसका मतलब है कि एक बार कोई प्रोसेस CPU ले ले, तो वह पूरा होने के बाद ही CPU छोड़ेगा।
- It is a non-preemptive scheduling algorithm which means once a process takes the CPU, it will leave the CPU only after completion.





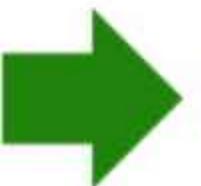
## → FCFS शेड्यूलिंग (First Come First Serve):

Process	Arrival time	Burst time	Complete time	TAT $(CP - AT)$	Waiting time $TA - BT$
P1	0	2	2	$2 - 0 = 2$	$2 - 2 = 0$
P2	1	3	5	$5 - 1 = 4$	$4 - 3 = 1$
P3	2	3	8	$8 - 2 = 6$	$6 - 3 = 3$
P4	5	4	12	$12 - 5 = 7$	$7 - 4 = 3$



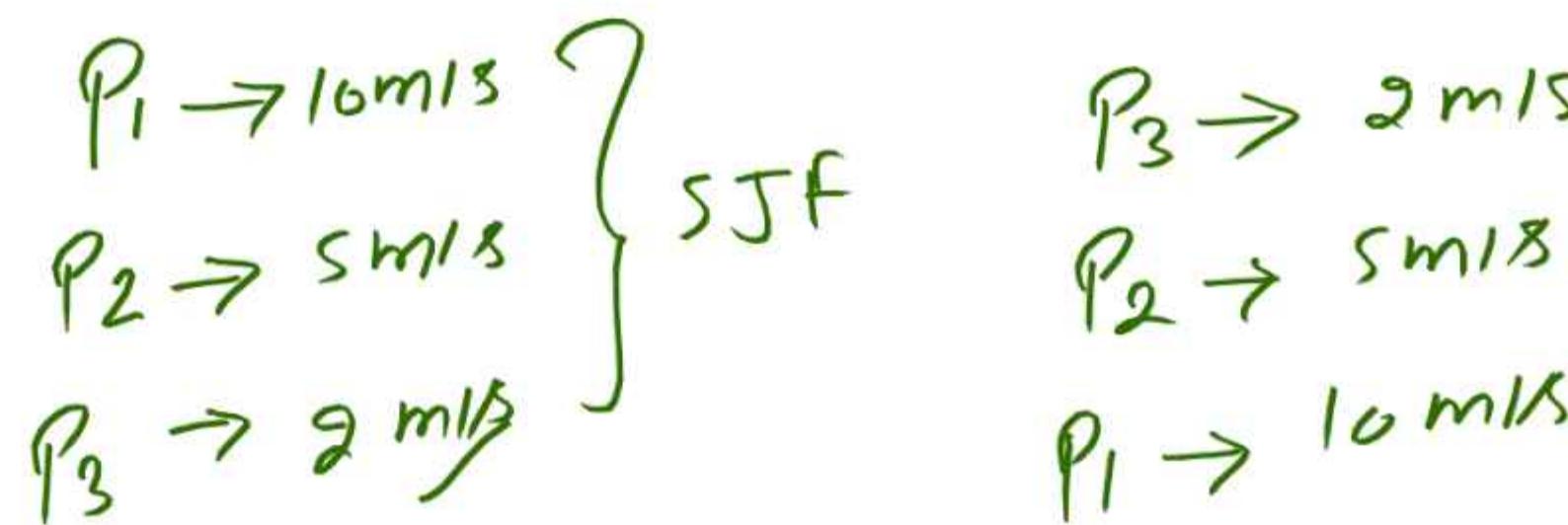
$$ATAT = \frac{2+4+6+7}{4} = \frac{19}{4} = 4.75 \text{ ms}$$

$$AWT = \frac{0+1+3+3}{4} = \frac{7}{4} = 1.75 \text{ ms}$$



## Shortest Job First (SJF) Scheduling Algorithm:

- Shortest Job First (SJF) एक CPU scheduling एल्गोरिदम है जिसमें सबसे कम burst time वाले process को पहले execute किया जाता है। इसे Shortest Next CPU Burst या Shortest Job Next (SJN) भी कहा जाता है।
- Shortest Job First (SJF) is a CPU scheduling algorithm in which the process with the shortest burst time is executed first. It is also called Shortest Next CPU Burst or Shortest Job Next (SJN).

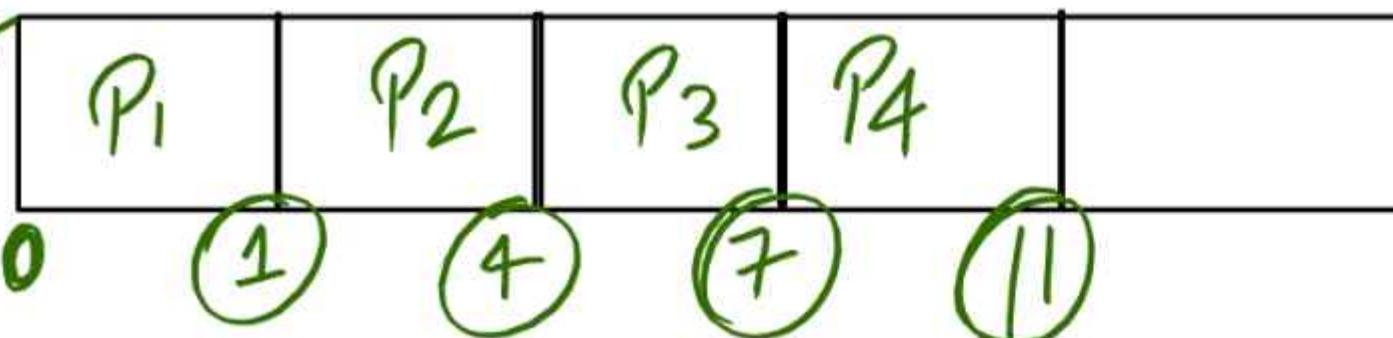




## Shortest Job First (SJF) Scheduling Algorithm:

Process	Arrival time	Burst time	Complete time	TAT	Waiting time
				$CT - AT$	$TAT - BT$
P1	0	1	1	$1 - 0 = 1$	$1 - 1 = 0$
P2	1	3	4	$4 - 1 = 3$	$3 - 3 = 0$
P3	2	3	7	$7 - 2 = 5$	$5 - 3 = 2$
P4	2	4	11	$11 - 2 = 9$	$9 - 4 = 5$

Giantt chart



$$\begin{aligned}ATAT &= \frac{1+3+5+9}{4} \\&= \frac{18}{4} \\&= 4.5 \text{ ms}\end{aligned}$$

$$\begin{aligned}AWT &= \frac{0+0+2+5}{4} \\&= \frac{7}{4} \\&\Rightarrow 1.75 \text{ ms}\end{aligned}$$



## → Shortest Remaining Time First (SRTF):

- Preemptive SJF, जिसे Shortest Remaining Time First (SRTF) भी कहा जाता है, एक CPU scheduling algorithm है जिसमें सबसे कम बचा हुआ execution time (burst time) वाला process पहले execute होता है।
- Preemptive SJF, also called Shortest Remaining Time First (SRTF), is a CPU scheduling algorithm in which the process with the shortest remaining execution time (burst time) is executed first.
- यदि कोई नया process आता है और उसका burst time वर्तमान process से कम है, तो CPU उसे दे दिया जाता है, और पहले वाला process रुक जाता है (preempt हो जाता है)।
- If a new process arrives and its burst time is less than the current process, then the CPU is given to it, and the previous process is stopped (preempted).

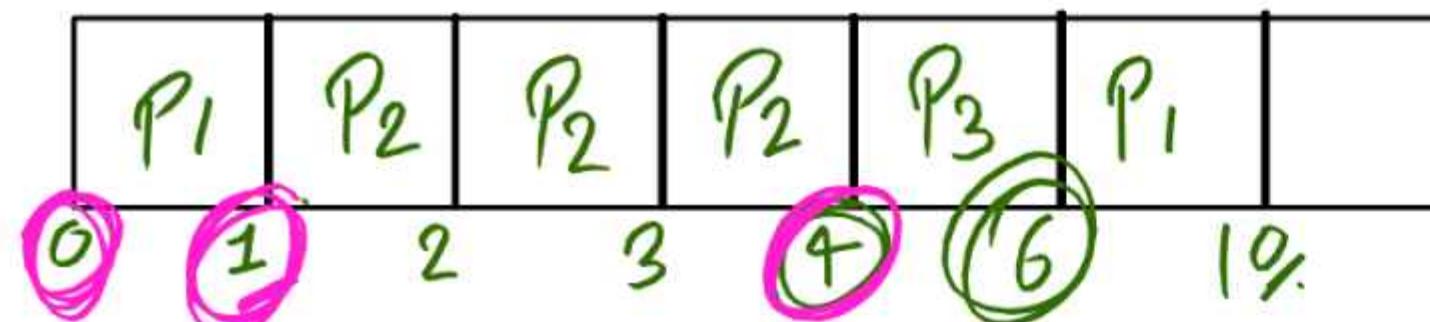
$P_1 \rightarrow \text{exec. (stop)}$

$P_2 \rightarrow (\text{exec})$



## Shortest Remaining Time First (SRTF):

Process	Arrival time	Burst time	Complete time	$\frac{TAT}{CT-AT}$	Waiting time	RS $FA-AT$
P1	0	5 4	10	$10-0=10$	$10-5=5$	$0-0=0$
(X) P2	1	3 2 10	4	$4-1=3$	$3-3=0$	$1-1=0$
(X) P3	2	2	6	$6-2=4$	$4-2=2$	$4-2=2$



$$ATAT = \frac{10+3+4}{3}$$

$$= \frac{17}{3}$$

$$\Rightarrow 5.66 \text{ ms}$$

$$AWT = \frac{5+0+2}{3}$$

$$\Rightarrow \frac{7}{3}$$

$$\Rightarrow 2.33 \text{ ms}$$



## Round Robin Scheduling Algorithm :

- Round Robin (RR) एक CPU शेड्यूलिंग एल्गोरिथम है जिसमें प्रत्येक प्रोसेस को एक निश्चित समय (Time Quantum) के लिए CPU दिया जाता है। यदि प्रोसेस उस समय में पूरा नहीं होता, तो उसे कतार के अंत में भेज दिया जाता है और अगला प्रोसेस CPU प्राप्त करता है।
- Round Robin (RR) is a CPU scheduling algorithm in which each process is given a CPU for a fixed time (Time Quantum). If the process does not complete in that time, it is sent to the end of the queue and the next process gets the CPU.

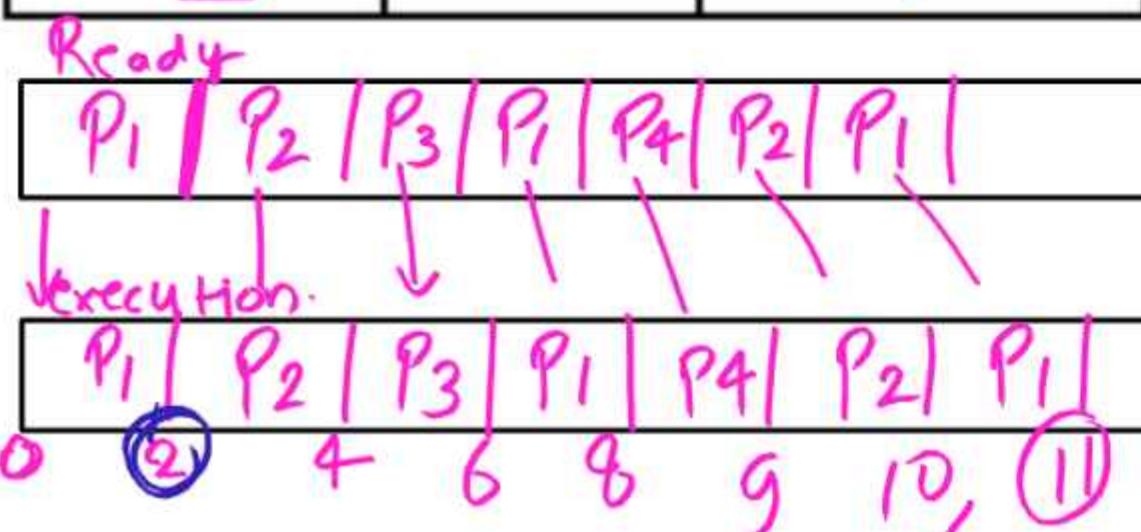
$$\begin{array}{l} 10m/s \leftarrow P_1 \\ 4m/s \leftarrow P_2 \\ 2m/s \leftarrow P_3 \end{array} \quad \left. \begin{array}{c} 2 \\ 2 \\ 2 \end{array} \right\} TQ = 2$$



## Round Robin Scheduling Algorithm:

$$T \text{ (Time Quantum)} = \underline{\underline{2}}$$

Process	Arrival time	Burst time	Complete time	TAT $(CT - AT)$	Waiting time $TA - BT$	RS
P1 ✓	0	5 ✓ 8 1	11	11 - 0 = 11	11 - 5 = 6	0 - 0 = 0
X P2	1	3 ✓ 1	10	10 - 1 = 9	9 - 3 = 6	2 - 1 = 1
X P3	2	2 - 2 = 0	6	6 - 2 = 4	4 - 2 = 2	4 - 2 = 2
X P4	4	1	9	9 - 4 = 5	5 - 1 = 4	8 - 4 = 4



$$ATAT = \frac{11 + 9 + 4 + 5}{4} = \frac{29}{4} = 7.25 \text{ ms}$$

$$AWT = \frac{6 + 6 + 2 + 4}{4} = \frac{18}{4} = 4.5 \text{ ms}$$



## Difference Between Preemptive and Non-Preemptive Scheduling:

विशेषता / Feature	Preemptive Scheduling	Non-Preemptive Scheduling
1. CPU Control	CPU एक process से छीना जा सकता है	CPU एक बार मिलने के बाद process को पूरा चलने दिया जाता है
2. Response Time	कम होता है (Better for interactive systems)	ज्यादा हो सकता है
3. Complexity	थोड़ी <b>complex</b> होती है implement करना	<b>Simple</b> होती है implement करना
4. Throughput	ज्यादा (क्योंकि idle time कम होता है)	थोड़ा कम
5. Process Switch (Context Switch)	बार-बार switching होता है (Overhead ज्यादा)	कम switching होता है (Overhead कम)
7: Examples	Round Robin, SRTF, Preemptive Priority	FCFS, SJF (non-preemptive), Non-preemptive Priority

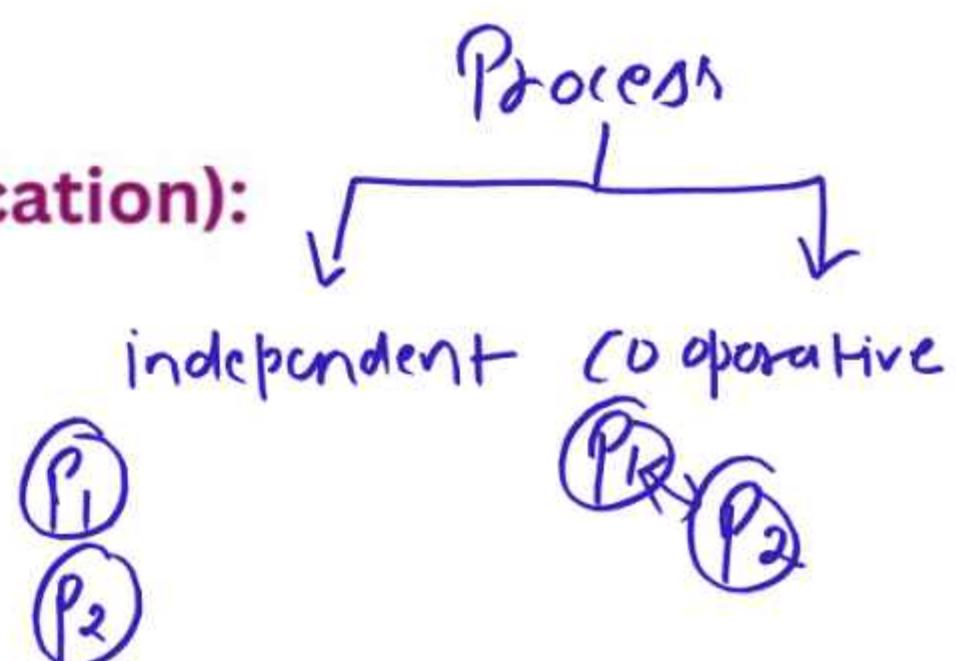


## Inter process Communication:

- इंटरप्रोसेस कम्युनिकेशन (IPC) वह तकनीक है जिसके माध्यम से एक प्रोसेस (Process) दूसरे प्रोसेस के साथ डेटा साझा (Data Sharing) या संदेश भेजने (Message Passing) का कार्य कर सकता है। जब एक ही ऑपरेटिंग सिस्टम में कई प्रोसेस एक साथ चलते हैं, तो उन्हें कभी-कभी एक-दूसरे से संपर्क (Communication) करने की जरूरत होती है। यही कार्य IPC के माध्यम से किया जाता है
- Interprocess communication (IPC) is a technique through which a process can share data or send messages with another process. When many processes run simultaneously in the same operating system, they sometimes need to communicate with each other. This is done through IPC.

### इंटरप्रोसेस कम्युनिकेशन के प्रकार (Types of Interprocess Communication):

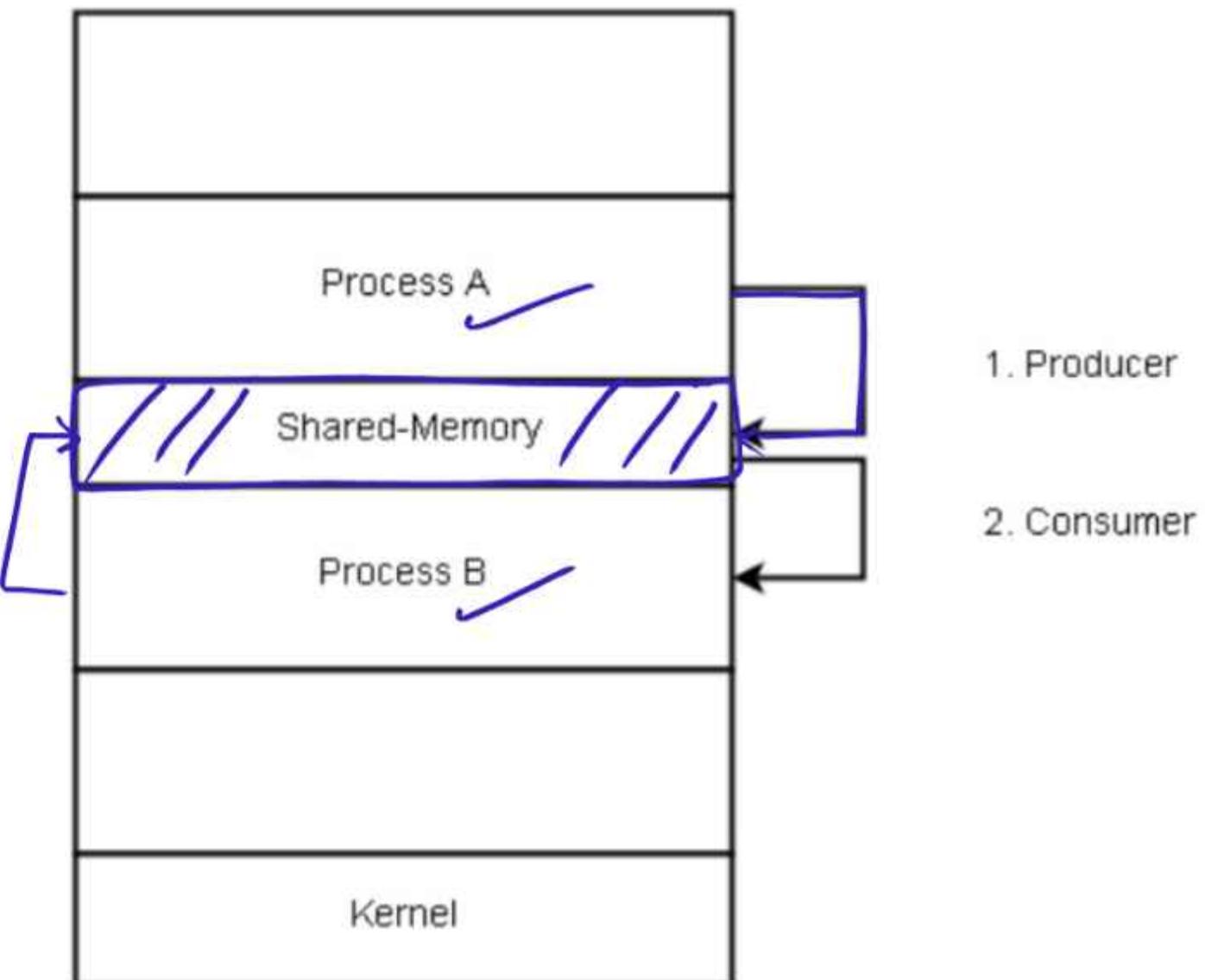
- शेयर्ड मेमोरी (Shared Memory)
- मैसेज पासिंग (Message Passing)





## 1. शेयर्ड मेमोरी (Shared Memory)

- इसमें दो या अधिक प्रोसेस एक साझा (शेयर्ड) मेमोरी एरिया का उपयोग करते हैं, जहां वे डेटा पढ़ और लिख सकते हैं।
- In this two or more processes use a shared memory area where they can read and write data.
- उदाहरण:** एक नोटपैड की तरह जहाँ दो लोग एक ही फाइल में कुछ लिख सकते हैं और पढ़ सकते हैं।
- Like a notepad where two people can write and read something in the same file.



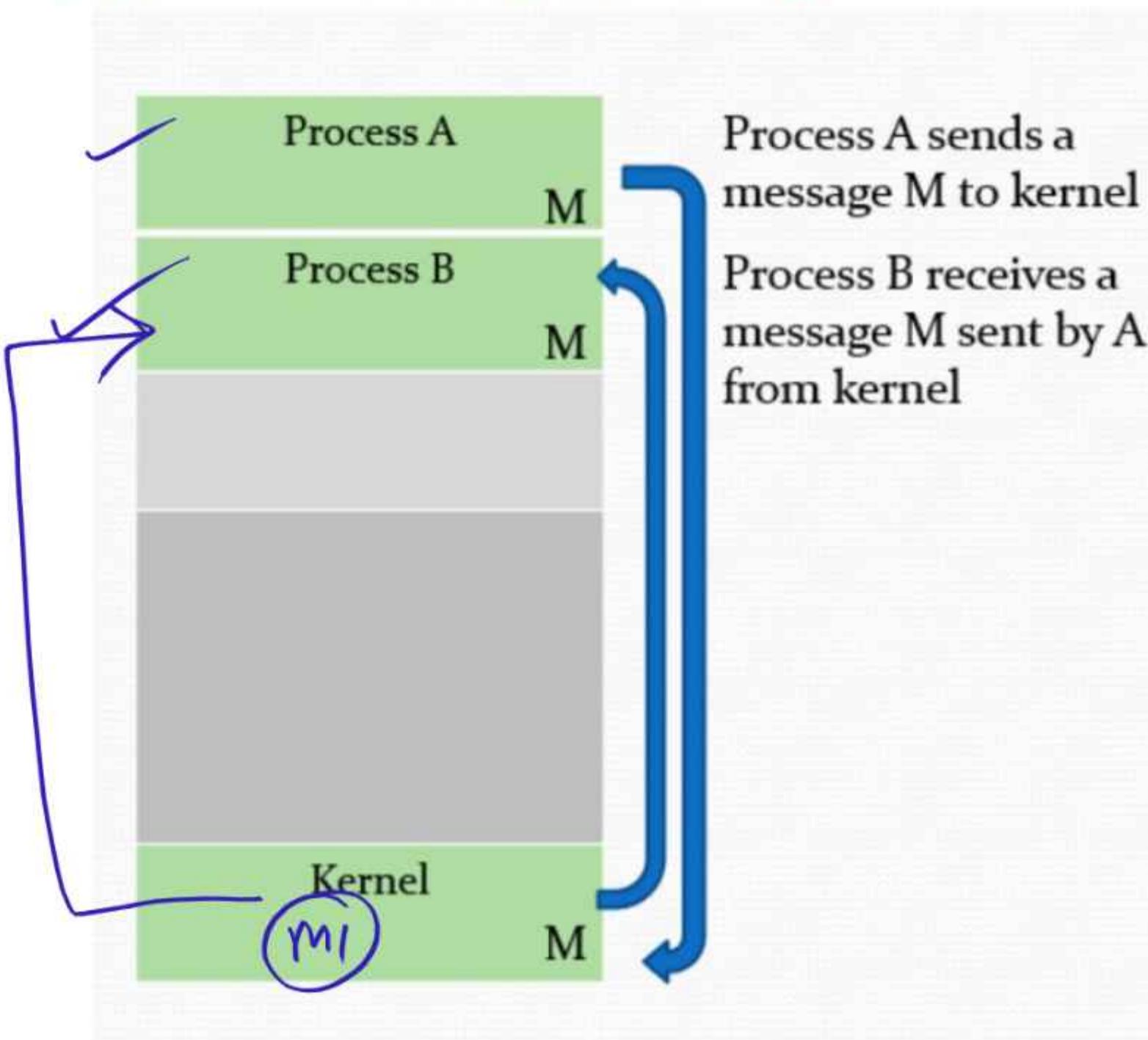


## 2: मैसेज पासिंग के प्रकार (Types of message passing):

- मैसेज पासिंग एक तरीका है जिससे दो या अधिक प्रोसेस (Processes) आपस में संदेश (Message) भेज और प्राप्त कर सकते हैं। जब दो प्रोसेस को एक-दूसरे से बात करनी होती है लेकिन उनके पास साझा मेमोरी (Shared Memory) नहीं होती, तब वे मैसेज पासिंग का उपयोग करते हैं।
- Message passing is a way that two or more processes can send and receive messages between each other. When two processes need to talk to each other but they do not have shared memory, they use message passing.



## 2: मैसेज पासिंग के प्रकार (Types of message passing):

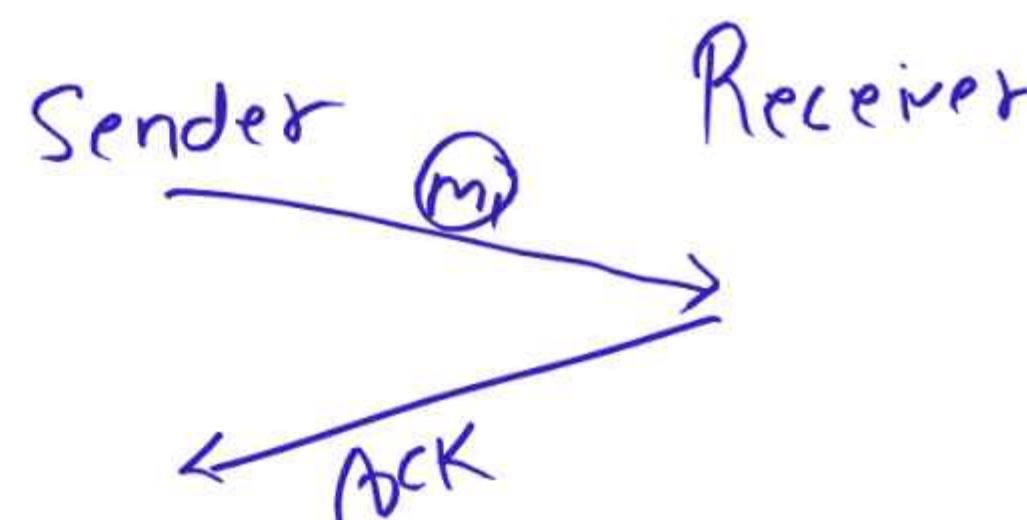




मैसेज पासिंग के प्रकार (Types of message passing):

### 1. सिंक्रोनस (Synchronous Message Passing):

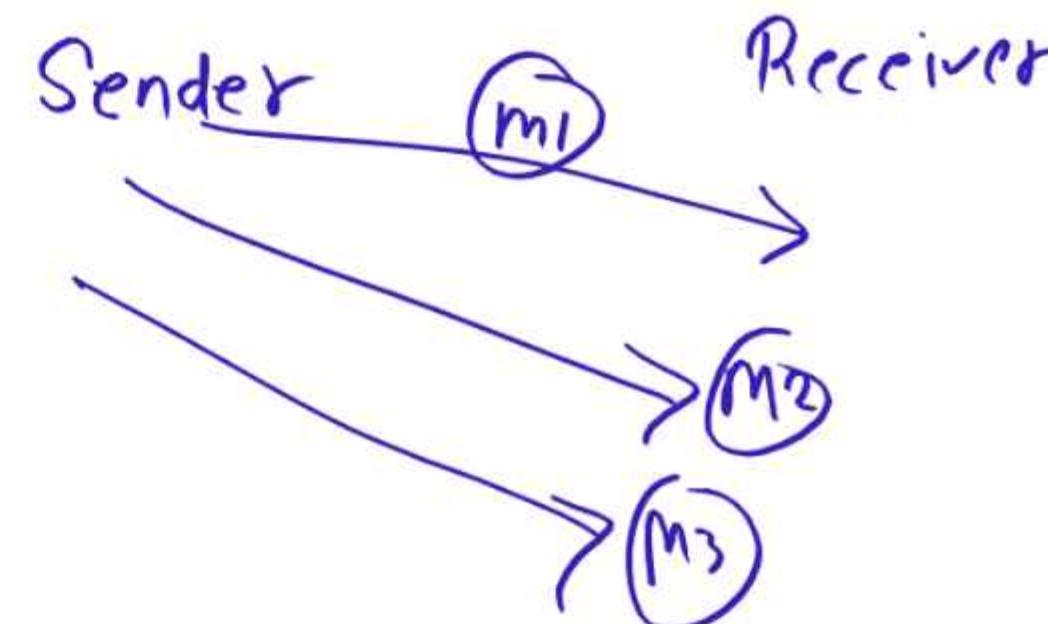
- जब भेजने वाला प्रोसेस संदेश भेजता है, तो उसे तब तक इंतजार करना पड़ता है जब तक कि प्राप्त करने वाला प्रोसेस उसे स्वीकार (Receive) न कर ले।
- When the sending process sends a message, it has to wait until the receiving process receives it.
- यह धीमा (Slow) होता है लेकिन सुरक्षित (Safe) होता है।
- It is slow but safe.





## 2. Asynchronous Message Passing :

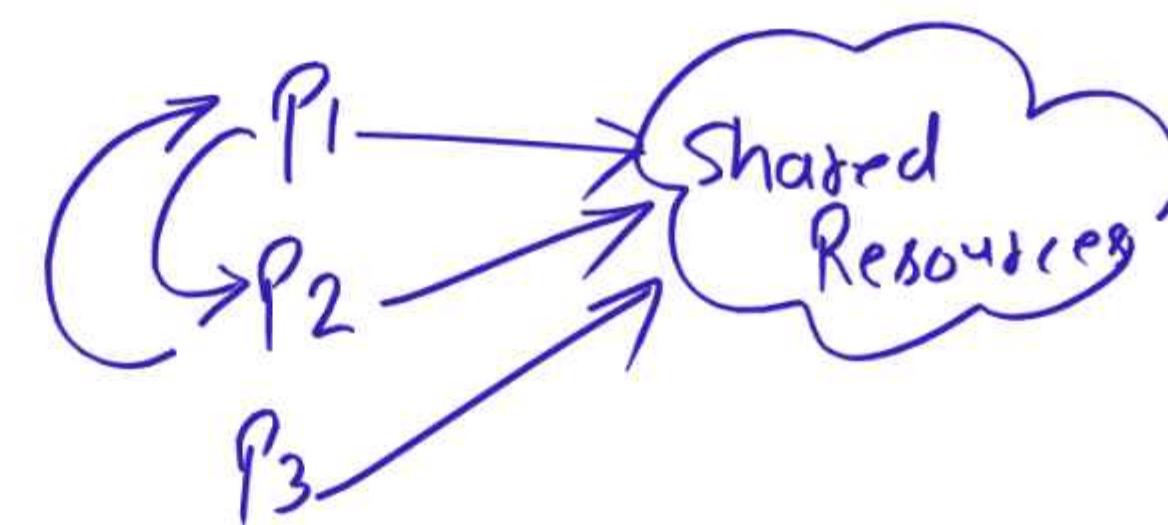
- एक संचार तकनीक है जिसमें दो प्रोसेस (या थ्रेड) एक-दूसरे को मैसेज भेजते हैं, लेकिन उन्हें तुरंत जवाब का इंतजार नहीं करना पड़ता। इसका मतलब है कि भेजने वाला (sender) अपना काम जारी रख सकता है, जबकि प्राप्त करने वाला (receiver) जब समय मिले, तब मैसेज को प्रोसेस कर सकता है।
- A communication technique in which two processes (or threads) send messages to each other, but do not have to wait for an immediate response. This means that the sender can continue with its work, while the receiver can process the message when it has time.





## प्रोसेस सिंक्रोनाइज़ेशन (Process Synchronization):

- जब एक ही समय पर कई प्रोसेस (कार्य) एक ही रिसोर्स (संसाधन) को एक्सेस करने की कोशिश करते हैं, तो सही समन्वय (synchronization) होना बहुत जरूरी होता है ताकि डेटा गड़बड़ न हो और सिस्टम सही से काम करे। प्रोसेस सिंक्रोनाइज़ेशन इसी समस्या का हल निकालने की प्रक्रिया है।
- When multiple processes (tasks) try to access the same resource at the same time, proper synchronization is very important so that the data is not corrupted and the system works properly. Process synchronization is the process to solve this problem.





## प्रोसेस सिंक्रोनाइज़ेशन के समाधान (Solutions for Process Synchronization):

### 1: म्यूटेक्स (Mutex - Mutual Exclusion):

- यह एक लॉक की तरह काम करता है। जब एक प्रोसेस डेटा एक्सेस कर रहा होता है, तो दूसरा तब तक इंतजार करता है जब तक पहला प्रोसेस पूरा नहीं हो जाता।
- It acts like a lock. When one process is accessing data, the other waits until the first process is completed.



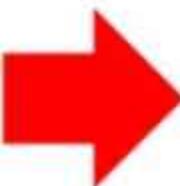
### 2. सेमाफोर (Semaphore):

- यह म्यूटेक्स जैसा ही होता है, लेकिन इसमें एक काउंटर होता है जो बताता है कि कितने प्रोसेस एक साथ चल सकते हैं।
- It is similar to a mutex, but has a counter that indicates how many processes can run simultaneously.



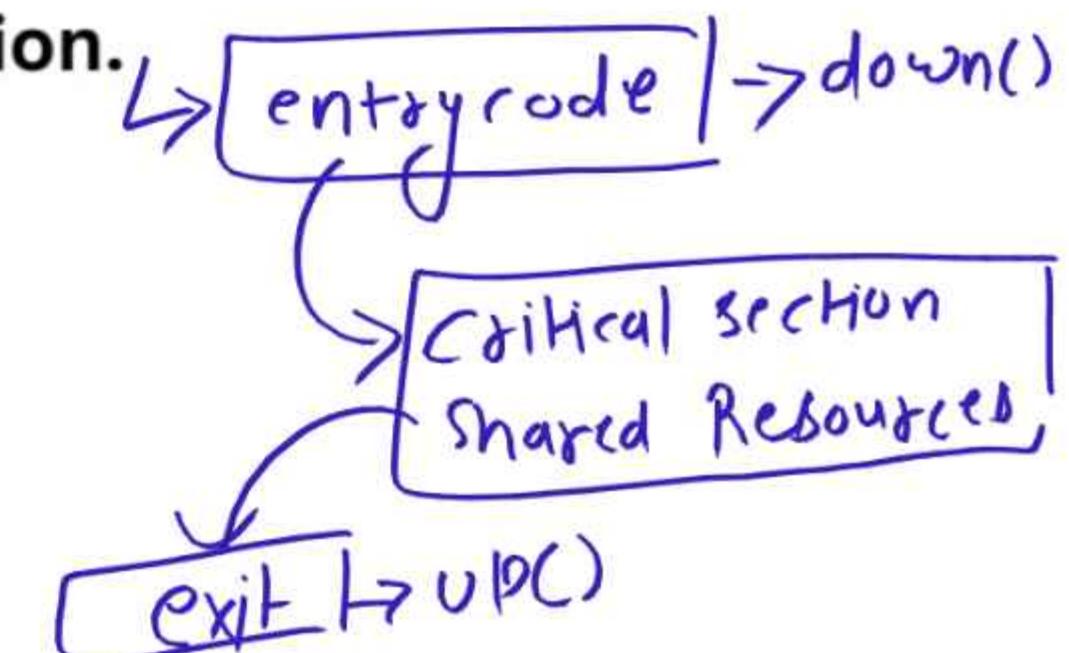
### 3. मॉनिटर (Monitor):

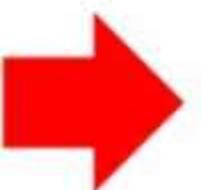
- यह एक हाई-लेवल मैनेजमेंट सिस्टम है, जो म्यूटेक्स और सेमाफोर को मैनेज करता है और प्रोसेस को नियंत्रित करता है।
- It is a high-level management system that manages mutexes and semaphores and controls the process.



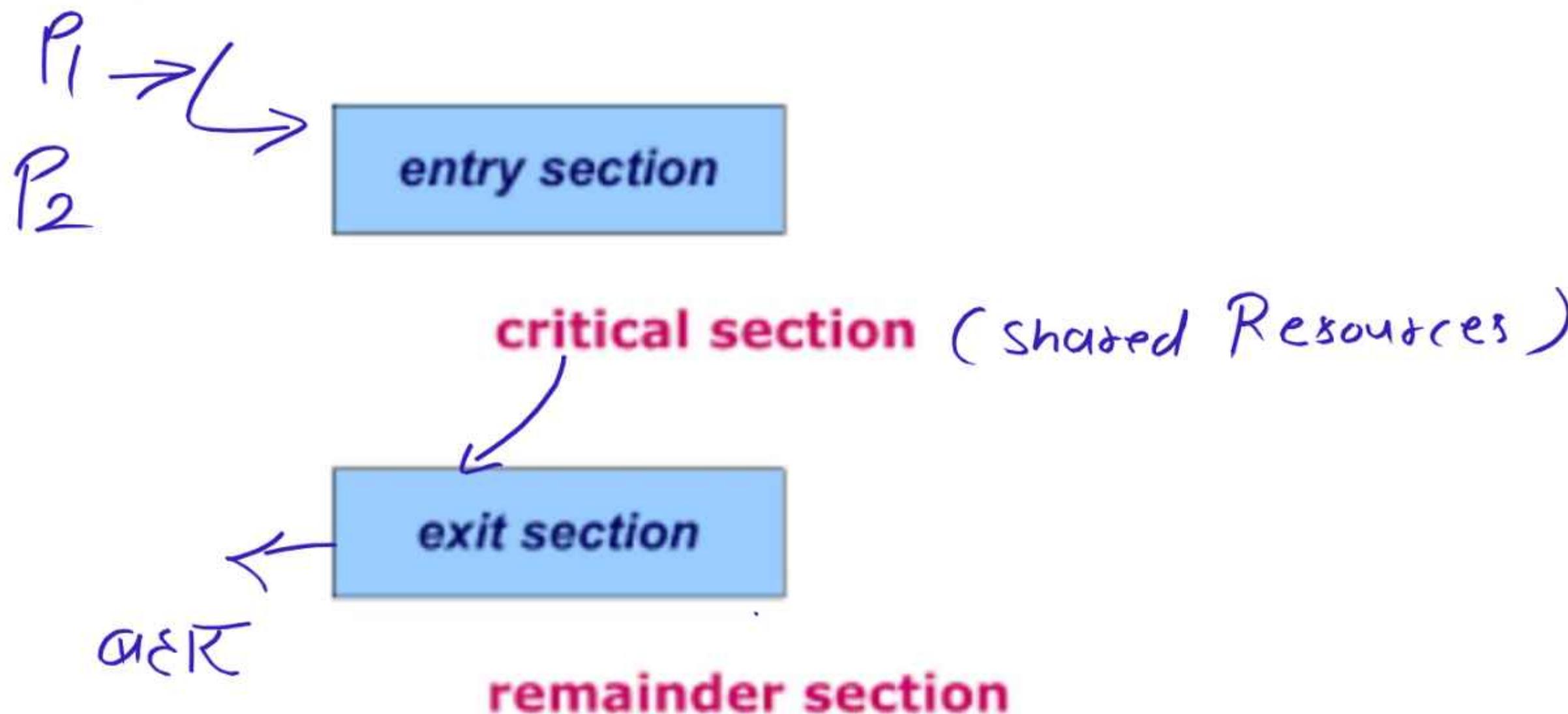
## Critical Section:

- Critical Section प्रोग्राम का वह भाग होता है जहाँ पर कोई प्रक्रिया (process) shared resources (जैसे – डेटा, फ़ाइल, या मेमोरी) को उपयोग करती है।
- A Critical Section is a part of a program where shared resources (like variables, files, or memory) are accessed.
- अगर एक से ज़्यादा process एक ही समय पर Critical Section में पहुँच जाती हैं, तो डेटा गड़बड़ (data inconsistency) या race condition हो सकती है।
- If multiple processes try to enter their critical sections at the same time, it may cause data inconsistency or race condition.





## Critical Section:





## Solution Techniques to the Critical Section Problem (समाधान की विधियाँ):

### 1. Peterson's Algorithm:

- Peterson's Algorithm एक software आधारित समाधान है जो दो प्रक्रियाओं (processes) के बीच critical section को सुरक्षित तरीके से शेयर करने देता है। इसमें दो variables – flag[] और turn का उपयोग होता है।
- Peterson's Algorithm is a software-based solution for the critical section problem that allows two processes to share a single-use resource without conflict, using two variables: flag[] and turn.
- **flag[i]** – बताता है कि Process-i को critical section में जाना है।
- **flag[i]** – Indicates that Process-i has to enter the critical section.
- **turn** – यह तय करता है कि किसकी बारी है।
- **turn** – Determines whose turn it is.



## 2. Semaphore:

- सेमाफोर एक समन्वयन (synchronization) उपकरण है, जिसका उपयोग क्रिटिकल सेक्शन समस्या को हल करने के लिए किया जाता है। यह साझा संसाधन (shared resource) तक पहुँच को नियंत्रित करने के लिए दो परमाणु क्रियाओं (atomic operations) – `wait()` और `signal()` – का उपयोग करता है।
- A semaphore is a synchronization tool that is used to solve the critical section problem by controlling access to the shared resource using two atomic operations: `wait()` and `signal()`.



### 3. Mutex (Mutual Exclusion Object):

- Mutex एक lock आधारित synchronization tool है जो यह सुनिश्चित करता है कि एक समय में केवल एक प्रक्रिया या थ्रेड ही critical section में जा सके।
- Mutex is a locking mechanism used to enforce mutual exclusion. It ensures that only one thread or process can access the critical section at any given time.
- **lock()** – जब कोई thread critical section में प्रवेश करता है।
- when a thread enters the critical section.
- **unlock()** – जब वो critical section से बाहर आता है।
- when it exits the critical section.



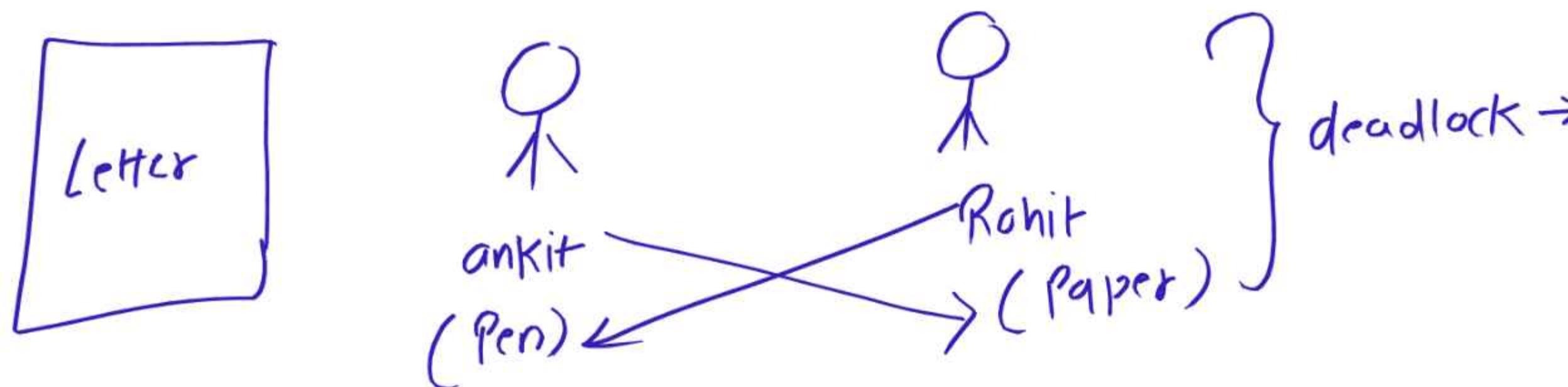
#### 4. Monitors:

- Monitor एक उच्च स्तरीय संरचना (high-level structure) है जिसमें shared variables, functions और condition variables होते हैं जो critical section की सुरक्षा के लिए एक module के रूप में काम करते हैं।
- Monitor is a high-level synchronization construct that combines shared variables, operations, and condition variables into a single module to safely handle the critical section problem.
- Monitor का उपयोग कर processes आपस में synchronized तरीके से shared resources को access करते हैं।
- Using monitors, processes access shared resources in a synchronized manner.



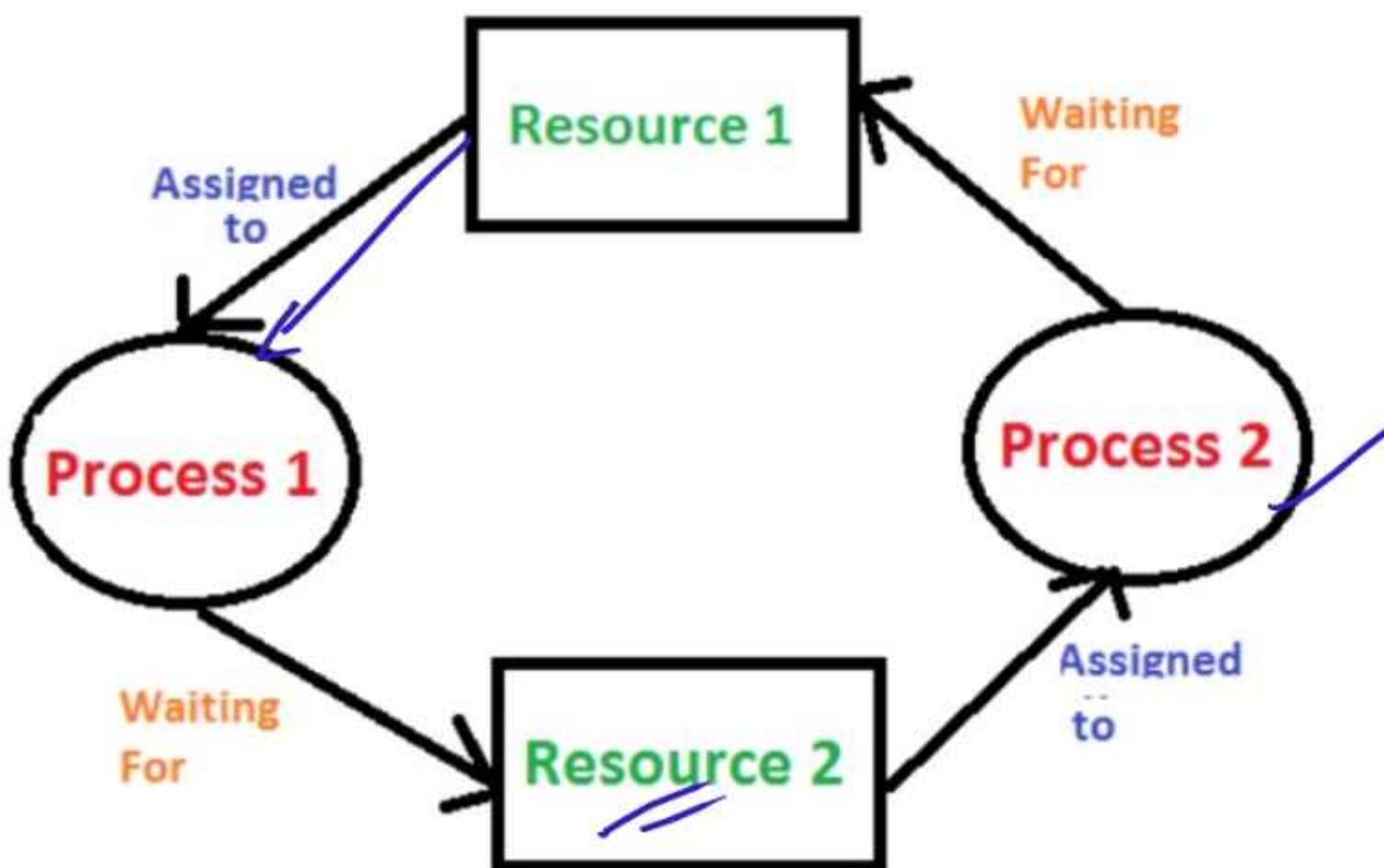
### डेडलॉक (Deadlock):

- डेडलॉक एक ऐसी स्थिति है जिसमें दो या अधिक प्रोसेस एक-दूसरे के संसाधन (resource) को पाने का इंतजार करते रहते हैं, और कोई भी प्रोसेस आगे नहीं बढ़ पाता। इसे “प्रोग्रामिंग जाम” भी कह सकते हैं, क्योंकि यह सिस्टम को रोक देता है।
- Deadlock is a situation in which two or more processes keep waiting for each other's resources, and no process is able to proceed. It can also be called a "programming jam" because it stops the system.





## डेडलॉक (Deadlock):





### डेडलॉक के 4 कंडीशन्स (4 conditions of deadlock):

#### 1. म्युचुअल एक्सक्लूजन (Mutual Exclusion -)

- कोई संसाधन (resource) एक समय में सिर्फ एक ही प्रोसेस के पास हो सकता है।
- A resource can be owned by only one process at a time.
- जैसे, एक ही प्रिंटर पर दो लोग एक साथ प्रिंटिंग नहीं कर सकते।
- For example, two people cannot print simultaneously on the same printer

#### 2. होल्ड एंड वेट (Hold and Wait)

- एक प्रोसेस अपने पास मौजूद संसाधन को छोड़ने के बजाय, और संसाधनों के लिए इंतजार करता है।
- A process waits for more resources, rather than releasing the resources it has.

$P_1 \rightarrow (R_1)$  allocate  
 $P_1 \rightarrow (R_2)$  allocate



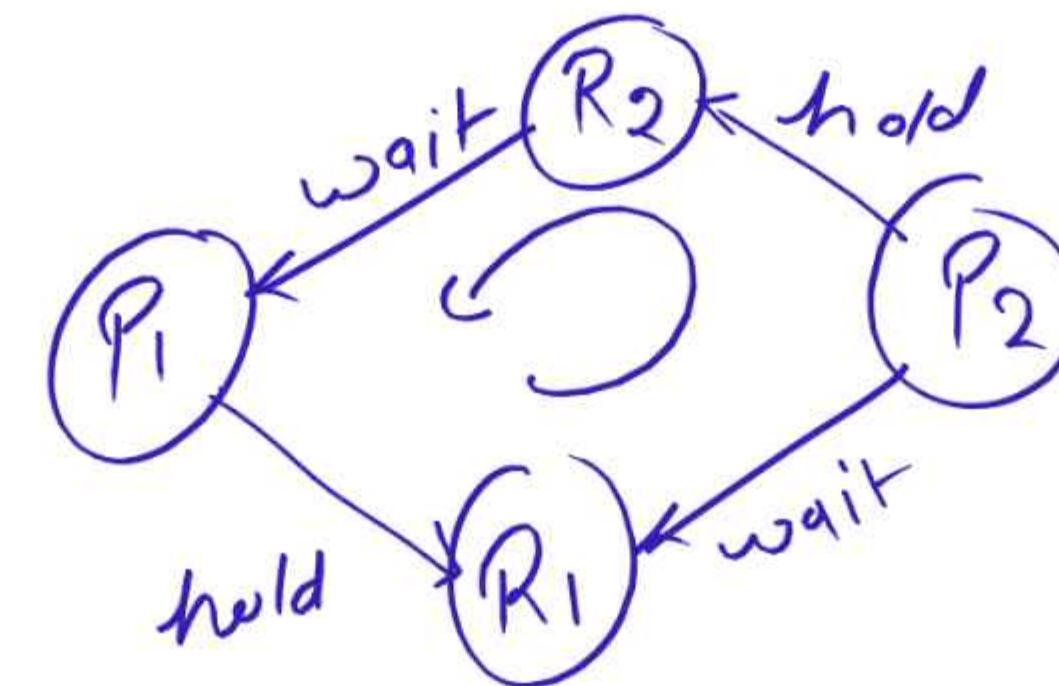
### 3. नो प्रीएम्पशन (No Preemption -)

- कोई प्रोसेस जब तक खुद से संसाधन को छोड़ना न चाहे, कोई दूसरा उसे छीन नहीं सकता।
- A process cannot take over a resource unless it wishes to give it up.



### 4. सर्कुलर वेट (Circular Wait):

- जब कई प्रोसेस एक चक्र (circle) में एक-दूसरे के संसाधन के लिए इंतजार कर रहे होते हैं।
- When multiple processes are waiting for each other's resources in a circle.





## डेडलॉक को हैंडल करने के तरीके (Methods for Handling Deadlocks):

- जब सिस्टम में डेडलॉक (Deadlock) हो जाता है, तो इसे रोकने (Prevent), टालने (Avoid), पता लगाने (Detect) और हल करने (Recover) के कुछ तरीके होते हैं।
- When a deadlock occurs in a system, there are some ways to prevent, avoid, detect, and recover from it.
  - डेडलॉक रोकना (Deadlock Prevention)
  - डेडलॉक टालना (Deadlock Avoidance)
  - डेडलॉक का पता लगाना (Deadlock Detection)
  - डेडलॉक से रिकवरी (Deadlock Recovery)



### 1. डेडलॉक रोकना (Deadlock Prevention):

- इसमें डेडलॉक को होने से पहले ही रोक दिया जाता है। चूंकि डेडलॉक तभी होता है जब उसकी चार शर्तें (Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait) पूरी होती हैं, इसलिए किसी एक शर्त को तोड़कर डेडलॉक को रोका जा सकता है।
- In this, deadlock is prevented before it happens. Since deadlock occurs only when its four conditions (Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait) are fulfilled, deadlock can be prevented by breaking any one of the conditions.



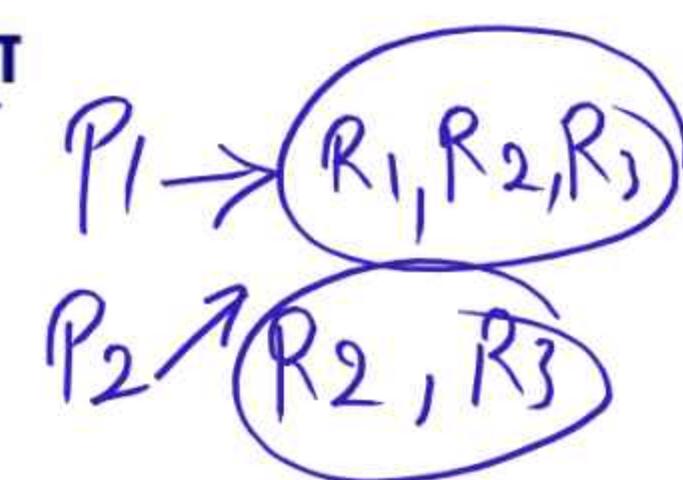
### ◆ Deadlock की चार आवश्यक शर्तें:

- **Mutual Exclusion** - एक समय में एक ही प्रोसेस किसी resource का उपयोग कर सकती है।
- Only one process can use a resource at a time.
- **Hold and Wait** - एक प्रोसेस ने कोई resource पकड़ा है और वह दूसरे के लिए इंतजार कर रही है।
- One process has held a resource and is waiting for another.
- **No Preemption** - एक resource को जबरदस्ती वापस नहीं लिया जा सकता।
- A resource cannot be forcibly taken back.
- **Circular Wait** - प्रोसेस एक दूसरे से resource की चेन में इंतजार कर रही हैं।
- Processes are waiting on each other in a chain for a resource.



## 2. डेडलॉक टालना (Deadlock Avoidance):

- डेडलॉक टालने (Avoidance) का मतलब यह है कि सिस्टम पहले से ही यह जांच कर ले कि संसाधन (Resource) देने से डेडलॉक होगा या नहीं। अगर डेडलॉक होने की संभावना है, तो सिस्टम उस प्रोसेस को संसाधन नहीं देता और उसे इंतजार कराता है।
- Deadlock avoidance means that the system checks in advance whether giving resources will cause deadlock or not. If there is a possibility of deadlock, then the system does not give resources to that process and makes it wait.
- डेडलॉक टालने के लिए सबसे ज्यादा बैंकर्स एल्गोरिदम (Banker's Algorithm) का इस्तेमाल किया जाता है।
- Banker's Algorithm is most commonly used to avoid deadlock.





### Banker's Algorithm is most commonly used to avoid deadlock:

- इसमें हर process को पहले ही बताना होता है कि उसे अधिकतम कितने resource चाहिए होंगे। System तब चेक करता है कि क्या सभी process को उनकी जरूरत के अनुसार future में resource allocate किए जा सकेंगे।
- अगर answer “yes” है, तो request allow की जाती है; अन्यथा reject या delay कर दी जाती है।
- In this, every process has to state in advance the maximum amount of resources it will need. The system then checks whether resources can be allocated to all processes as per their requirement in future.
- If the answer is “yes”, then the request is allowed; otherwise it is rejected or delayed.

$$\left\{ \begin{array}{l} P_1 \rightarrow R_1, R_2, R_3 \\ P_2 \rightarrow R_1, R_2 \end{array} \right.$$

**उदाहरण:**

- मान लीजिए 3 processes हैं:
- P1 को total 5 units चाहिए
- P1 ने अभी 2 units लिए हैं
- अब P1 और 1 unit मांगती है
- System check करेगा कि 1 और देने पर बाकी processes को future में क्या जरूरतें हैं और क्या पूरा किया जा सकता है। अगर नहीं, तो 1 unit नहीं दी जाएगी।

**Example:**

- Suppose there are 3 processes:
- P1 needs a total of 5 units
- P1 has currently taken 2 units
- Now P1 asks for 1 more unit
- The system will check if the other processes have future needs and can be fulfilled by giving 1 more. If not, 1 unit will not be given.



## ◆ Advantages (लाभ):

- सिस्टम लचीला होता है
- संसाधनों का कुशल उपयोग संभव
- वास्तविक समय में निर्णय लेना

## ◆ Disadvantages (हानियाँ):

- Algorithm complex होता है
- Processes को पहले से ही अपनी maximum need बतानी पड़ती है
- Overhead ज्यादा होता है

## ◆ Advantages (लाभ):

- System flexible होता है
- Efficient resource utilization possible
- Real-time decision making

## ◆ Disadvantages (हानियाँ):

- The algorithm is complex
- Processes have to specify their maximum requirement in advance
- Overhead is high





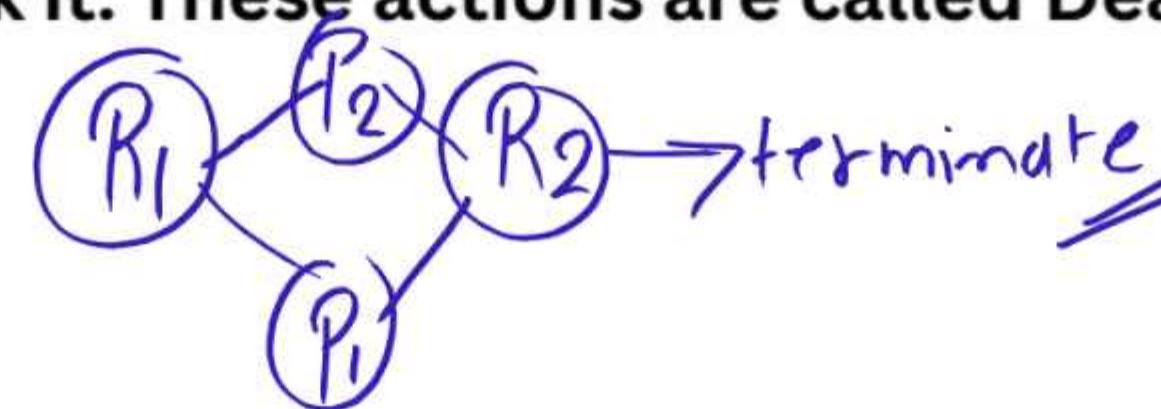
### 3: Deadlock Detection:

- Deadlock Detection एक ऐसा तरीका है जिससे Operating System यह पहचानता है कि क्या system में कोई deadlock हुआ है या नहीं।
- Deadlock detection is a way in which an operating system identifies whether a deadlock has occurred in a system.

$\Rightarrow$ (DAG )

### 4: Deadlock Recovery:

- जब सिस्टम में Deadlock detect हो जाता है, तो Operating System को उसे खत्म (break) करने के लिए कुछ actions लेने होते हैं। इन्हीं actions को Deadlock Recovery कहा जाता है।
- When a deadlock is detected in the system, the Operating System has to take some actions to break it. These actions are called Deadlock Recovery.



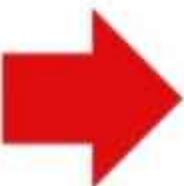


## Unit No 3:

## UNIT 3: Memory Management:

- Memory allocation ✓
- First Fit, Best fit & Worst Fit ✓
- Swapping, Paging and Segmentation, Page faults,
- Page Replacement Algorithm
- Fragmentation & Compaction
- Concept of virtual Memory.

[10 → 15  
Number]

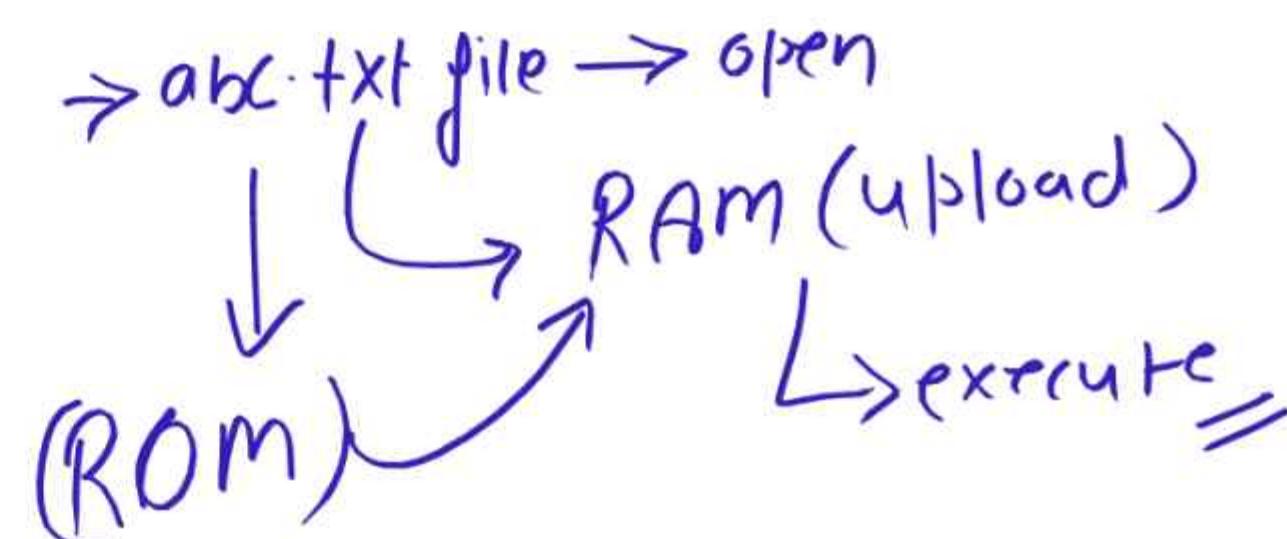


## Memory Allocation:

- Memory Allocation एक प्रक्रिया है जिसमें Operating System (OS) किसी प्रोग्राम (Process) को मेमोरी (RAM) में स्थान प्रदान करता है ताकि वह एक्सीक्यूट हो सके। जब कोई प्रोग्राम चलाया जाता है, तो उसे कुछ मेमोरी की आवश्यकता होती है। OS यह तय करता है कि किस प्रोग्राम को कितनी मेमोरी और कब दी जाए।
- Memory allocation is a process in which the Operating System (OS) allocates space in memory (RAM) to a program (process) so that it can execute. When a program is run, it requires some memory. The OS decides how much memory is given to which program and when.

## Types of Memory Allocation :

- 1. Static Memory Allocation
- 2. Dynamic Memory Allocation





## 1. Static Memory Allocation

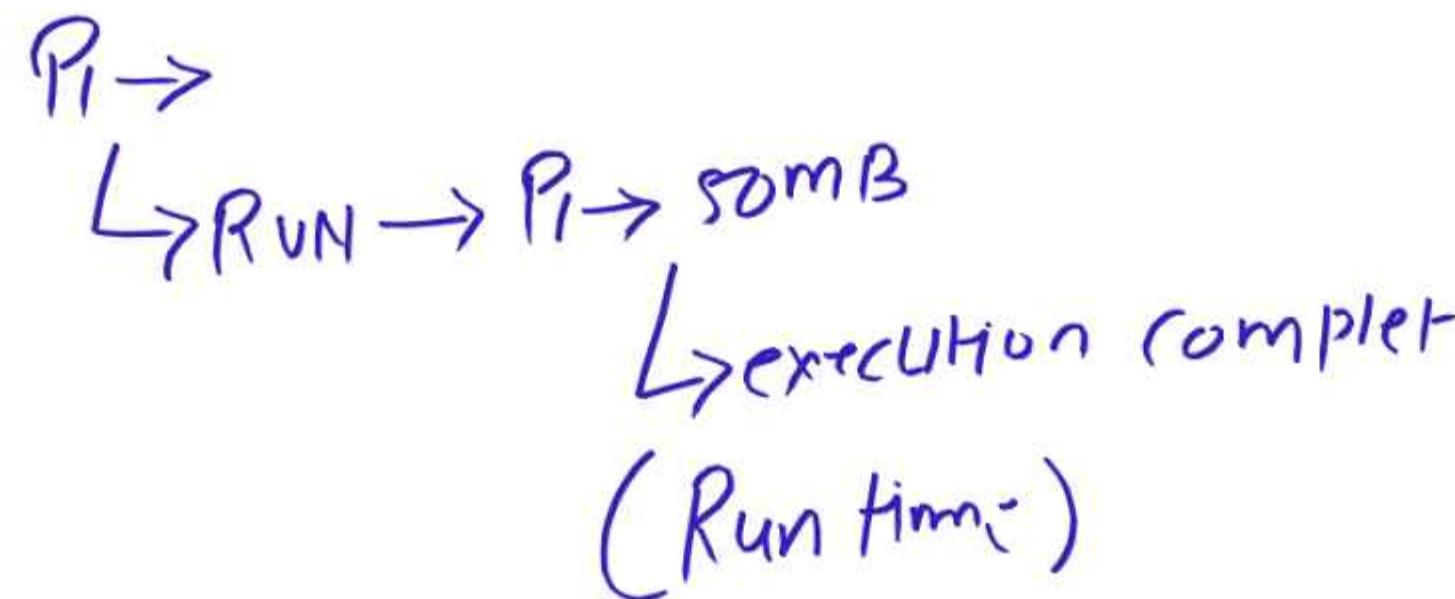
- इसमें मेमोरी को Compile Time पर ही तय कर दिया जाता है।
- जब प्रोग्राम रन होता है, तो उसे पहले से निर्धारित मेमोरी का ही उपयोग करना पड़ता है।
- यह एक फिक्स्ड एलोकेशन होती है, जिसे बाद में बदला नहीं जा सकता।
- In this the memory is decided at compile time itself.
- When the program runs, it has to use the pre-determined memory only.
- This is a fixed allocation, which cannot be changed later.

$P_1 \rightarrow 10\text{mB}$   
↳ execute ( $20\text{mB}$ ) fail



## 2. Dynamic Memory Allocation:

- इसमें मेमोरी को रन टाइम (Run Time) पर एलोकेट किया जाता है।
- अगर जरूरत हो तो मेमोरी को बढ़ाया या घटाया जा सकता है।
- उपयोग के बाद मेमोरी को मुक्त (free) भी किया जा सकता है।
- In this the memory is allocated at run time.
- If needed the memory can be increased or decreased.
- The memory can also be freed after use.



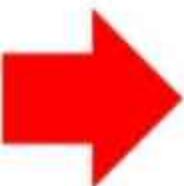


## Difference Between Static & Dynamic Memory Allocation:

विशेषता	Static Memory Allocation	Dynamic Memory Allocation
अर्थ (Meaning)	मेमोरी compile time पर निर्धारित होती है	मेमोरी run time पर allocate होती है
मेमोरी का समय (Timing)	Compile time	Run time
लचीलापन (Flexibility)	कम - पहले से तय होती है	अधिक - जरूरत के अनुसार मेमोरी मिलती है
प्रभाव (Efficiency)	तेज़, पर मेमोरी का waste हो सकता है	कुशल, मेमोरी का बेहतर उपयोग
Memory Use	Fixed size memory मिलती है	Size रन टाइम पर decide होता है
Function Example (C language)	नहीं होता (int a[10]; जैसे)	malloc(), calloc(), free()



<b>Reallocation</b>	संभव नहीं	Possible using realloc()
<b>Fragmentation</b>	Internal fragmentation हो सकता है	External fragmentation हो सकता है
<b>Access Speed</b>	तेज (fast access)	थोड़ा धीमा (due to pointers)
<b>उदाहरण</b>	Array, Static variables	Linked List, Trees (runtime nodes)



### ♦ Allocation method in continuous memory allocation:

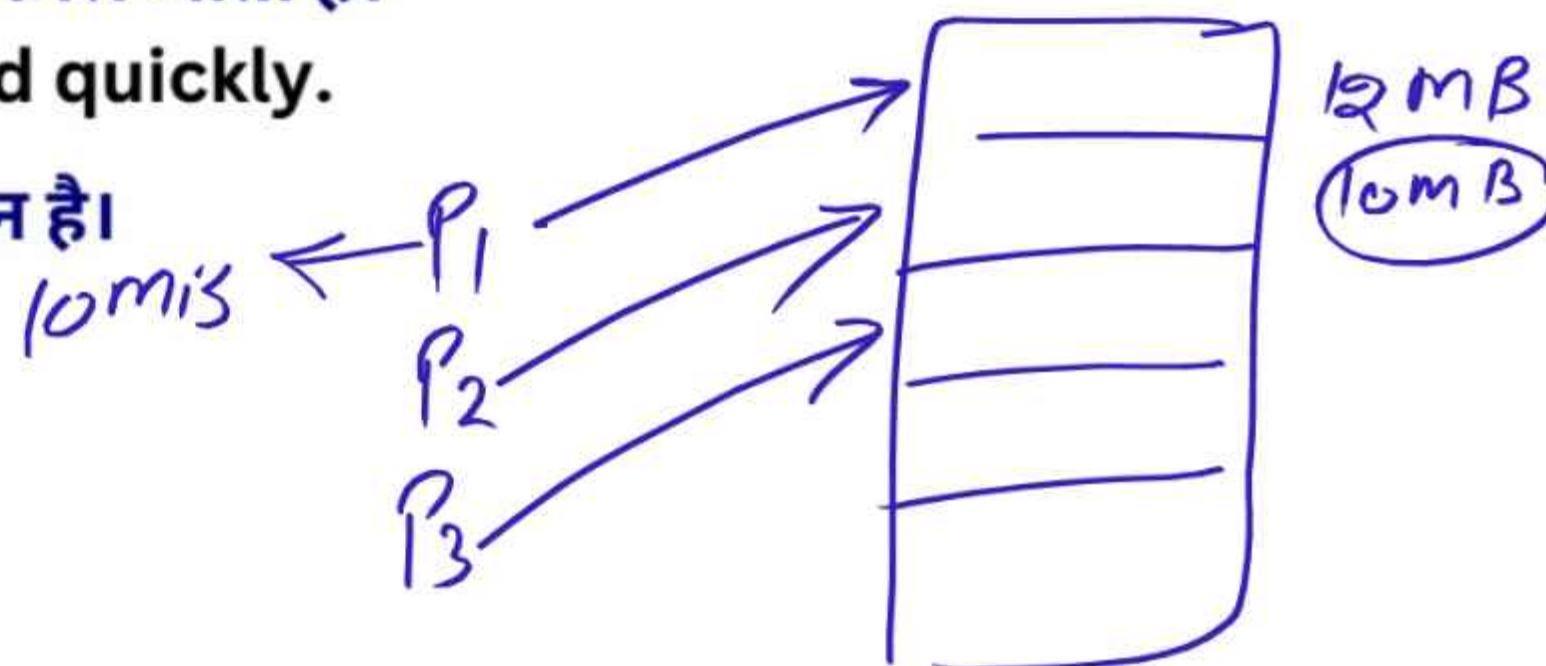
- Contiguous Memory Allocation में हर प्रोसेस को RAM में एक लगातार (continuous) ब्लॉक में जगह दी जाती है। लेकिन जब कई प्रोसेसेस आते हैं, तो किसे कितनी और कहाँ मेमोरी मिलेगी, इसे तय करने के लिए तीन मुख्य तरीके (Allocation Methods) होते हैं।
- In Contiguous Memory Allocation, each process is given space in a continuous block of RAM. But when multiple processes come, there are three main allocation methods to decide who will get how much memory and where.

{ First Fit (पहला फिट)  
Best Fit (सबसे अच्छा फिट)  
Worst Fit (सबसे खराब फिट)



### ◆ First Fit (पहला फिट):

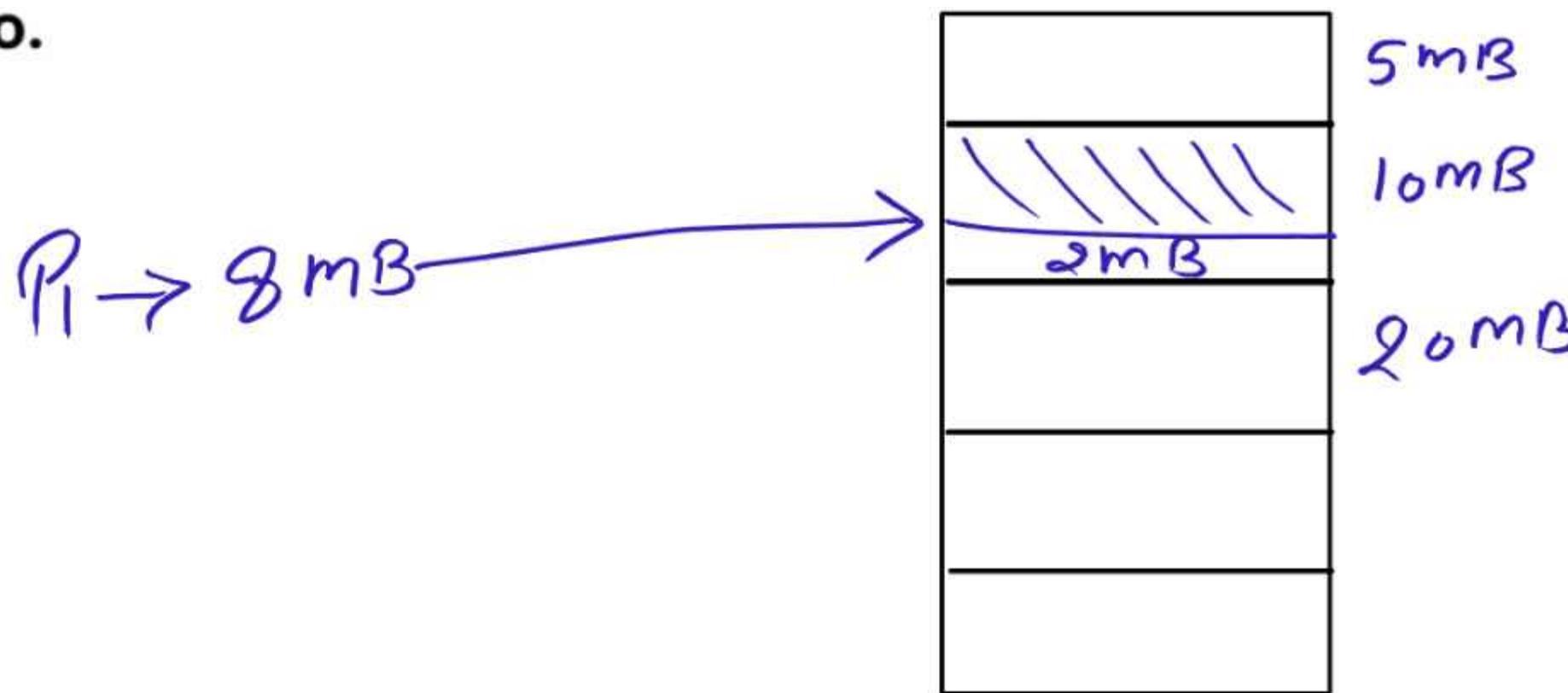
- इस विधि में, पहला उपलब्ध (खाली) मेमोरी ब्लॉक, जो प्रोसेस के लिए पर्याप्त हो, उसे आवंटित कर दिया जाता है।
- In this method, the first available (empty) memory block that is sufficient for the process is allocated.
- तेज़ (Fast) – पहला उपलब्ध स्पेस जल्दी मिल जाता है।
- The first available space is found quickly.
- सरल (Simple) – इसे लागू करना आसान है।
- It is easy to implement.





◆ Example:

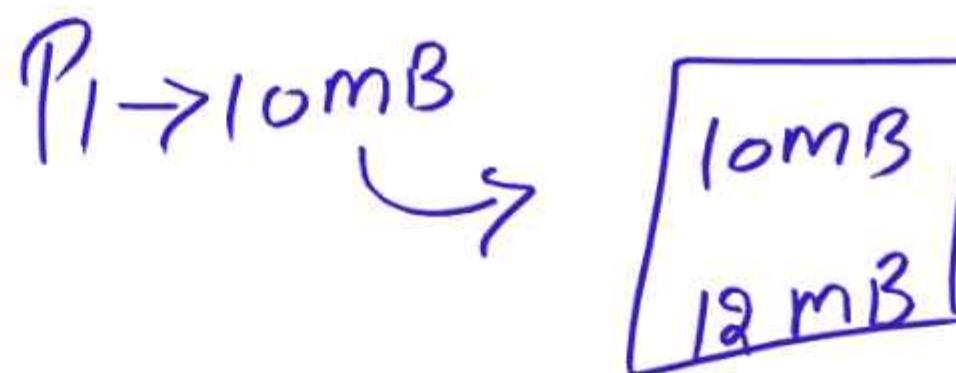
- अगर मेमोरी में ~~5MB, 10MB और 20MB~~ के खाली ब्लॉक हैं और 8MB का प्रोसेस आता है, तो इसे ~~10MB~~ वाले ब्लॉक में रखा जाएगा, क्योंकि वह पहला उपलब्ध ब्लॉक है जो इसे फिट कर सकता है।
- If the memory has ~~5MB, 10MB, and 20MB~~ empty blocks, and an 8MB process arrives, it will be placed in the ~~10MB~~ block because that is the first available block it can fit into.





◆ Best Fit (सबसे अच्छा फिट):

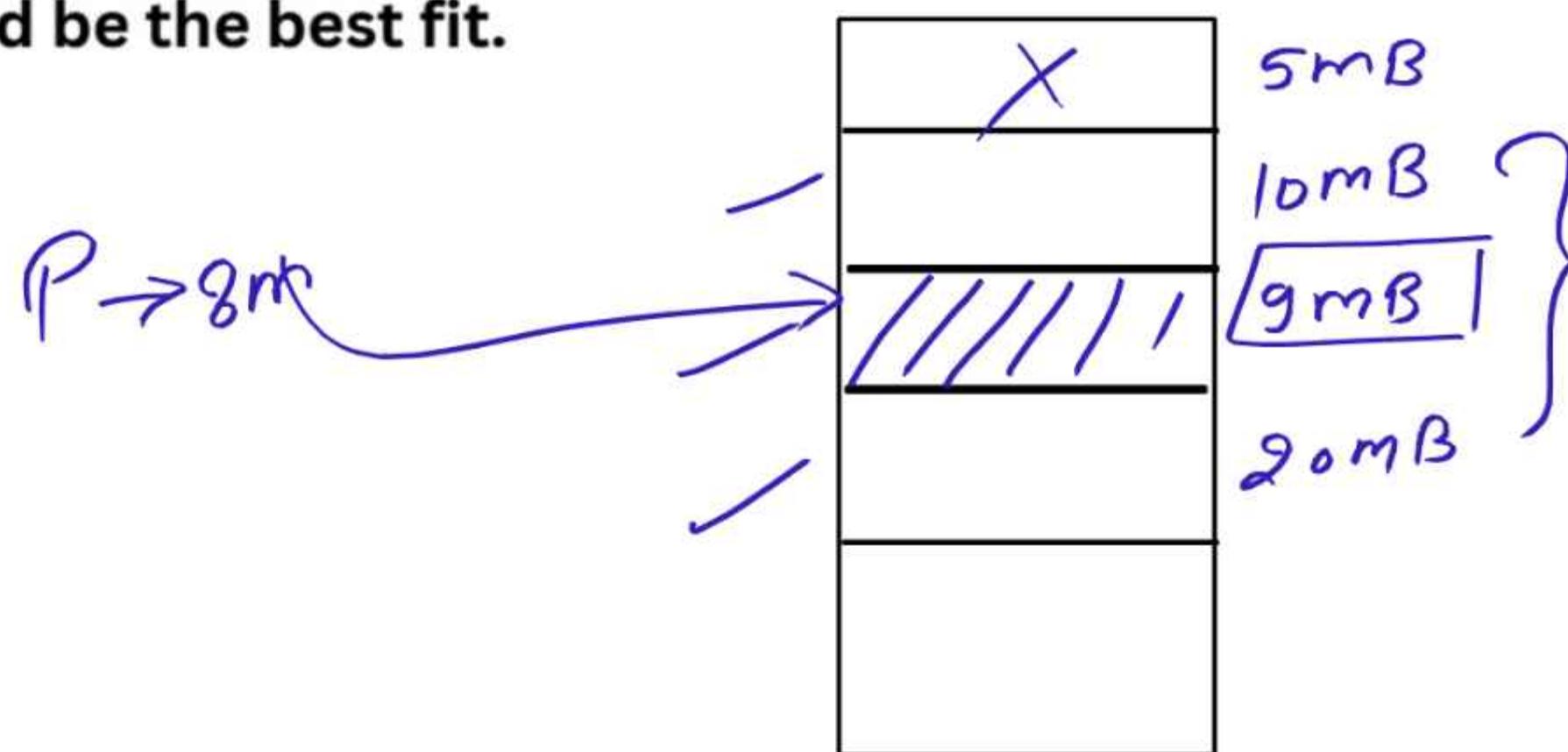
- इस विधि में, सबसे छोटा उपलब्ध खाली ब्लॉक ढूँढ़ा जाता है, जिसमें प्रोसेस पूरी तरह फिट हो सके।
- In this method, the smallest available empty block is found in which the process can fit completely.
- मेमोरी का उपयोग बेहतर तरीके से होता है।
- Memory is used in a better way.
- Internal Fragmentation कम होती है।
- Internal fragmentation is reduced.





### ◆ Example:

- अगर मेमोरी में 5MB, 10MB , 9MB और 20MB के खाली ब्लॉक हैं और 8MB का प्रोसेस आता है, तो इसे 10MB वाले ब्लॉक में रखा जाएगा (न कि 20MB में), क्योंकि यह सबसे अच्छा फिट होगा।
- If there are 5MB, 10MB, and 20MB free blocks in memory, and an 8MB process arrives, it will be placed in the 10MB block (not the 20MB one), because that would be the best fit.





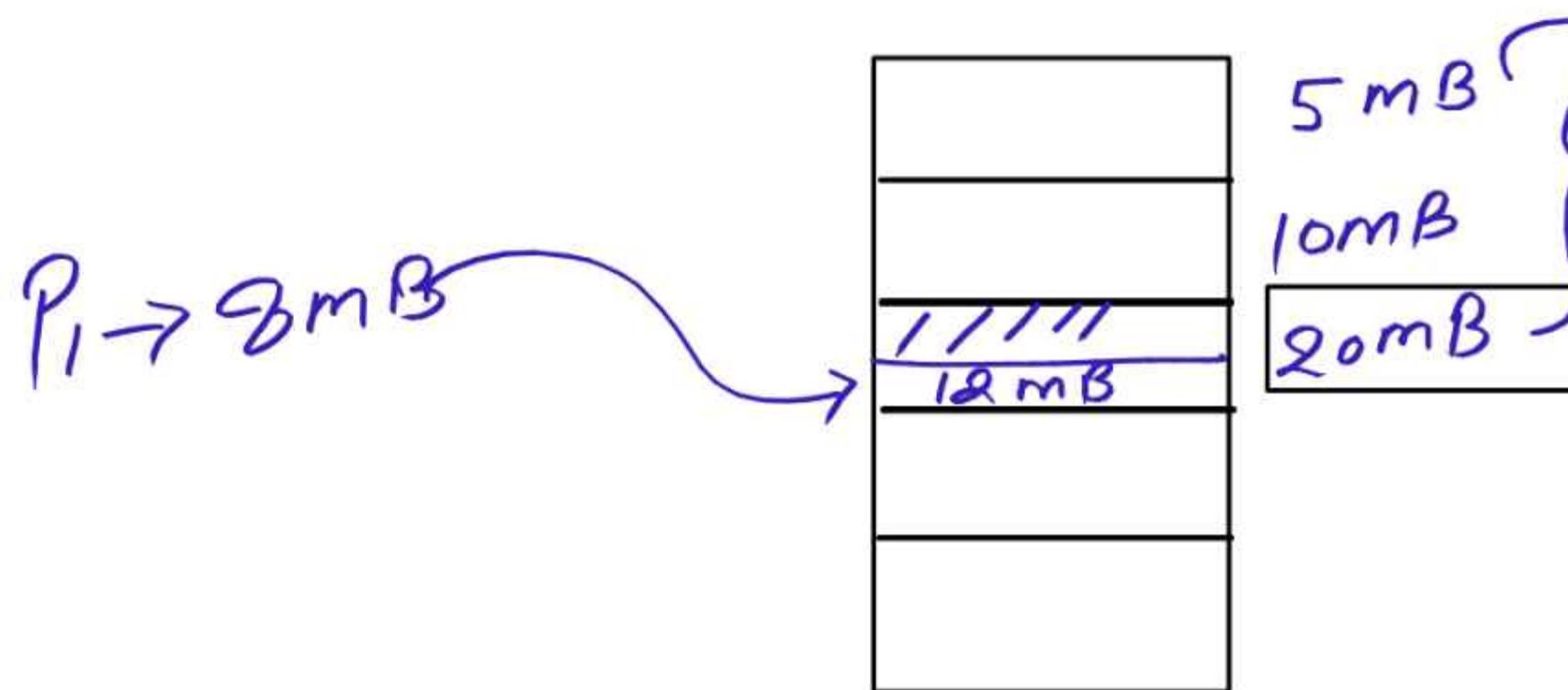
◆ **Worst Fit (सबसे खराब फिट):**

- इस विधि में, सबसे बड़ा उपलब्ध खाली ब्लॉक चुना जाता है ताकि मेमोरी में और छोटे हिस्से बच सकें।
- In this method, the largest available empty block is selected so that more smaller chunks can be saved in the memory.
- बड़े ब्लॉक्स में छोटे प्रोसेस को जगह मिल सकती है।
- Smaller processes can be accommodated in larger blocks.
- कुछ मामलों में External Fragmentation कम हो सकती है।
- External Fragmentation may be reduced in some cases.



◆ Example:

- अगर मेमोरी में 5MB, 10MB और 20MB के खाली ब्लॉक हैं और 8MB का प्रोसेस आता है, तो इसे 20MB वाले ब्लॉक में रखा जाएगा, जिससे 12MB बचा रहेगा और छोटे प्रोसेस उसमें फिट हो सकें।
- If the memory has 5MB, 10MB, and 20MB free blocks and an 8MB process arrives, it will be placed in the 20MB block, leaving 12MB free for smaller processes to fit into.

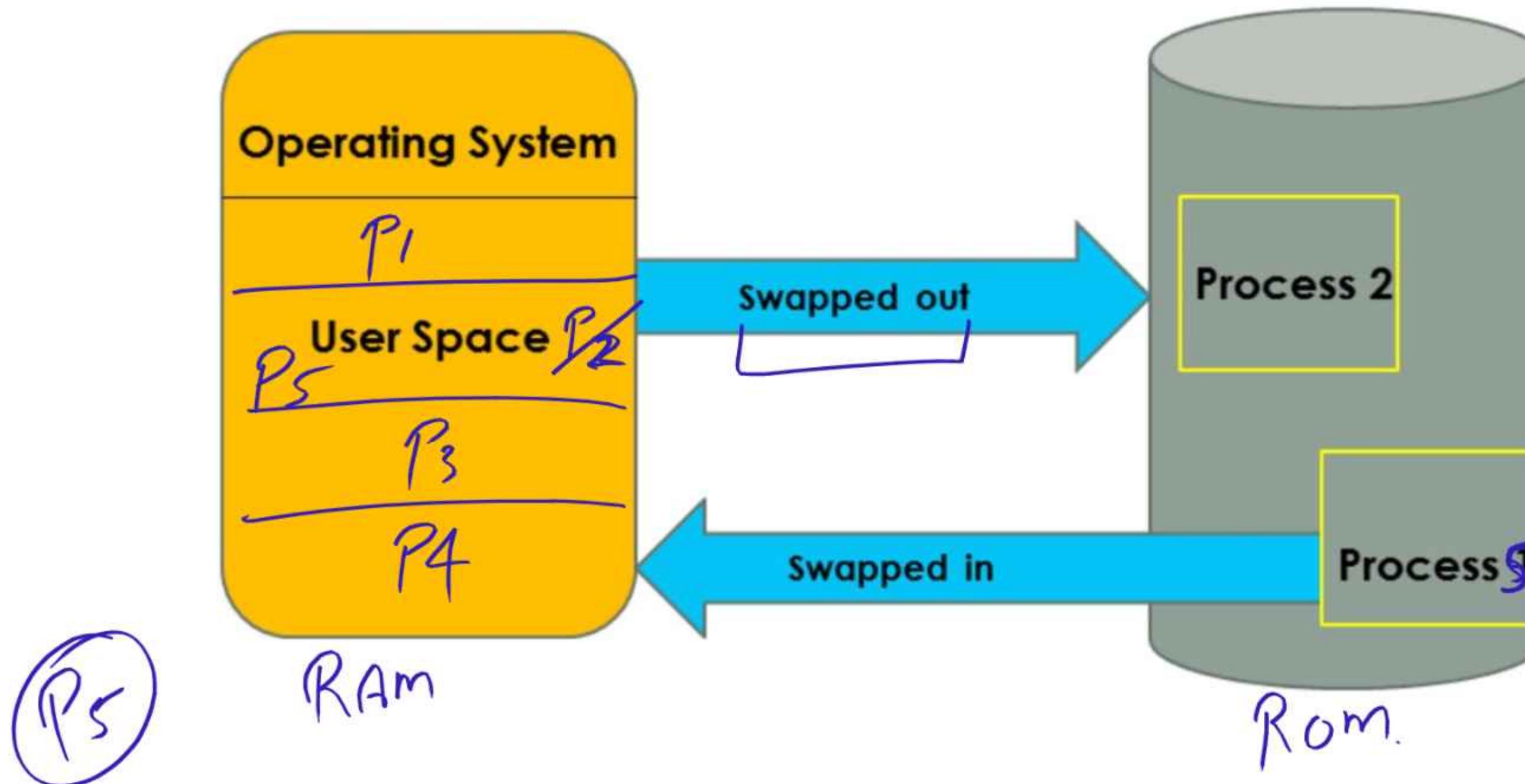


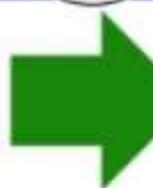


## ➤ ~~Swapping:~~

- Swapping ऑपरेटिंग सिस्टम (OS) की एक Memory Management Technique है जिसमें किसी प्रोसेस को मेमोरी (RAM) से हटाकर हार्ड डिस्क (secondary storage) में ले जाया जाता है, और जरूरत होने पर फिर से RAM में वापस लाया जाता है।
- Swapping is a memory management technique in an operating system (OS) in which a process is removed from memory (RAM) and moved to the hard disk (secondary storage), and then brought back to RAM when needed.

RAM → limited (8GB, 16GB, 64GB)  
ROM → 500GB, 1TB





## Process of Swapping :

- RAM में पहले से कुछ प्रोसेसेज़ चल रही होती हैं। ✓
- Swapping RAM already has some processes running in it.
- कोई नया प्रोसेस आने पर अगर RAM में जगह नहीं है ✓
- If there is no space in RAM when a new process comes in
- OS एक पुराने/Idle प्रोसेस को चुनता है और उसे Swap Out कर देता है (Disk पर भेजता है) ✓
- OS selects an old/idle process and swaps it out (sends it to disk).
- RAM में नया प्रोसेस लोड हो जाता है। ✓
- The new process gets loaded in RAM.
- जब पुराने प्रोसेस की जरूरत फिर से होती है, तो OS उसे Swap In करके RAM में वापस लाता है।
- When the old process is needed again, OS swaps it in and brings it back to RAM.



## Advantages of Swapping (स्वैपिंग के फायदे):

### 1: Better Memory Utilization (बेहतर मेमोरी उपयोग):

- RAM की लिमिट होने पर भी ज्यादा प्रोसेसेज़ को हैंडल किया जा सकता है।
- More processes can be handled even if RAM is limited.

### 2: Increased Multiprogramming (मल्टीप्रोग्रामिंग को बढ़ावा):

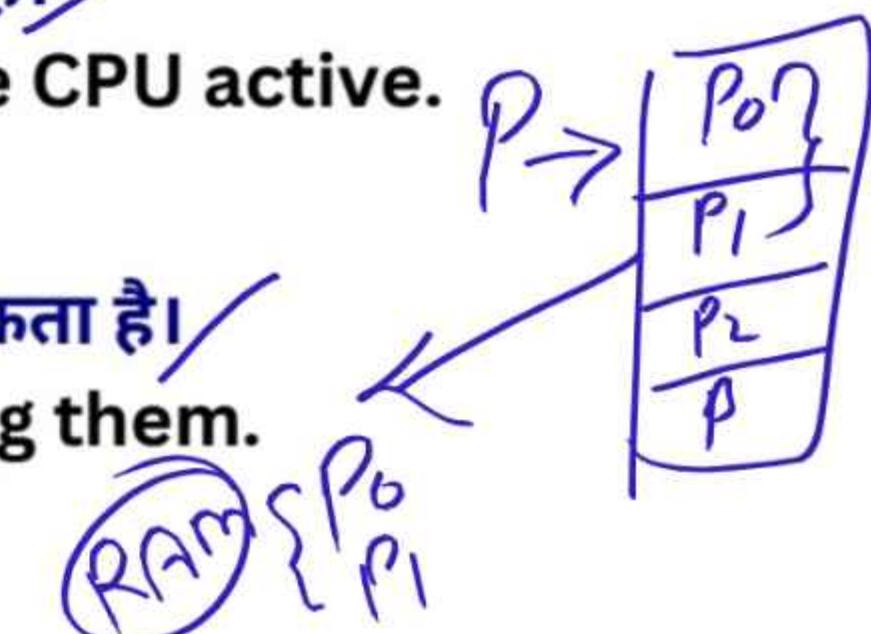
- एक साथ ज्यादा प्रोग्राम्स चल सकते हैं, क्योंकि कुछ को डिस्क में रखा जा सकता है।
- More programs can run simultaneously, since some can be stored on disk.

### 3: CPU Utilization Improves (CPU का उपयोग बढ़ता है):

- Idle CPU को active रखने के लिए RAM से नए प्रोसेस लाए जा सकते हैं।
- New processes can be brought from RAM to keep the idle CPU active.

### 4: Support for Larger Processes (बड़े प्रोग्राम्स को सपोर्ट):

- जो processes RAM में फिट नहीं होते, उन्हें swap करके चलाया जा सकता है।
- Processes which do not fit in RAM can be run by swapping them.





Dk

## Advantages of Swapping (स्वैपिंग के फायदे):

### 1: Slow Performance (धीमी गति):

- Disk की read/write speed RAM से काफी कम होती है, जिससे performance कम हो जाती है।
- The read/write speed of disk is much lower than RAM, which reduces performance.

### 2: Thrashing Problem (थ्रैशिंग की समस्या):

- बार-बार swapping होने से CPU सिर्फ swapping में ही बिजी हो जाता है, जिससे system hang करने लगता है।
- Due to frequent swapping, the CPU becomes busy only in swapping, due to which the system starts hanging.

### 3 Power Consumption (ऊर्जा की खपत):

- बार-बार डिस्क access करने से ज्यादा बिजली खर्च होती है (खासतौर पर लैपटॉप्स में)।
- Frequent disk access consumes more power (especially in laptops).



## Paging:

*Imp*

- Paging एक मेमोरी मैनेजमेंट तकनीक है, जिसमें प्रोसेस को छोटे-छोटे बराबर साइज के भागों (Pages) में तोड़ा जाता है और उन्हें RAM में स्टोर किया जाता है।
- Paging is a memory management technique in which the process is broken down into small equal sized parts (pages) and stored in RAM.
- पेजिंग को लागू करने की मूल विधि में फिजिकल मेमोरी (RAM) को एक निश्चित आकार के ब्लॉक्स में विभाजित किया जाता है, जिन्हें फ्रेम (Frames) कहा जाता है, और लॉजिकल मेमोरी (प्रोसेस) को उसी आकार के ब्लॉक्स में विभाजित किया जाता है, जिन्हें पेज (Pages) कहा जाता है।
- The basic method of implementing paging is to divide physical memory (RAM) into blocks of a fixed size, called frames, and logical memory (processes) into blocks of the same size, called pages.

$P_1 \rightarrow 6 \text{ byte}$

↳ every page size is equal

**1: पेज नंबर (P):**

- लॉजिकल एड्रेस स्पेस में मौजूद पेज को दर्शाने के लिए आवश्यक बिट्स की संख्या, जिसे पेज नंबर भी कहा जाता है।
- The number of bits required to represent a page in the logical address space, also called the page number.

**2: पेज ऑफसेट (d):**

- किसी विशेष वर्ड (डेटा) को पेज के भीतर दर्शाने के लिए आवश्यक बिट्स की संख्या, जिसे पेज का साइज, या पेज ऑफसेट भी कहा जाता है।
- The number of bits required to represent a particular word (data) within a page, also called the page size, or page offset.

**3: फ्रेम नंबर (F):**

- फिजिकल एड्रेस स्पेस में किसी फ्रेम को दर्शाने के लिए आवश्यक बिट्स की संख्या, जिसे फ्रेम नंबर भी कहा जाता है।
- The number of bits required to represent a frame in the physical address space, also called the frame number.



#### 4: फ्रेम ऑफसेट (d):

- किसी विशेष वर्ड (डेटा) को फ्रेम के भीतर दर्शाने के लिए आवश्यक बिट्स की संख्या, जिसे फ्रेम का साइज, या फ्रेम ऑफसेट भी कहा जाता है।
- The number of bits required to represent a particular word (data) within a frame, also called the frame size, or frame offset.

Process  $\rightarrow$  6 byte

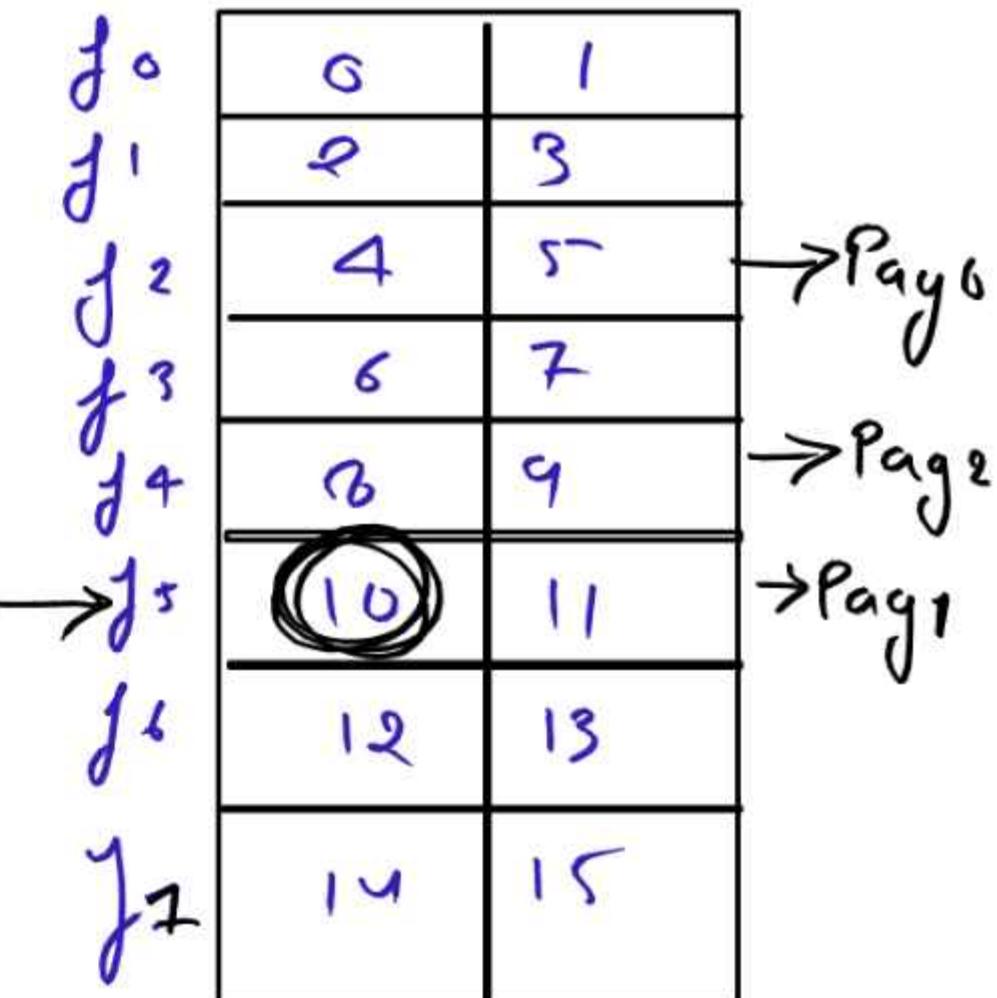
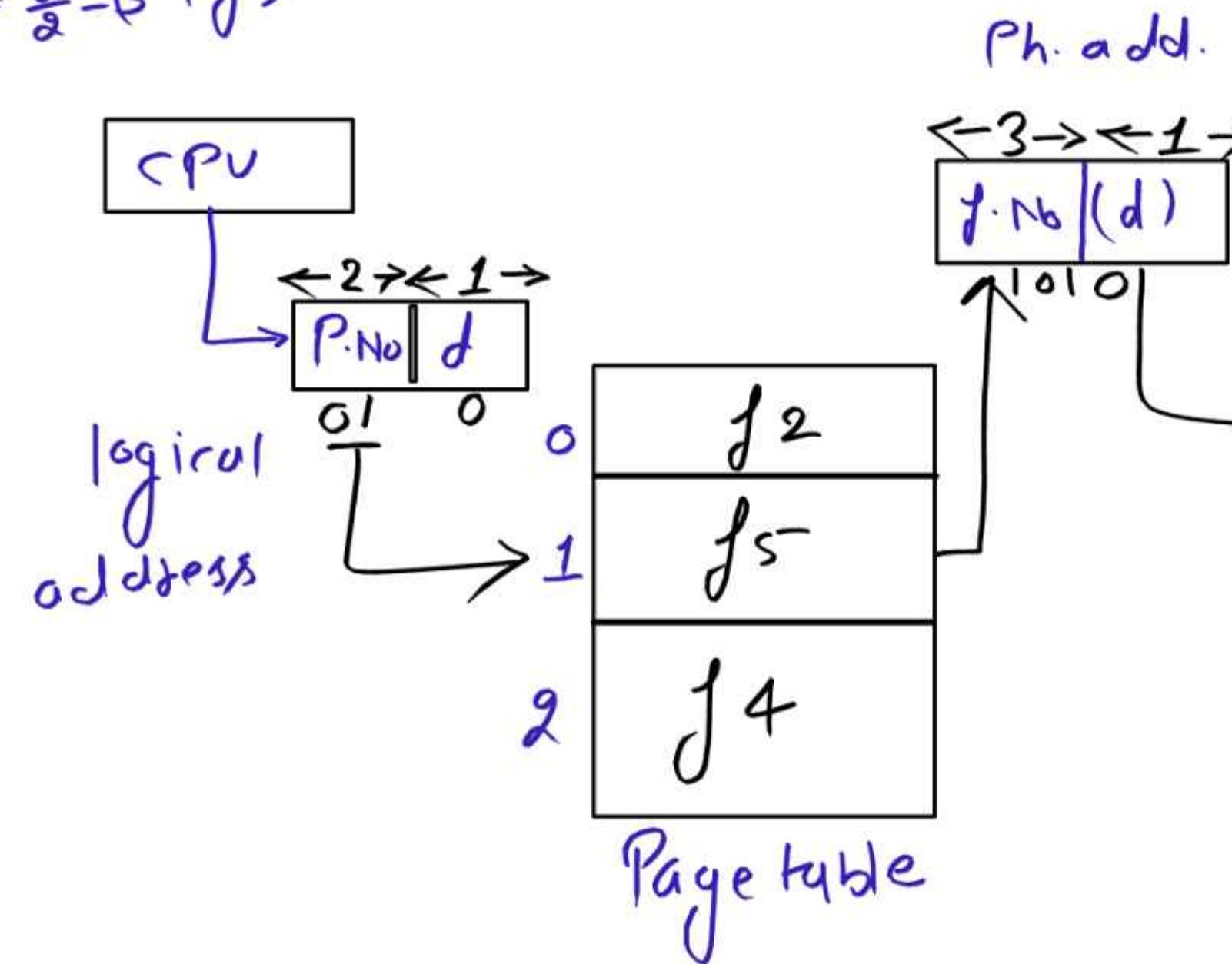
Page size  $\rightarrow$  2 byte

$$\text{No of Pages} = \frac{6}{2} = 3 \text{ Page}$$

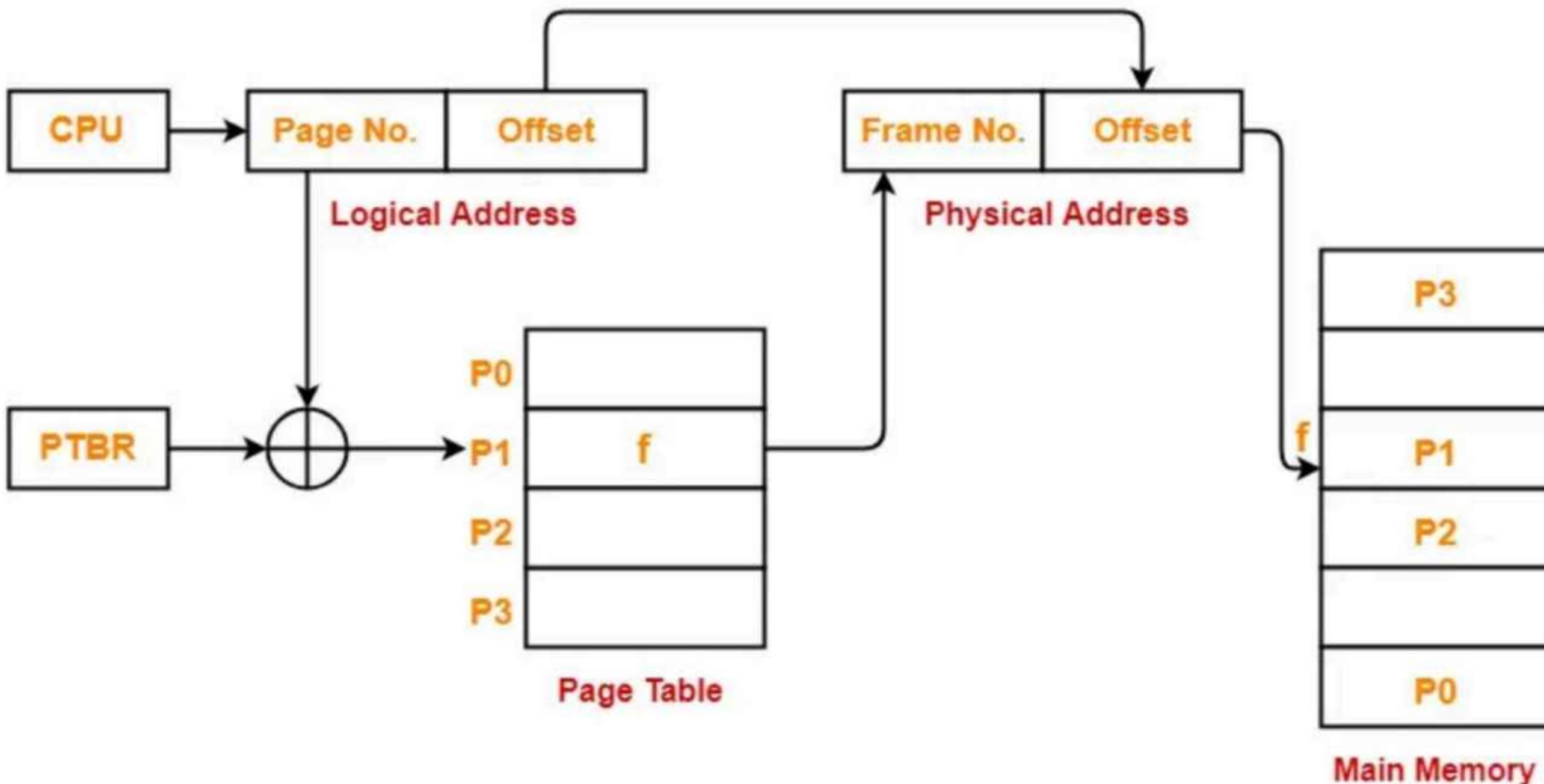
	0	1
0	0	1
1	2	3

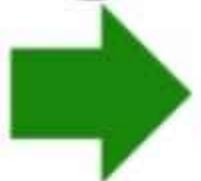
Process

RAM  $\rightarrow$  16 byte  
frame size  $\rightarrow$  2 byte  
No of frame  $\rightarrow \frac{16}{2} = 8$  frame



$j \rightarrow 111$        $1010 \rightarrow$   
 $s \rightarrow 101$        $\hookrightarrow 10$





## Advantage of paging:

- बाहरी फ्रैगमेंटेशन नहीं होता
- मेमोरी का बेहतर उपयोग होता है
- निर्धारित (Fixed) आकार होने से लागू करना आसान
- वर्चुअल मेमोरी को सपोर्ट करता है
- लगातार (Contiguous) मेमोरी की आवश्यकता नहीं होती
- Pages को स्वैप करना आसान होता है
- No External Fragmentation ✓
- Better Memory Utilization ✓
- Easy Implementation (Fixed-size blocks)
- Supports Virtual Memory ✓
- Non-contiguous Memory Allocation Possible ✓
- Efficient Swapping of Pages ✓



## Disadvantage of paging:

- आंतरिक Fragmentation हो सकती है।
- मेमोरी प्रबंधन (Memory Management) का एल्गोरिदम जटिल होता है।
- Page Table को स्टोर करने के लिए ज्यादा मेमोरी की जरूरत होती है।
- Multi-level Paging से मेमोरी एक्सेस करने में समय ज्यादा लग सकता है।
- Paging के लिए खास हार्डवेयर चाहिए, जिससे कंप्यूटर महंगा हो सकता है।
- Internal fragmentation may occur.
- The memory management algorithm is complex.
- More memory is required to store the page table.
- Multi-level paging may take longer to access memory.
- Paging requires special hardware, which can make the computer expensive.



## Segmentation:

- Segmentation एक मेमोरी प्रबंधन तकनीक है। Segmentation किसी प्रोग्राम को छोटे-छोटे भागों (Segments) में विभाजित करता है। एक लॉजिकल एड्रेस स्पेस (Logical Address Space) कई Segments का समूह होता है। प्रत्येक Segment का एक नाम (Name) और एक लंबाई (Length) होती है।
- Segmentation is a memory management technique. Segmentation divides a program into smaller parts (segments). A logical address space is a group of many segments. Each segment has a name and a length.

• **Base:** यह सेगमेंट का शुरुआती (बेस) पता होता है।

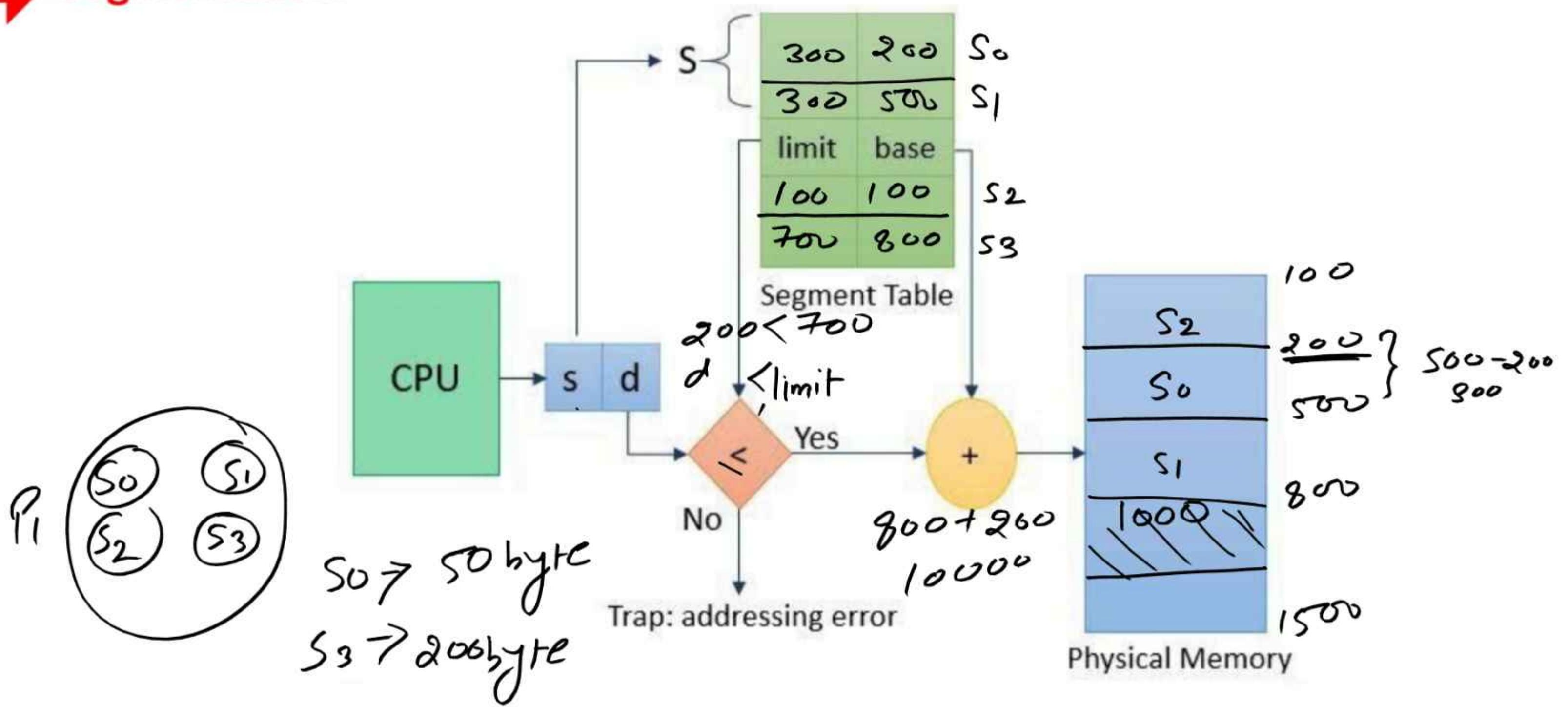
- It is the base address of the segment.

• **Limit:** यह सेगमेंट की कुल लंबाई (साइज़) होती है।

- It is the length of the segment.



## Segmentation:





## Segmentation के लाभ :

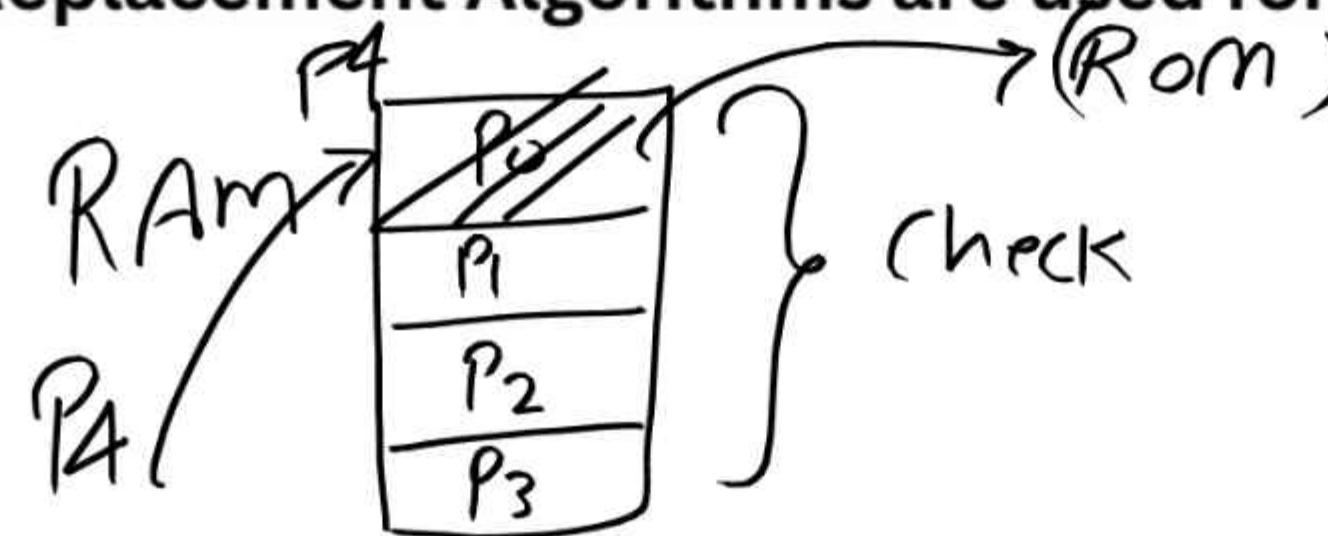
1. मेमोरी का तार्किक (logical) विभाजन
2. बेहतर सुरक्षा (Protection)
3. आंतरिक Fragmentation नहीं होता
4. सेगमेंट का डायनामिक रूप से बढ़ना संभव
5. प्रोग्राम प्रबंधन आसान होता है
6. मेमोरी शेयरिंग और Relocation सरल

1. Logical division of memory
2. Better memory protection
3. No internal fragmentation
4. Supports dynamic segment growth
5. Easier program management
6. Simplifies memory sharing and relocation



## Page Replacement:

- जब मेमोरी (RAM) में कोई नया पेज लाने के लिए जगह नहीं होती, तो हमें पुराने पेज को हटाना पड़ता है। यह प्रक्रिया Page Replacement कहलाती है। इसके लिए अलग-अलग **Page Replacement Algorithms** का उपयोग किया जाता है।
- When there is no space in the memory (RAM) to bring a new page, we have to remove the old page. This process is called Page Replacement. Different Page Replacement Algorithms are used for this.





### 1. FIFO (First In First Out) Algorithm:

- जो सबसे पहले मेमोरी में आया था, उसे सबसे पहले हटा दिया जाता है।
- The first thing that came into the memory is the first one to be removed.
- इसे Queue (FIFO Queue) के आधार पर लागू किया जाता है।
- It is implemented on the basis of Queue (FIFO Queue)

**Example:**

- Page Reference String: 5, 0, 1, 2, 0, 3, 4, 2, 3, 0, 3, 2
- Frames = 3

$$\text{Fault Ratio} = \frac{\text{T.F}}{\text{T.S}} \times 100$$

$f_2$		1	1	1	1	4	4	4	4	4	
$f_1$	0	0	0	3	3	3	3	3	3	2	
$f_0$	5	5	5	2	2	2	2	2	0	0	
*	*	*	*	hit	*	*	hit	hit	*	hit	*

Page fault = 8

Page hit = 4

$$\text{hit Ratio} = \frac{\text{Total hit}}{\text{total string}} \times 100 = \frac{4}{12} \times 100$$



## 2. LRU (Least Recently Used) Algorithm:

- जो पेज सबसे लंबे समय से उपयोग नहीं हुआ है, उसे हटाया जाता है।
- The page which has not been used for the longest time is removed.
- इसे Stack या Counter के आधार पर लागू किया जाता है।
- This is implemented on Stack or Counter basis.



## Example:

- Page Reference String: 5, 0, 1, 2, 0, 3, 4, 2, 3, 0, 3, 2
  - Frames = 3

$f_2$	*	*	1	1	<del>1</del>	3	3	3	3	3	3	3
$f_1$	0	0	0	0	0	<del>0</del>	2	2	2	2	2	2
$f_0$	5	5	<del>5</del>	2	2	<del>2</del>	4	4	<del>4</del>	0	0	6

hit = 4

$$f_{\text{aur}} = \emptyset$$

$$\text{Hit Ratio} = \frac{\text{TH}}{\text{T. Storing}} \times 100$$

$$\text{fault Ratio} = \frac{T_{\text{fault}}}{\text{total string}} \times 100$$



### 3. Optimal Page Replacement Algorithm:

- जो भविष्य में सबसे देर से उपयोग होगा, उसे हटाया जाता है।
- The page which will be used least in the future is removed.
- यह सबसे कम Page Faults देता है, लेकिन प्रैक्टिकली लागू करना मुश्किल है क्योंकि भविष्य में कौन-सा पेज उपयोग होगा, यह पहले से नहीं पता होता।
- This gives least number of Page Faults but is difficult to implement practically because it is not known in advance which page will be used in the future.



## Example:

- Page Reference String: 5, 0, 1, 2, 0, 3, 4, 2, 3, 0, 3, 2
  - Frames = 3

hit = 5

fault = 7



## Fragmentation:

- जब कंप्यूटर की RAM (मेमोरी) में छोटे-छोटे खाली हिस्से बच जाते हैं और वे किसी नए प्रोग्राम (Process) के काम नहीं आते, तो इसे Fragmentation (खंडन) कहते हैं।
- यह तब होता है जब मेमोरी में जगह उपलब्ध होती है, लेकिन उसे सही तरीके से उपयोग नहीं किया जा सकता।
- When small empty parts remain in the computer's RAM (memory) and they are not useful for any new program (process), then it is called Fragmentation.
- This happens when space is available in the memory, but it cannot be used properly.



## Types of Fragmentation:

### 1: Internal Fragmentation:

- जब किसी प्रोसेस को जरूरत से ज्यादा मेमोरी मिल जाती है और बची हुई मेमोरी बेकार चली जाती है, तो इसे Internal Fragmentation कहते हैं।
- When a process gets more memory than required and the remaining memory goes waste, it is called Internal Fragmentation.

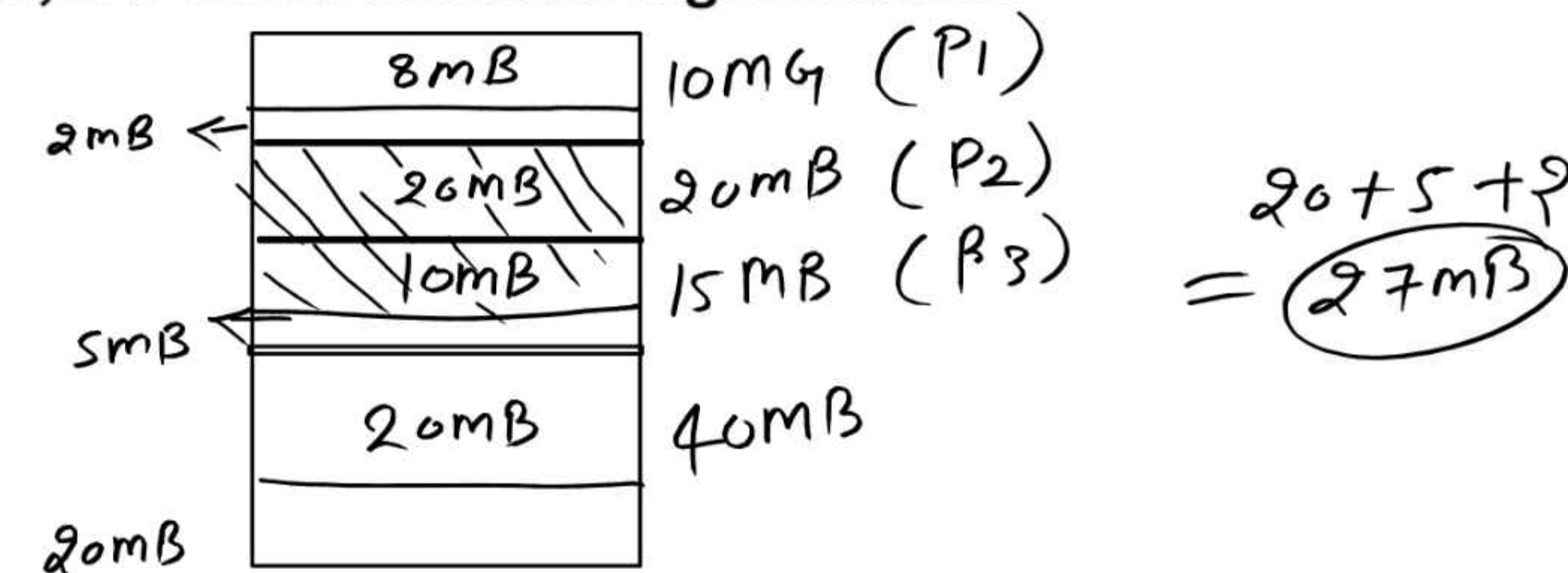
$$P_1 \rightarrow 8\text{mB}$$

$$P_2 \rightarrow 20\text{mB}$$

$$P_3 \rightarrow 10\text{mB}$$

$$P_4 \rightarrow 20\text{mB}$$

$$P_5 \rightarrow 25\text{mB}$$





## Advantages of Internal Fragmentation:

- फिक्स्ड-साइज़ ब्लॉक्स के कारण मेमोरी अलोकेशन तेज़ होता है।
- मेमोरी मैनेजमेंट सिस्टम सरल और आसान बनता है।
- सभी ब्लॉक्स समान होने से प्रोसेसिंग प्रेडिक्टेबल (पूर्वानुमान योग्य) होती है।
- तेजी से मेमोरी access और allocation संभव होता है।
- Fixed-size blocks make memory allocation faster.
- The memory management system is simplified and easier.
- Processing is predictable since all blocks are equal.
- Faster memory access and allocation is possible.



## Disadvantages of Internal Fragmentation:

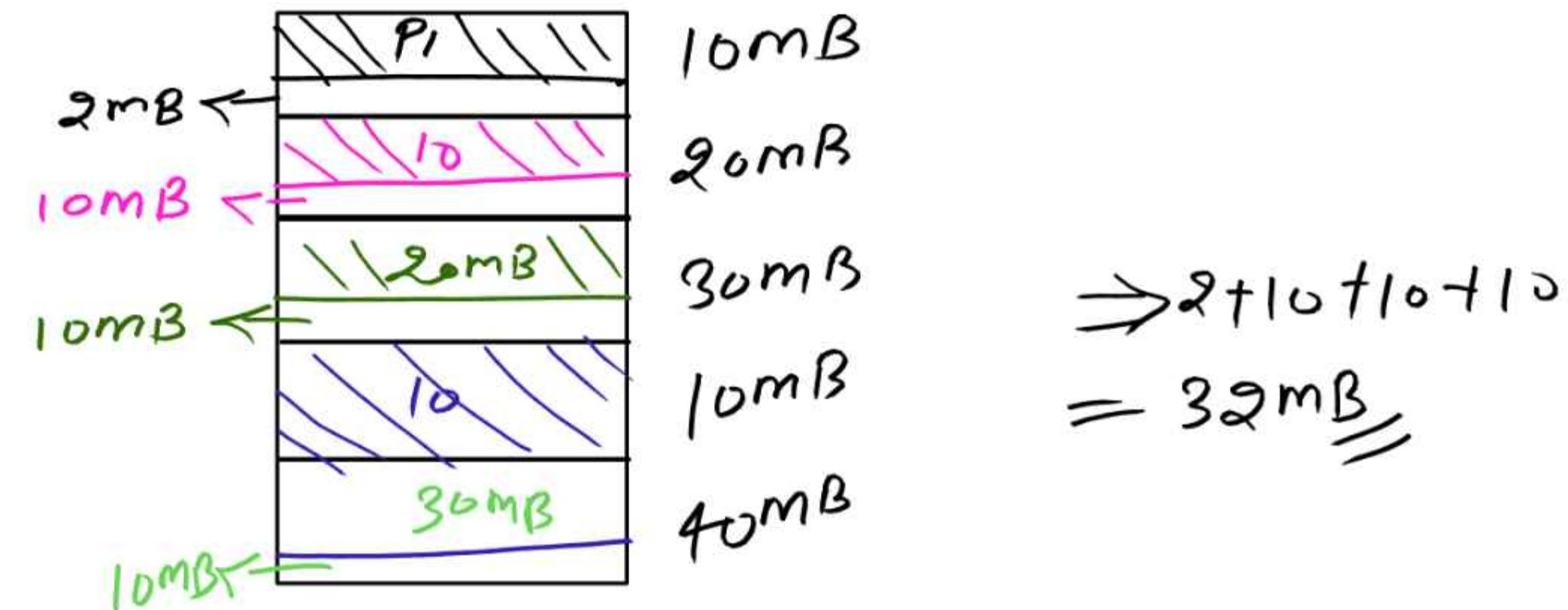
- मेमोरी की बर्बादी होती है।
- उपलब्ध मेमोरी पूरी तरह उपयोग नहीं हो पाती।
- सिस्टम की परफॉर्मेंस कम हो जाती है।
- छोटे प्रोसेस को भी बड़ा ब्लॉक मिल जाता है।
- स्केलेबिलिटी में समस्या आती है।
- Memory is wasted.
- Available memory is not fully utilized.
- System performance decreases.
- Even small processes get large blocks.
- There is a problem in scalability.



## 2: External Fragmentation:

- जब मेमोरी में पर्याप्त खाली स्थान होता है, लेकिन वह छोटे-छोटे टुकड़ों में बंटा होता है, जिससे कोई बड़ा प्रोग्राम उसमें फिट नहीं हो सकता, तो इसे External Fragmentation कहते हैं।
- When there is enough free space in the memory, but it is divided into small pieces so that a large program cannot fit in it, it is called External Fragmentation.

$P_1 \rightarrow 8\text{ MB}$   
 $P_2 \rightarrow 10\text{ MB}$   
 $P_3 \rightarrow 20\text{ MB}$   
 $P_4 \rightarrow 10\text{ MB}$   
 $P_5 \rightarrow 30\text{ MB}$   
 $P_6 \rightarrow 20\text{ MB}$





## Advantages of External Fragmentation:

- प्रोसेस को मेमोरी ब्लॉक उनके असली साइज के अनुसार मिलते हैं।
  - मेमोरी का यूज थोड़ा बेहतर होता है (Fixed Size ब्लॉक्स की तुलना में)।
  - छोटे प्रोसेस को छोटे ब्लॉक्स आसानी से अलॉट किए जा सकते हैं।
  - यह सिस्टम को ज्यादा फ्लेक्सिबल बनाता है।
- 
- Processes get memory blocks according to their actual size.
  - Memory utilization is slightly better (compared to fixed size blocks).
  - Smaller blocks can be easily allocated to smaller processes.
  - This makes the system more flexible.



## Disadvantages of External Fragmentation:

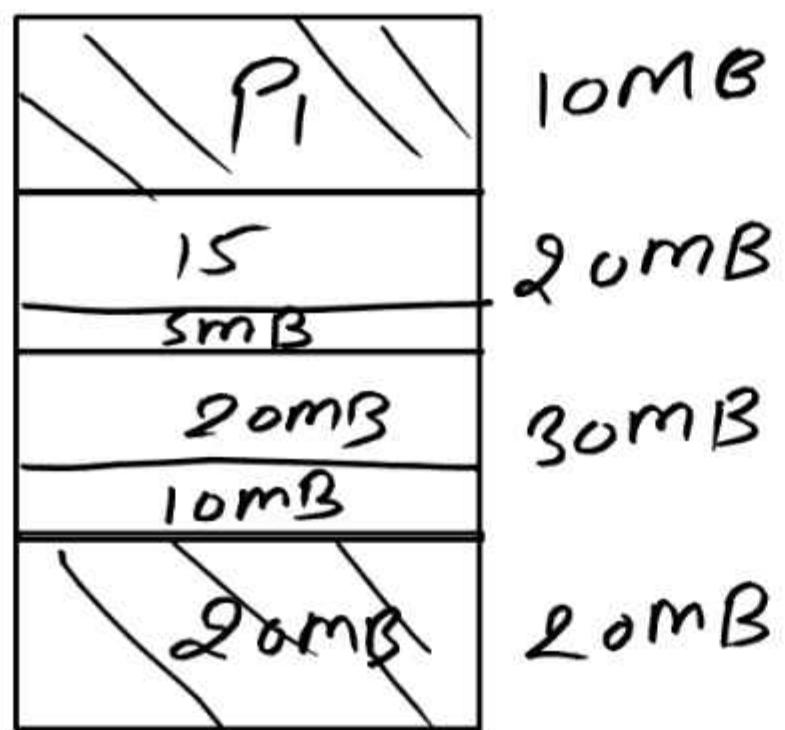
- छोटे-छोटे खाली हिस्से usable नहीं रहते।
- लगातार बड़ी मेमोरी ब्लॉक मिलना मुश्किल हो जाता है।
- मेमोरी का उपयोग पूरी तरह से नहीं हो पाता।
- कंप्लेक्स मेमोरी मैनेजमेंट की जरूरत होती है।
- परफॉर्मेंस धीमी हो सकती है क्योंकि memory compact करना पड़ता है।
- Small empty parts are not usable.
- It becomes difficult to get large memory blocks continuously.
- Memory is not fully utilized.
- Complex memory management is required.
- Performance may be slow because memory has to be compacted.



## Compaction:

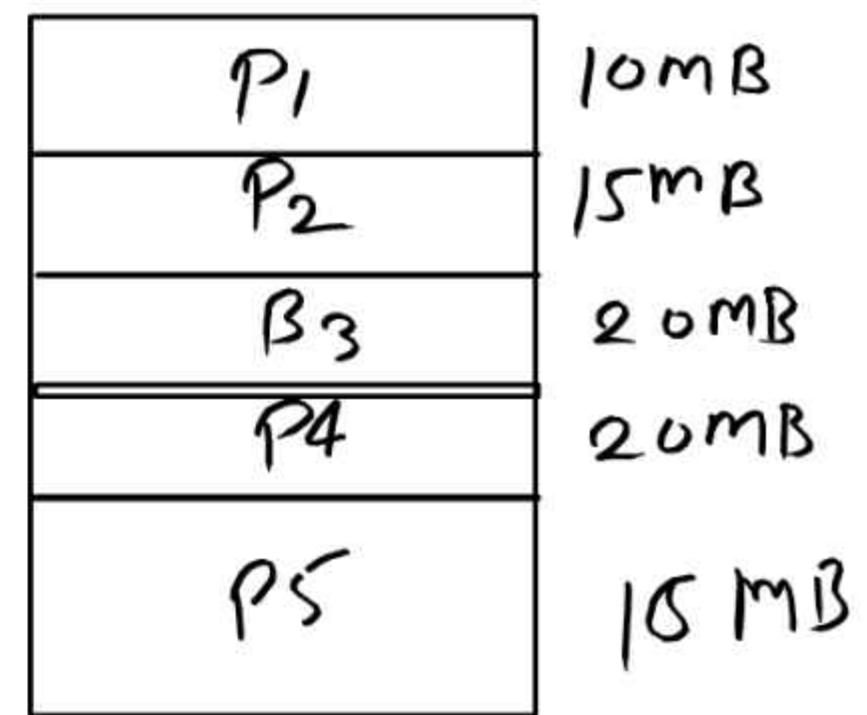
- Compaction एक मेमोरी मैनेजमेंट तकनीक है, जिसका उपयोग External Fragmentation को कम करने के लिए किया जाता है।
- जब मेमोरी में छोटे-छोटे खाली स्थान (Fragments) बच जाते हैं और किसी बड़े प्रोसेस को जगह नहीं मिलती, तो Compaction उन खाली स्थानों को एक साथ जोड़कर एक बड़ा खाली ब्लॉक बनाता है, ताकि नया प्रोसेस उसमें फिट हो सके।
- Compaction is a memory management technique used to reduce external fragmentation.
- When there are small empty spaces (fragments) left in the memory and no space is available for a large process, Compaction joins those empty spaces together to form a large empty block so that the new process can fit in it.

$P_1 \rightarrow 10MB$   
 $P_2 \rightarrow 15MB$   
 $P_3 \rightarrow 20MB$   
 $P_4 \rightarrow 20MB$   
 $P_5 \rightarrow 12MB$



$$10 + 5 = 15MB$$

Compaction





## वर्चुअल मेमोरी (Virtual Memory) :

- वर्चुअल मेमोरी एक तकनीक (technique) है जिसका उपयोग ऑपरेटिंग सिस्टम (OS) तब करता है जब फिजिकल रैम (RAM) पूरी भर जाती है। यह हार्ड डिस्क के एक हिस्से को अस्थायी रूप से रैम की तरह इस्तेमाल करके अधिक प्रोग्राम्स को चलाने की सुविधा देता है।
- Virtual memory is a technique that the operating system (OS) uses when the physical RAM is full. It allows more programs to run by temporarily using a portion of the hard disk as RAM.
- जब कंप्यूटर में उपलब्ध रैम (RAM) कम होती है, लेकिन हमें एक से अधिक बड़े प्रोग्राम चलाने होते हैं, तब वर्चुअल मेमोरी काम आती है।
- When the available RAM in the computer is less, but we have to run more than one large program, then virtual memory is useful.



## Advantages of Virtual Memory :

1. अधिक प्रोग्राम एक साथ चला सकते हैं (Can run more programs simultaneously)
  - अगर आपके कंप्यूटर में कम RAM है, तो भी आप एक साथ कई सॉफ्टवेयर और एप्लिकेशन चला सकते हैं।
  - Even if your computer has less RAM, you can run multiple software and applications simultaneously.
2. कंप्यूटर के हैंग होने से बचाता है (Prevents the computer from hacking)
  - अगर आप एक साथ कई भारी सॉफ्टवेयर (Heavy Software) का उपयोग कर रहे हैं, तो वर्चुअल मेमोरी बैकअप मेमोरी प्रदान करके उन्हें सुचारू रूप से चलाने में मदद करती है।
  - If you are using multiple heavy software simultaneously, virtual memory helps them run smoothly by providing backup memory.



### 3. RAM की सीमाओं को बढ़ाता है(Expands the limits of RAM)

- अगर आपके पास 4GB RAM है, लेकिन कोई सॉफ्टवेयर 8GB मेमोरी मांगता है, तो वर्चुअल मेमोरी हार्ड डिस्क का एक हिस्सा इस्तेमाल करके अतिरिक्त स्पेस प्रदान करती है।
- If you have 4GB of RAM, but a software requests 8GB of memory, virtual memory provides additional space by using a portion of the hard disk.



## Unit No 4.

### UNIT 4: I/O System.

- Mass storage structure - Overview
- Disk structure ✓
- Disk attachment
- Disk scheduling algorithms,
- swap space management ✓
- RAID types.



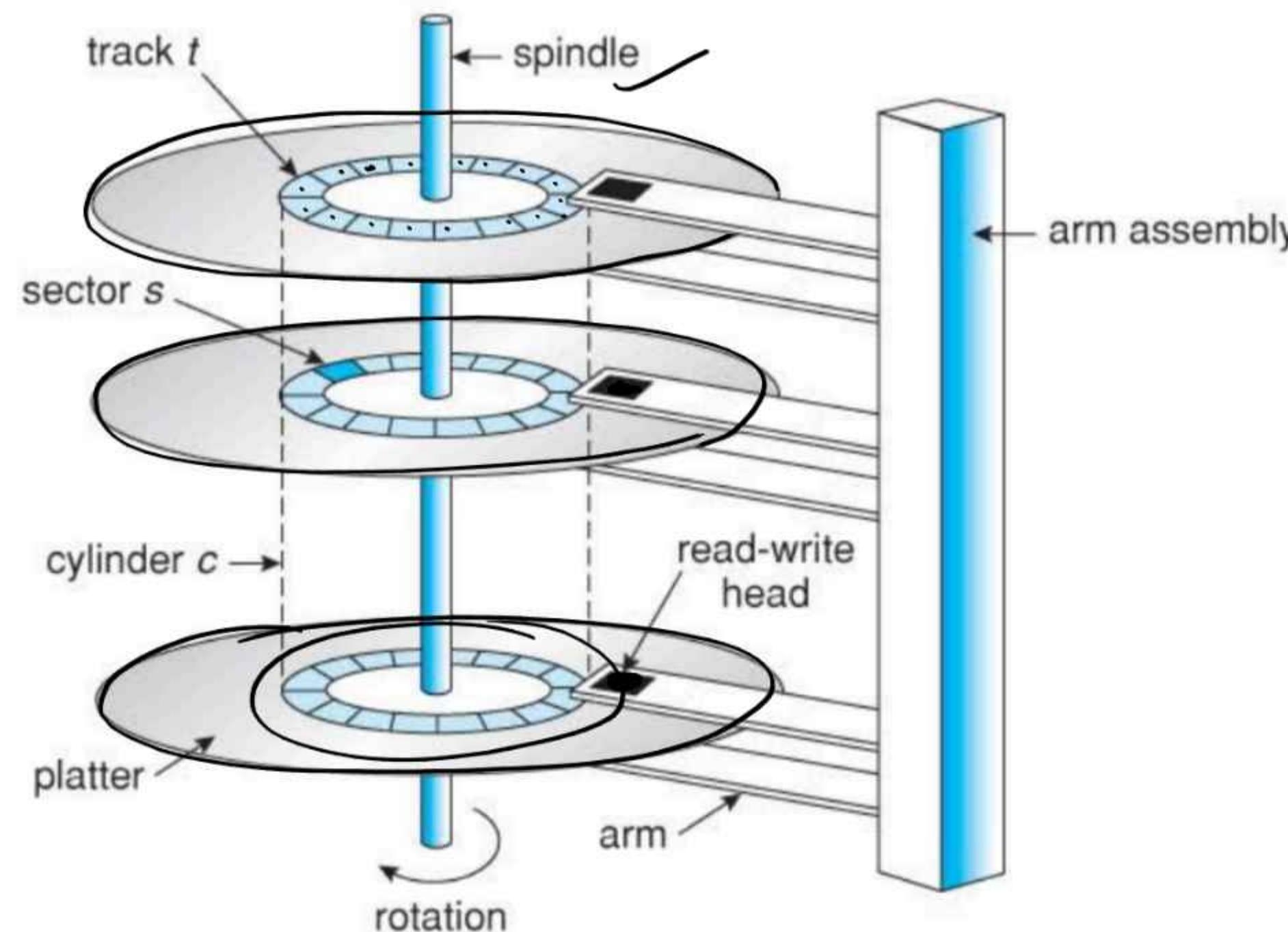
## Mass storage structure - Overview:

- Mass Storage Structure का मतलब है – ऐसे स्टोरेज डिवाइस (हार्ड डिस्क, SSD, मैग्नेटिक टेप, ऑप्टिकल डिस्क) जिन्हें स्थायी रूप से (Permanent) डेटा स्टोर करने के लिए इस्तेमाल किया जाता है।
- Mass Storage Structure means – storage devices (hard disk, SSD, magnetic tape, optical disk) that are used to store data permanently.
- Main Memory (RAM) वोलाटाइल होती है यानी बिजली जाने पर डेटा मिट जाता है।
- Main Memory (RAM) is volatile i.e. data is erased when the power goes off.
- लेकिन Mass Storage Devices Non-Volatile होती हैं यानी बिजली जाने पर भी डेटा सुरक्षित रहता है।
- But Mass Storage Devices are non-volatile i.e. data remains safe even when the power goes off.



## Magnetic Disk (मैग्नेटिक डिस्क) :

- Magnetic Disk एक Secondary Storage Device है, जिसका उपयोग डेटा को स्टोर करने के लिए किया जाता है। इसमें डेटा को मैग्नेटिक फॉर्म (चुंबकीय रूप) में स्टोर किया जाता है।
- Magnetic Disk is a secondary storage device, which is used to store data. In this, data is stored in magnetic form.
- यह डिस्क गोल (circular) होती है और इसमें कई सारे concentric circles होते हैं जिन्हें Track कहते हैं, और इन्हें छोटे टुकड़ों में बाँटा गया होता है जिन्हें Sector कहते हैं
- This disc is circular and consists of many concentric circles called tracks, which are divided into smaller pieces called sectors





## Component of Magnetic Disk:

### 1. Platter (प्लैटर):

- यह गोल डिस्क होती है जिस पर डेटा स्टोर किया जाता है। यह खास तरह के मैग्नेटिक मटेरियल से बना होता है। एक हार्ड डिस्क में एक से ज्यादा प्लैटर हो सकते हैं।
- It is a round disk on which data is stored. It is made of a special type of magnetic material. A hard disk can have more than one platter.

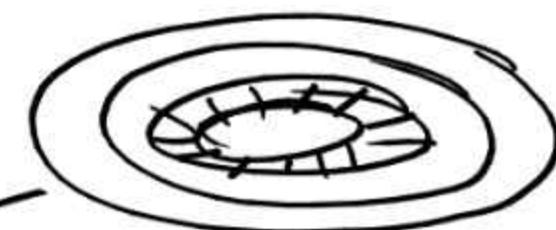
### 2. Spindle (स्पिंडल):

- यह एक धुरी (shaft) है जिस पर सारे प्लैटर लगे होते हैं। यह spindle, प्लैटर को high speed पर घुमाता है (rotation)। यह घुमाने का काम spindle motor करती है।
- This is a shaft on which all the platters are mounted. This spindle rotates the platters at high speed. This rotation is done by the spindle motor.



### 3. Track (ट्रैक):

- हर प्लैटर पर गोल-गोल कई concentric circles बने होते हैं, इन्हें track कहते हैं।
- यह वह रास्ता है जिस पर डेटा read या write होता है। एक प्लैटर पर हजारों ट्रैक हो सकते हैं।
- There are many concentric circles on each platter, these are called tracks.
- This is the path on which data is read or written. There can be thousands of tracks on a platter.



### 4. Sector (सेक्टर s):

- हर ट्रैक को छोटे-छोटे भागों में बँटा जाता है जिन्हें sectors कहते हैं। एक सेक्टर में आमतौर पर 512 bytes या 4 KB तक डेटा स्टोर होता है।
- Each track is divided into smaller parts called sectors. A sector usually stores up to 512 bytes or 4 KB of data.



### 5. Read-Write Head (रीड-राइट हेड):

- यह एक छोटा सा उपकरण है जो डेटा को प्लैटर से पढ़ता (read) या उस पर लिखता (write) है।
- हर प्लैटर के लिए कम से कम एक head होता है।
- It is a small device that reads or writes data from the platter.
- There is at least one head for each platter.

### 6. Arm (आर्म):

- यह एक mechanical हाथ है जो read-write head को अलग-अलग tracks तक ले जाता है।
- यह बहुत तेजी से move करता है ताकि डेटा जल्दी access किया जा सके।
- It is a mechanical arm that moves the read-write head to different tracks.
- It moves very fast so that data can be accessed quickly.



### Magnetic Disk के लाभ (Advantages):

- बड़ी मात्रा में डेटा स्टोर कर सकते हैं।
- सस्ता और reliable होता है।
- डेटा को तेजी से access किया जा सकता है।
- Rewritable – डेटा को कई बार बदला जा सकता है।
- Can store large amounts of data.
- It is cheap and reliable.
- Data can be accessed quickly
- Rewritable – Data can be changed many times.



### नुकसान (Disadvantages):

- मैकेनिकल पार्ट्स होने के कारण यह जल्दी खराब हो सकता है।
- SSD की तुलना में धीमा होता है।
- Physical shock (झटके) से डेटा खो सकता है।
- Due to having mechanical parts, it can get damaged quickly.
- It is slower than SSD.
- Data can be lost due to physical shock.



## Disk Attachment:

- Disk Attachment का मतलब है किसी secondary storage device (जैसे Hard Disk, SSD, CD/DVD, USB drive आदि) को computer system से जोड़ना ताकि CPU और Operating System उस disk से data पढ़ (read) और लिख (write) सकें।
- Disk Attachment means connecting a secondary storage device (such as Hard Disk, SSD, CD/DVD, USB drive, etc.) to a computer system so that the CPU and Operating System can read and write data from that disk.



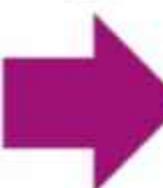
## Host-Attached Storage (HAS):

- Host-Attached Storage (HAS) एक ऐसा storage है जो सीधे किसी एक host (जैसे computer या server) से जुड़ा होता है। ये SATA, SCSI, IDE, USB जैसे interfaces के माध्यम से connect होता है और इसे manage करने का काम host का operating system करता है। इसका storage सिर्फ उसी host के लिए available रहता है।
- Host-Attached Storage (HAS) is a type of storage device that is directly attached to a single host (computer/server) using interfaces like SATA, SCSI, IDE, or USB, and is entirely managed by the host's operating system. It provides local storage that can only be accessed by that specific host.

SATA → (Hard disk, SSD) (Serial Advanced Technology Attachment)

SCSI (server) → (Small computer System interface)

IDE (motherboard) (Integrated Device Electronic)      USB → Universal Serial Bus



## Key Characteristics of HAS:

### 1: Direct Connection:

- Storage directly host system (computer/server) से जुड़ा होता है।

### 2: Single Host Access:

- Storage केवल उसी host के लिए accessible होता है।

### 3: OS Controlled:

- Storage management host के operating system द्वारा होता है।

Examples : Internal HDD, SSD, External USB drives.



## Advantages of HAS:

- आसान installation और configuration
  - Low cost solution
  - High speed (क्योंकि direct connection होता है)
  - Best for single-user or standalone systems
- 
- Easy installation and configuration
  - Low cost solution
  - High speed (because there is a direct connection)
  - Best for single-user or standalone systems



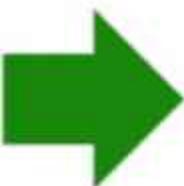
## Disadvantages of HAS:

- Multi-user access possible नहीं
- Scalability (capacity बढ़ाना) limited
- अगर host fail हो जाए → storage inaccessible हो जाएगा
- Multi-user access is not possible
- Scalability (increasing capacity) is limited
- If the host fails → storage will become inaccessible



## 2. Network-Attached Storage (NAS):

- Network-Attached Storage (NAS) एक special storage device होता है, जो network से जुड़ा रहता है और कई hosts या clients को एक साथ data (file-level access) provide करता है। इसे एक dedicated NAS operating system manage करता है और user इसे network protocols (जैसे NFS, SMB, FTP) के ज़रिए access करते हैं।
- Network-Attached Storage (NAS) is a specialized storage device that connects to a network and provides file-level data access to multiple hosts or clients simultaneously. It is managed by a dedicated NAS operating system and accessed using standard network protocols such as NFS, SMB/CIFS, or FTP.



## Key Characteristics of NAS:

### 1: Network Based Storage :

- NAS storage LAN (Local Area Network) से जुड़ा रहता है।

### 2: Multiple Client Access:

- एक साथ कई users या hosts इसे access कर सकते हैं।

### 3: File-Level Access:

- NAS users को file system की तरह storage provide करता है।

### 4: Protocols Used:

- Commonly NFS (Linux/Unix), SMB/CIFS (Windows), FTP

### 5: Dedicated OS:

- इसमें खुद का छोटा dedicated operating system होता है storage manage करने के लिए।



## Advantages of NAS:

- Centralized storage (सभी data एक जगह store)
- Multiple users को एक साथ access करने की सुविधा
- Easy backup और data sharing
- Cheaper than SAN (Storage Area Network)
- Expand करना आसान (add more disks)
- Centralized storage (all data stored in one place)
- Facility for multiple users to access simultaneously
- Easy backup and data sharing
- Cheaper than SAN (Storage Area Network)
- Easy to expand (add more disks)



## Disadvantages of NAS:

- Network पर depend करता है → अगर network slow है तो access भी slow होगा
- File-level storage only (block-level नहीं)
- बहुत बड़े organizations के लिए limited performance (SAN से slower)
- Depends on the network → if the network is slow then access will also be slow
- File-level storage only (not block-level)
- Limited performance for very large organizations (slower than SAN)



## Disk Scheduling :

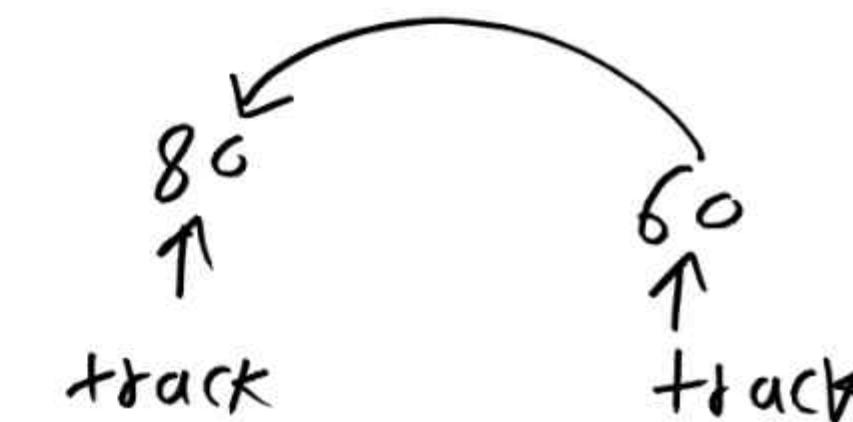
- Disk Scheduling एक ऐसी प्रक्रिया है जिसका प्रयोग ऑपरेटिंग सिस्टम (Operating System) द्वारा तब किया जाता है जब एक से अधिक इनपुट/आउटपुट (I/O) अनुरोध एक साथ डिस्क ड्राइव को प्राप्त होते हैं। चूंकि डिस्क ड्राइव एक समय में केवल एक ही अनुरोध को प्रोसेस कर सकती है, इसलिए आवश्यक होता है कि उन सभी अनुरोधों को एक विशेष क्रम (order) में निष्पादित किया जाए,
- Disk scheduling is a process used by the operating system when multiple input/output (I/O) requests are received by the disk drive simultaneously. Since the disk drive can process only one request at a time, it is necessary that all those requests be executed in a particular order,

$a \rightarrow$   
 $\nwarrow FCFSS \rightarrow$   
 $\downarrow SCAN, C-SCAN$   
 $\downarrow LOOK, C-LOOK$



### Purpose:

- Disk arm की movement को कम किया जाए
- Disk arm movement should be reduced
- Seek time को minimize किया जाए
- Seek time should be minimized
- Throughput को बढ़ाया जाए
- Throughput should be increased
- System performance को बेहतर बनाया जाए
- System performance should be improved



\*

$$\text{Seek Time} = \text{total track} * \text{time Per track}$$



## → FCFS Disk Scheduling:

- FCFS (First-Come, First-Served) Disk Scheduling एक सरल एल्गोरिद्धि है जिसमें डिस्क के I/O अनुरोधों को उसी क्रम में पूरा किया जाता है जिस क्रम में वे सिस्टम में आते हैं, बिना किसी प्राथमिकता या दूरी की गणना के।
- FCFS (First-Come, First-Served) Disk Scheduling is a simple algorithm in which I/O requests to a disk are served in the order they arrive into the system, without any priority or distance calculation.



## Question:

- Disk head की वर्तमान स्थिति = 53
- Disk Requests = [98, 183, 37, 122, 14, 124, 65, 67]
- Calculate Total head movement ?



## SSTF Disk Scheduling:

- SSTF (Shortest Seek Time First) Disk Scheduling एक ऐसा एल्गोरिद्धि है जिसमें डिस्क हेड उस अनुरोध (request) को पहले प्रोसेस करता है जो वर्तमान स्थिति से सबसे कम दूरी पर होता है, अर्थात् जिसके लिए seek time सबसे कम होता है।
- SSTF (Shortest Seek Time First) Disk Scheduling is an algorithm in which the disk head first processes the request that is at the shortest distance from the current position, i.e. for which the seek time is the shortest.



## Question:

- Disk head की वर्तमान स्थिति = 53
- Disk Requests = [98, 183, 37, 122, 14, 124, 65, 67]
- Calculate Total head movement ?



## SSTF के लाभ (Advantages):

- Head movement और seek time को कम करता है
- Reduces head movement and seek time
- System performance को improve करता है
- Improves system performance

## SSTF के नुकसान (Disadvantages):

- Starvation की समस्या हो सकती है (दूर के request लंबे समय तक pending रह सकते हैं)
- Starvation may be a problem (remote requests may remain pending for a long time)
- Implementation थोड़ा complex होता है FCFS की तुलना में
- Implementation is a bit complex compared to FCFS



## SCAN Disk (Elevator) Scheduling:

- **SCAN Disk Scheduling** एल्गोरिद्धि में डिस्क हेड एक दिशा में लगातार ट्रैकों को स्कैन करता है और जहाँ भी कोई अनुरोध मिलता है, उसे सर्व करता है। जब हेड अंतिम ट्रैक तक पहुँच जाता है, तो वह विपरीत दिशा में घूमता है और फिर से उसी तरह काम करता है।
- In the SCAN Disk Scheduling algorithm, the disk head continuously scans the tracks in one direction and serves requests wherever they are found. When the head reaches the last track, it rotates in the opposite direction and does the same thing again.





## Question:

- Disk head की वर्तमान स्थिति = 53
- Disk Requests = [98, 183, 37, 122, 14, 124, 65, 67]
- Calculate Total head movement ?
- Direction : Head moves towards higher tracks



## लाभ (Advantages):

- Starvation की संभावना नहीं है
- There is no chance of starvation
- SSTF की तुलना में head movement ज्यादा हो सकती है, पर fairness बनी रहती है
- There may be more head movement compared to SSTF, but fairness is maintained

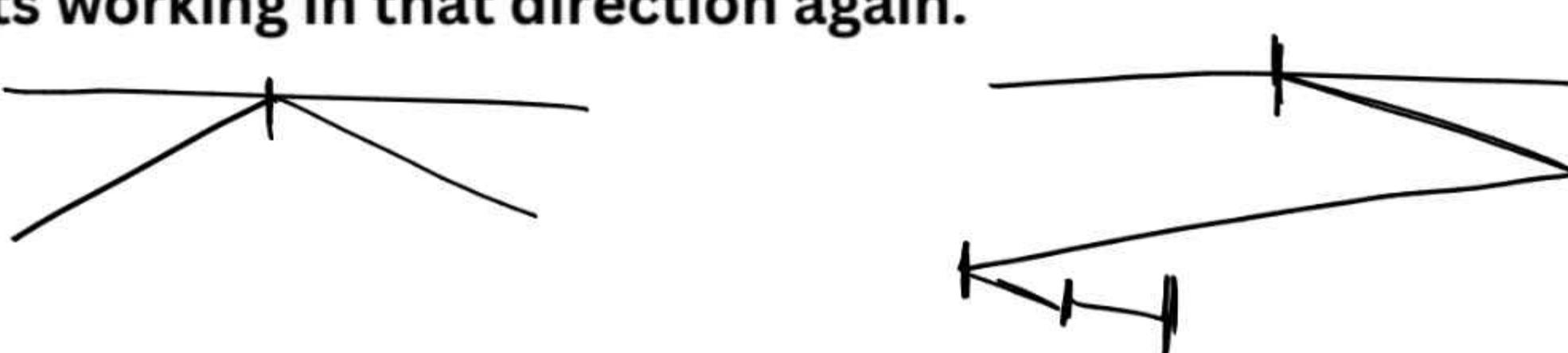
## नुकसान (Disadvantages):

- Head movement कभी-कभी ज्यादा हो सकती है
- Head movement can sometimes be excessive
- एक direction के request को पहले serve किया जाता है
- Requests for one direction are served first



## C-SCAN Disk Scheduling:

- C-SCAN (Circular SCAN) Disk Scheduling एक एल्गोरिद्धि है जिसमें डिस्क हेड एक दिशा में सभी अनुरोधों को सर्व करता है, और जब वह अंतिम ट्रैक तक पहुँच जाता है, तो सीधे शुरुआत (पहले ट्रैक) पर लौट आता है – बिना किसी सर्व किए – और फिर से उसी दिशा में काम करना शुरू करता है।
- C-SCAN (Circular SCAN) Disk Scheduling is an algorithm in which the disk head serves all requests in one direction, and when it reaches the last track, returns directly to the beginning (first track) – without serving anything – and starts working in that direction again.





## Question:

- Disk head की वर्तमान स्थिति = 53
- Disk Requests = [98, 183, 37, 122, 14, 124, 65, 67]
- Calculate Total head movement ?
- Direction : Head moves towards higher tracks



### लाभ (Advantages):

- Fairness – सभी requests को equal time मिलता है।
- Fairness – All requests get equal time.
- Starvation नहीं होती।
- There is no starvation.

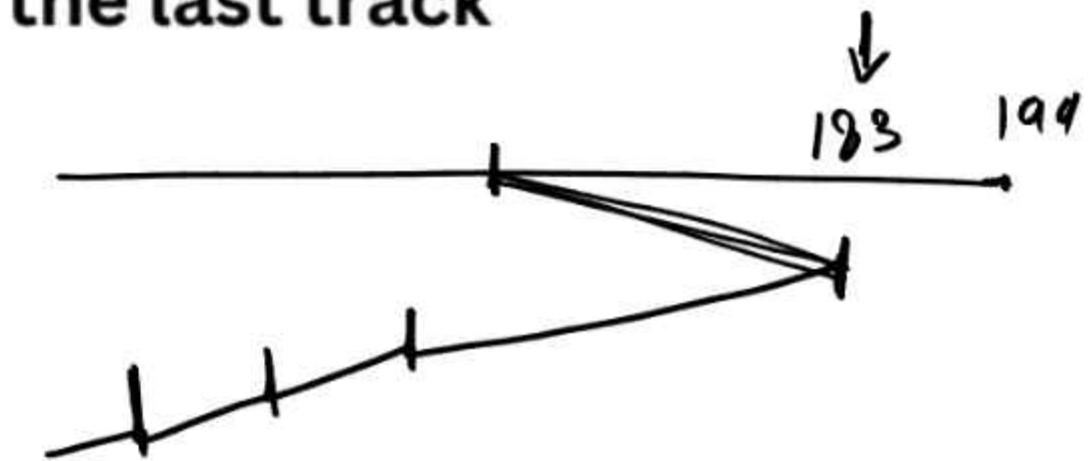
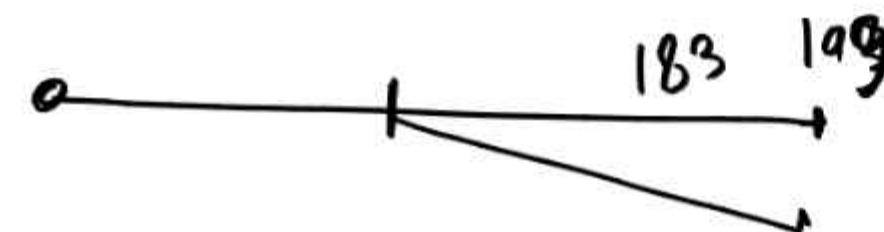
### नुकसान (Disadvantages):

- Return में जो jump होता है, वह अतिरिक्त head movement जोड़ता है।
- The jump that happens during return adds extra head movement.
- कभी-कभी थोड़ा कम efficient हो सकता है।
- Can sometimes be a little less efficient



## LOOK Disk Scheduling:

- LOOK Disk Scheduling एक ऐसा एल्गोरिद्धि है जिसमें डिस्क हेड केवल उन ट्रैकों तक जाता है जहाँ तक I/O अनुरोध मौजूद होते हैं, और फिर दूसरी दिशा में घूमकर बाकी बचे अनुरोधों को पूरा करता है। यह एल्गोरिद्धि SCAN का ही एक बेहतर रूप है जिसमें हेड बेवजह आखिरी ट्रैक तक नहीं जाता।
- LOOK Disk Scheduling is an algorithm in which the disk head travels only to those tracks where there are I/O requests, and then moves in the other direction to fulfill the remaining requests. This algorithm is an improved form of SCAN in which the head does not travel to the last track unnecessarily.





## Question:

- Disk head की वर्तमान स्थिति = 53
- Disk Requests = [98, 183, 37, 122, 14, 124, 65, 67]
- Calculate Total head movement ?
- Direction : Head moves towards higher tracks



## लाभ (Advantages):

- Unnecessary Head Movement नहीं होती
- There is no unnecessary head movement
- SCAN से बेहतर Performance
- Better performance than SCAN
- Starvation नहीं होती
- There is no starvation.

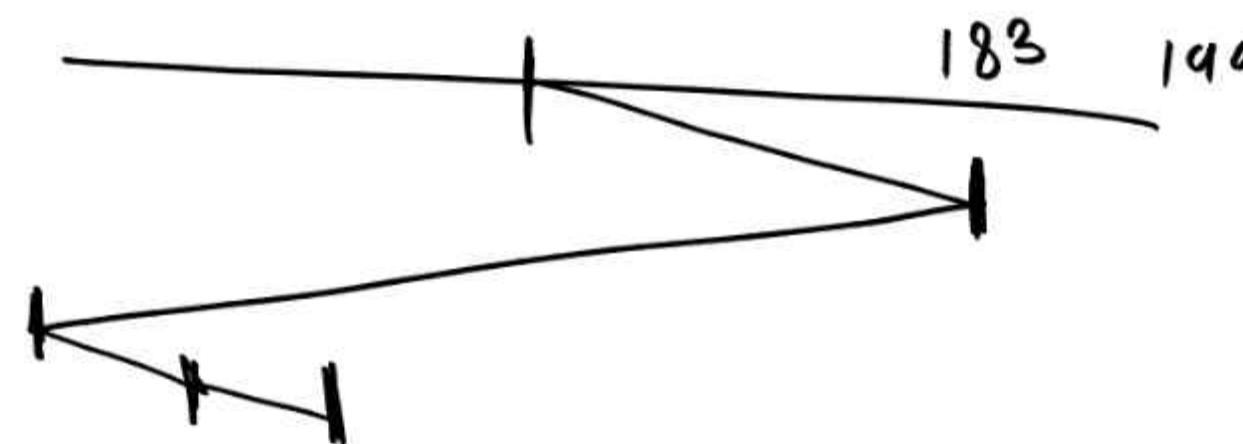
## नुकसान (Disadvantages):

- थोड़ा Complex Algorithm है
- It is a little complex algorithm
- हर बार Direction बदलना पड़ता है
- Direction has to be changed every time



### C-LOOK Disk Scheduling:

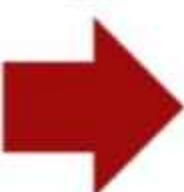
- C-LOOK एक Disk Scheduling Algorithm है जिसमें हेड सिर्फ उन ट्रैकों तक जाता है जहाँ I/O requests होती हैं (जैसे LOOK में), लेकिन जब वह आखिरी request तक पहुँच जाता है, तो वह सीधे सबसे पहली request पर jump करता है – बिना बीच के ट्रैकों को चेक किए
- C-LOOK is a disk scheduling algorithm in which the head moves only to tracks that have I/O requests (as in LOOK), but when it reaches the last request, it jumps directly to the first request – without checking the intermediate tracks.





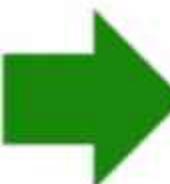
## Question:

- Disk head की वर्तमान स्थिति = 53
- Disk Requests = [98, 183, 37, 122, 14, 124, 65, 67]
- Calculate Total head movement ?
- Direction : Head moves towards higher tracks



## Swap Space Management:

- Swap Space वह disk area (secondary storage) है जिसे ऑपरेटिंग सिस्टम main memory (RAM) का extension बनाने के लिए उपयोग करता है।
- Swap Space is the disk area (secondary storage) that the operating system uses to create an extension of the main memory (RAM).
- जब RAM में जगह कम पड़ जाती है और running processes के लिए memory allocate करना मुश्किल होता है, तो OS कुछ inactive या less-used pages को RAM से निकालकर swap space (hard disk पर reserved area) में भेज देता है। इस process को Swapping कहते हैं।
- When there is insufficient space in RAM and it becomes difficult to allocate memory for running processes, the OS removes some inactive or less-used pages from RAM and sends them to swap space (reserved area on the hard disk). This process is called swapping.



## Key Points of Swap Space:

- Swap space secondary storage (जैसे hard disk/SSD) का एक हिस्सा होता है।
- Swap space is a part of secondary storage (such as hard disk/SSD).
- यह RAM का extension की तरह काम करता है।
- It acts as an extension of RAM.
- जब RAM पूरी भर जाती है, तो unused memory pages swap space में move हो जाते हैं।
- When RAM is full, unused memory pages are moved to swap space.
- Swap space RAM से slow होता है, लेकिन memory shortage की problem solve करता है।
- Swap space is slower than RAM but solves the problem of memory shortage.
- Swap space को manage करने के लिए OS paging और virtual memory techniques use करता है।
- The OS uses paging and virtual memory techniques to manage swap space.



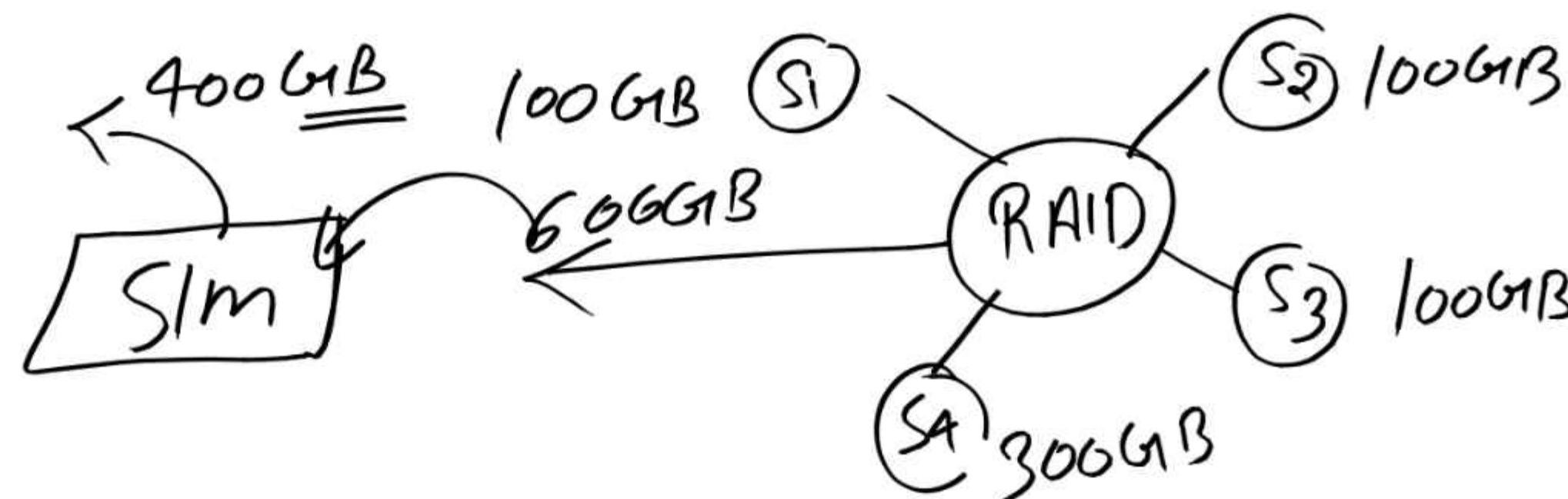
### Advantages of Swap Space:

- Programs को run करने के लिए extra memory मिलती है।
- Low RAM वाले system भी बड़े applications चला सकते हैं।
- Multiprogramming और multitasking आसान हो जाता है।
- System crash होने से बच जाता है जब RAM full हो जाती है।
- Extra memory is available to run programs.
- Systems with low RAM can also run large applications.
- Multiprogramming and multitasking become easier.
- The system is saved from crashing when the RAM becomes full.



## RAID:

- RAID (Redundant Array of Independent/Inexpensive Disks) एक data storage virtualization technology है जिसमें कई physical hard disks को logically combine किया जाता है ताकि system की performance, fault tolerance और data reliability को improve किया जा सके।
- RAID (Redundant Array of Independent/Inexpensive Disks) is a data storage virtualization technology in which multiple physical hard disks are logically combined to improve system performance, fault tolerance, and data reliability.

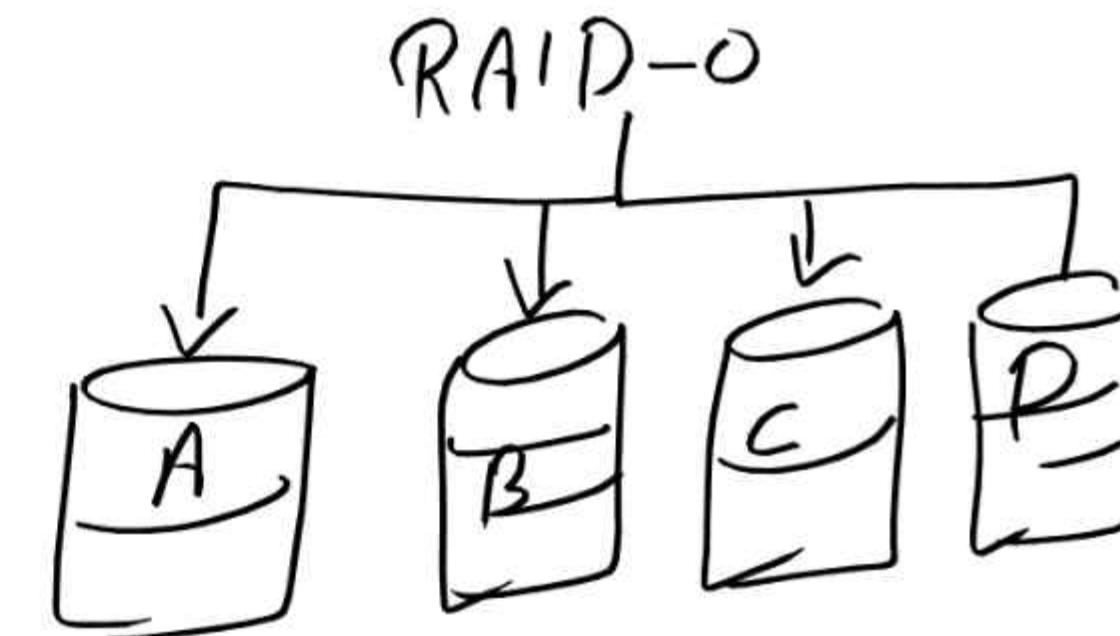
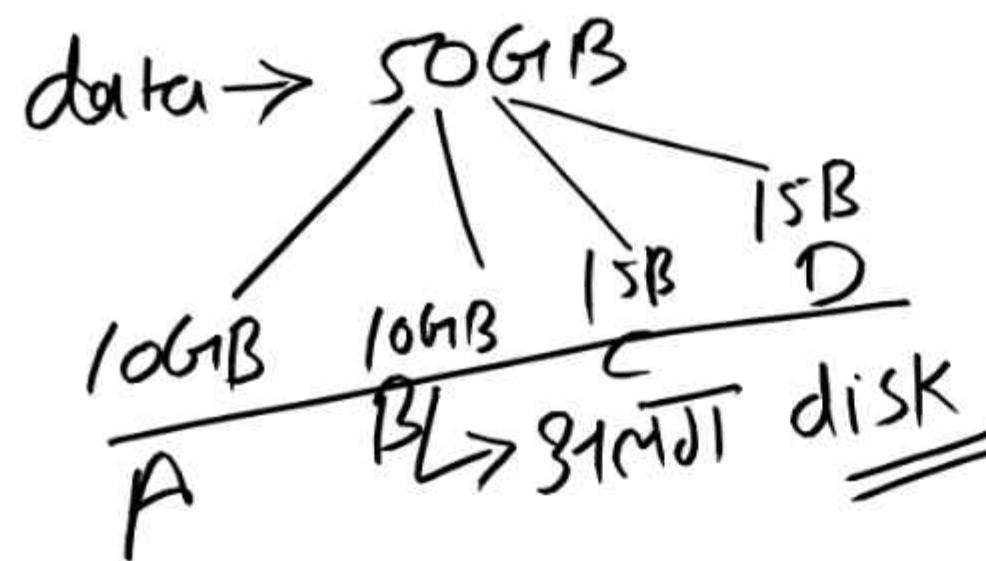




## RAID Type:

### 1: RAID 0 (Disk Striping):

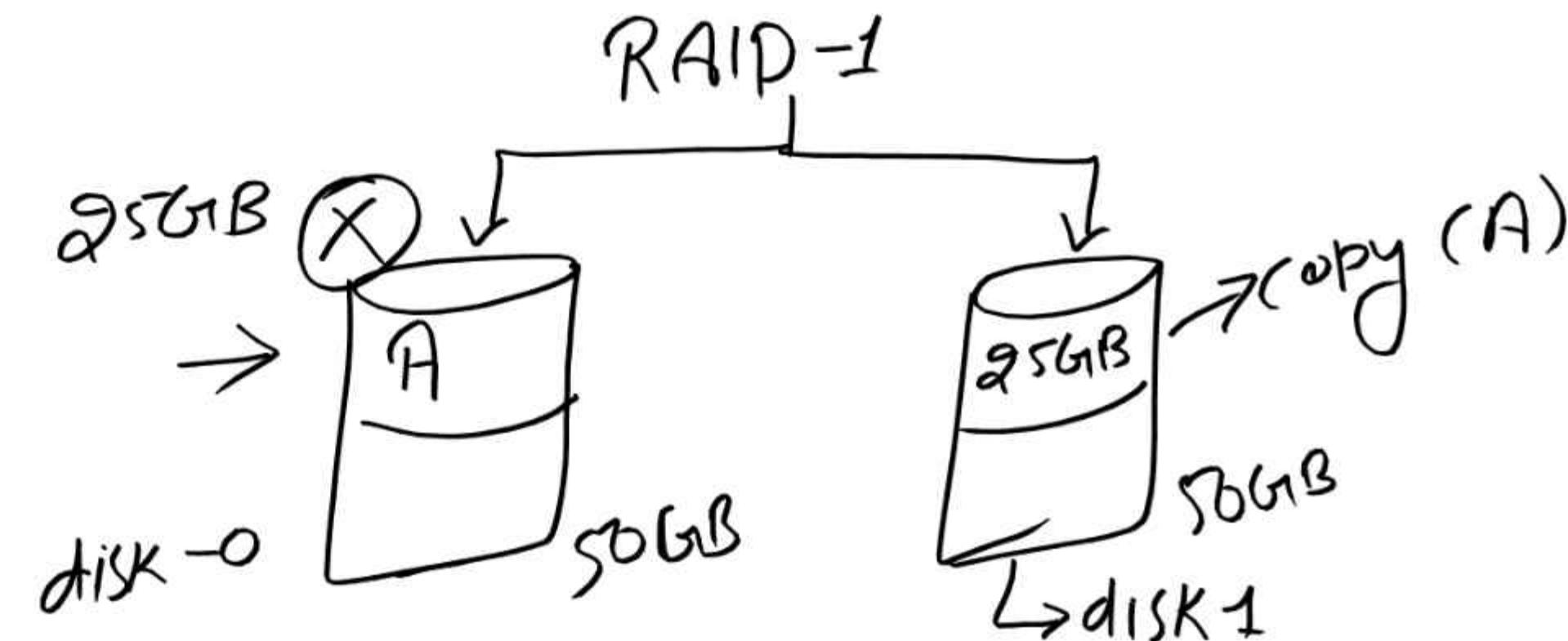
- RAID 0 में data को blocks में divide करके कई disks पर एक साथ store किया जाता है। इससे system की read/write performance बढ़ती है, लेकिन इसमें fault tolerance नहीं होता।
- In RAID 0, data is divided into blocks and stored simultaneously on multiple disks. This increases the read/write performance of the system, but there is no fault tolerance.





## 2: RAID 1 (Disk Mirroring)

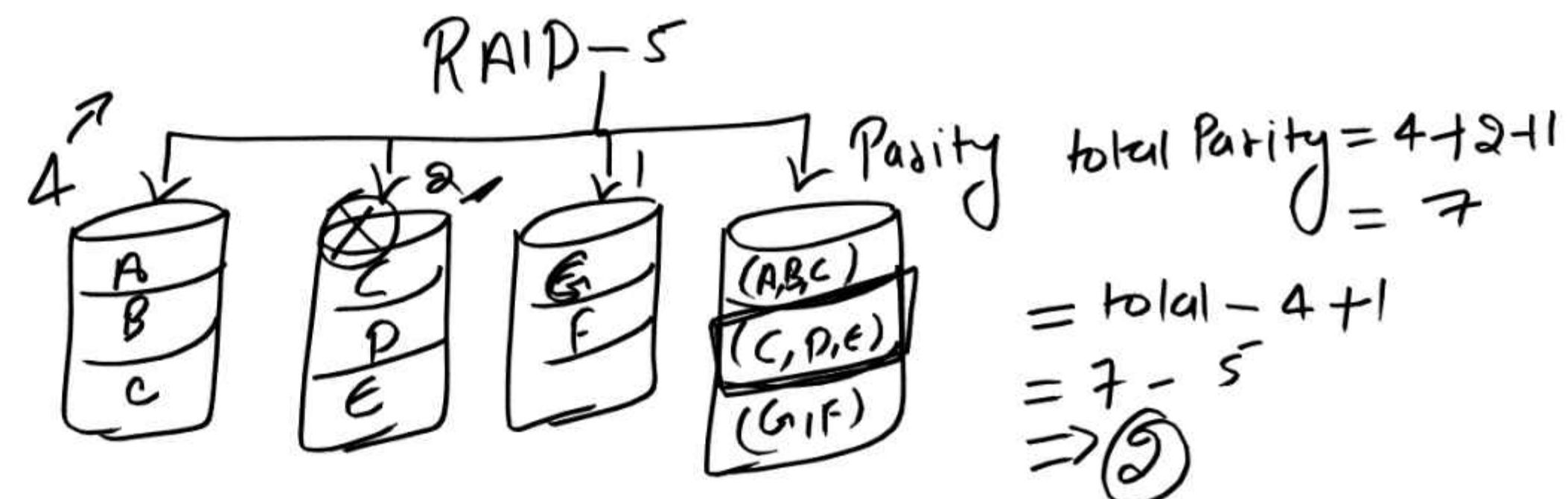
- RAID 1 में हर data की exact copy दूसरी disk पर बनाई जाती है। अगर एक disk fail हो जाए तो दूसरी disk से data recover किया जा सकता है। इससे **high reliability** मिलती है लेकिन **storage capacity** आधी हो जाती है
- In RAID 1, an exact copy of every data is made on the second disk. If one disk fails, data can be recovered from the second disk. This provides high reliability but the storage capacity is halved.





### 3: RAID 5 (Striping with Single Parity)

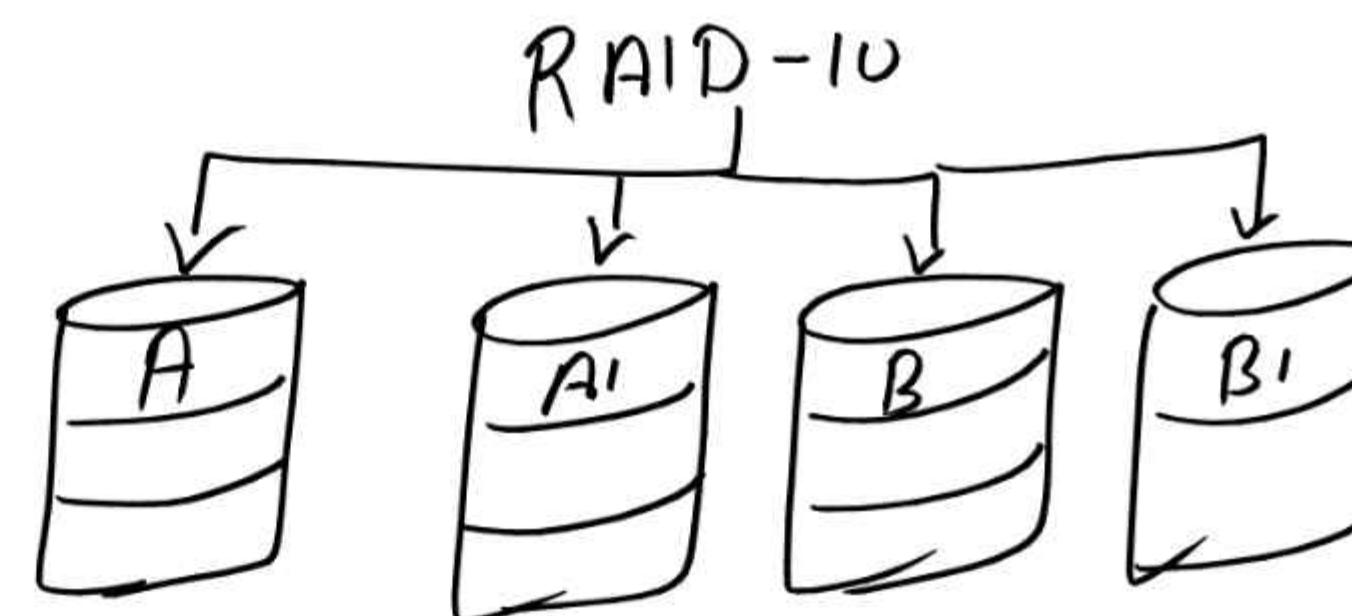
- RAID 5 में data को multiple disks पर divide किया जाता है और साथ ही एक parity block भी store होता है। अगर कोई एक disk fail हो जाए तो उस parity block से lost data को reconstruct किया जा सकता है। यह speed और fault tolerance का अच्छा balance देता है।
- In RAID 5, data is divided across multiple disks and a parity block is also stored. If one disk fails, the lost data can be reconstructed from that parity block. This gives a good balance of speed and fault tolerance.





### RAID 10 (Combination of RAID 1 + RAID 0):

- RAID 10 RAID 1 और RAID 0 का combination है। इसमें पहले data की mirroring की जाती है और फिर उसे striping के रूप में distribute किया जाता है। इसका फायदा है कि system को high speed और high reliability दोनों मिलते हैं, लेकिन इसके लिए कम से कम 4 disks की ज़रूरत होती है
- RAID 10 is a combination of RAID 1 and RAID 0. In this, the data is first mirrored and then distributed in the form of striping. Its advantage is that the system gets both high speed and high reliability, but it requires at least 4 disks.





## Unit No 5 - File management:

## UNIT 5: File management:

Concept of a file, access methods.

- file system structure and implementation, Different types of file systems, free-space management, efficiency and performance.

• directory structure, directory implementation.

⇒ 10 marks

• file system mounting.

• file sharing and protection.



### File (फाइल):

- “फाइल एक नामित संग्रह (named collection) होती है, जिसमें संबंधित डाटा या सूचना को स्टोर किया जाता है, ताकि उसे बाद में उपयोग किया जा सके। फाइल्स कंप्यूटर की स्टोरेज डिवाइस (जैसे हार्ड डिस्क) में सेव की जाती हैं और इनका एक्सेस फाइल नेम के माध्यम से किया जाता है।”
- “A file is a named collection in which related data or information is stored so that it can be used later. Files are saved in a computer's storage device (such as a hard disk) and are accessed through the file name.”





## Types of Files :

### 1. Text File (टेक्स्ट फाइल):

- Text File एक ऐसी फाइल होती है जिसमें केवल अक्षरों, शब्दों और वाक्यों के रूप में साधारण टेक्स्ट डाटा स्टोर किया जाता है।
- A Text File is a file in which only simple text data is stored in the form of letters, words and sentences.
- Extension: .txt, .doc, .docx, .rtf

### 2. Binary File (बाइनरी फाइल):

- Binary File वह फाइल होती है जिसमें डाटा को बाइनरी (0 और 1) फॉर्मेट में स्टोर किया जाता है, जिसे केवल विशेष प्रोग्राम ही पढ़ सकते हैं।
- Binary File is a file in which data is stored in binary (0 and 1) format, which can only be read by special programs.
- Extension: .bin, .dat



### 3. Image File (चित्र फाइल):

- Image File एक ग्राफ़िकल फाइल होती है जिसमें image या विज़ुअल डाटा स्टोर किया जाता है।
- An image file is a graphical file in which images or visual data is stored.
- Extension: .jpg(Joint Photographic Experts Group)
- .png( Portable Network Graphics), .gif( Graphics Interchange Format),
- .bmp(Bitmap Picture) etc.

### 4. Audio File (ऑडियो फाइल):

- Audio File वह फाइल होती है जिसमें ध्वनि (साउंड) या म्यूजिक रिकॉर्ड होता है, जिसे ऑडियो प्लेयर के माध्यम से सुना जा सकता है।
- An Audio File is a file that contains sound or music that can be heard through an audio player.
- Extension: .mp3, .wav, .aac, .flac



### 5. Video File (वीडियो फाइल):

- Video File वह फाइल होती है जिसमें चलती हुई तस्वीरें (motion pictures), ध्वनि के साथ स्टोर की जाती हैं।
- A video file is a file that stores motion pictures along with sound.
- Extension: .mp4, .avi, .mkv, .mov

### 6. Executable File (एक्जिक्यूटेबल फाइल):

- Executable File एक प्रकार की फाइल होती है जिसे डबल क्लिक करके रन किया जाता है और वह कोई विशेष प्रोग्राम या निर्देश (Instruction) चलाता है।
- Executable File is a type of file which is run by double clicking and it runs a specific program or instruction.
- Extension: .exe, .bat, .msi, .sh



### 7. Compressed File (कम्प्रेस्ड फाइल):

- Compressed File वह होती है जिसमें एक या एक से अधिक फाइलों को छोटे साइज में संग्रहित किया जाता है, ताकि उन्हें आसानी से ट्रांसफर या स्टोर किया जा सके।
- A compressed file is one in which one or more files are stored in a smaller size so that they can be easily transferred or stored.
- Extension: .zip, .rar, .7z, .tar

### 8. Database File (डेटाबेस फाइल):

- Database File वह फाइल होती है जिसमें टेब्ल, रिकॉर्ड और क्वेरी आदि के रूप में संरचित (Structured) डाटा स्टोर किया जाता है।
- Database File is a file in which structured data is stored in the form of tables, records and queries etc.
- Extension: .db, .mdb, .sqlite, .accdb



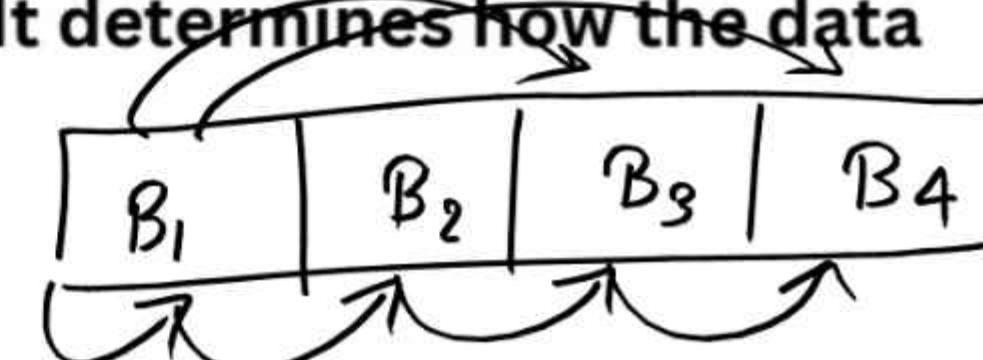
## File Access Method (फाइल एक्सेस विधियाँ) :

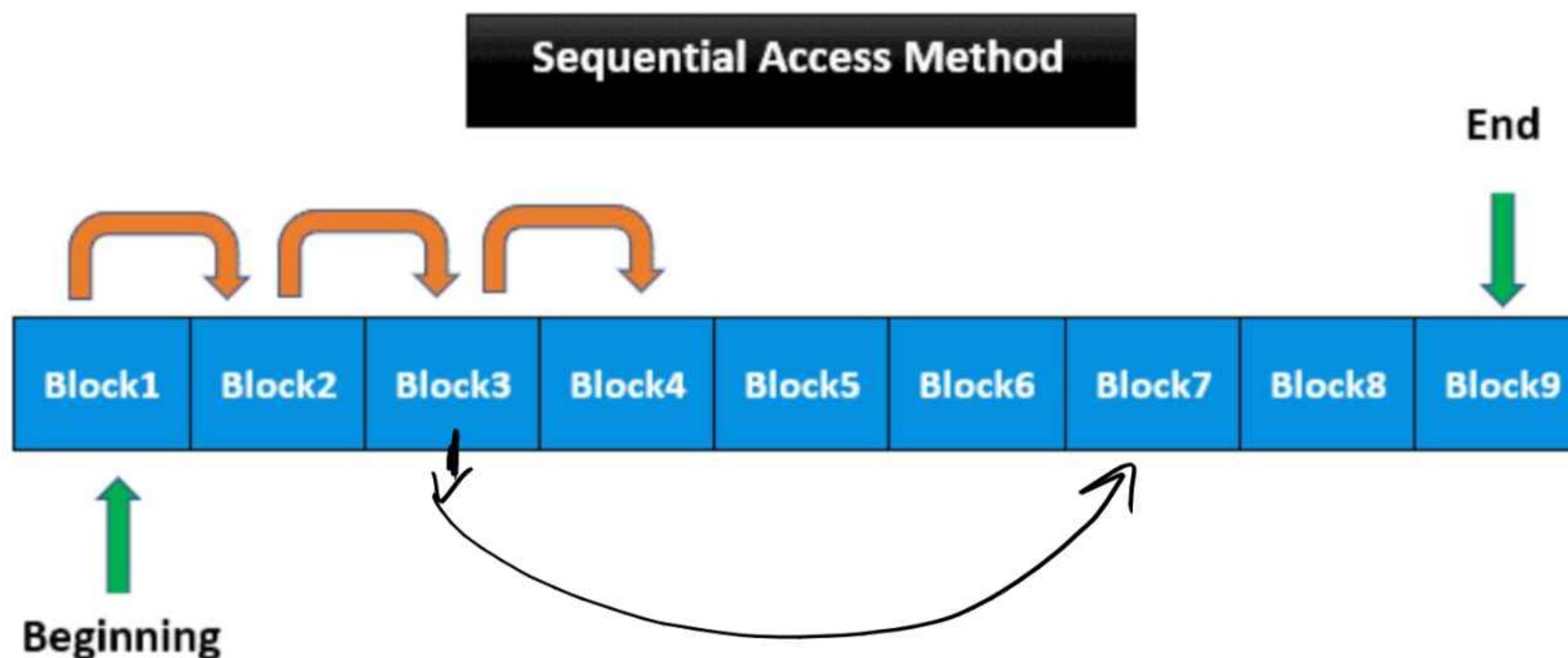
- File Access Method वह तकनीक है जिससे ऑपरेटिंग सिस्टम किसी फाइल के अंदर मौजूद डाटा को पढ़ता (Read), लिखता (Write), या उपयोग करता (Access) है। यह निर्धारित करता है कि फाइल के डाटा तक कैसे पहुँचा जाएगा।
- File Access Method is the technique by which the operating system reads, writes, or accesses the data contained within a file. It determines how the data in the file will be accessed.

### Types of File Access Methods):

#### 1. Sequential Access (अनुक्रमिक पहुँच विधि)

- इस विधि में डाटा को फाइल में एक के बाद एक क्रम में पढ़ा या लिखा जाता है। आप पहले रिकॉर्ड से शुरू करते हैं और एक-एक करके अगले रिकॉर्ड तक पहुँचते हैं।
- In this method data is read or written into the file sequentially. You start with the first record and move to the next record one by one.







## Advantages (फायदे):

- सरल और आसान प्रक्रिया (Simple and easy process).
- बैकअप और बड़े डेटा प्रोसेसिंग के लिए उपयुक्त
- Suitable for backup and large data processing
- Memory कम लगती है (Consumes less memory)

## Disadvantages (फायदे):

- किसी खास रिकॉर्ड तक पहुँचने के लिए पूरा डाटा पढ़ना पड़ता है
- To access a particular record, the entire data has to be read
- समय अधिक लगता है
- It takes more time

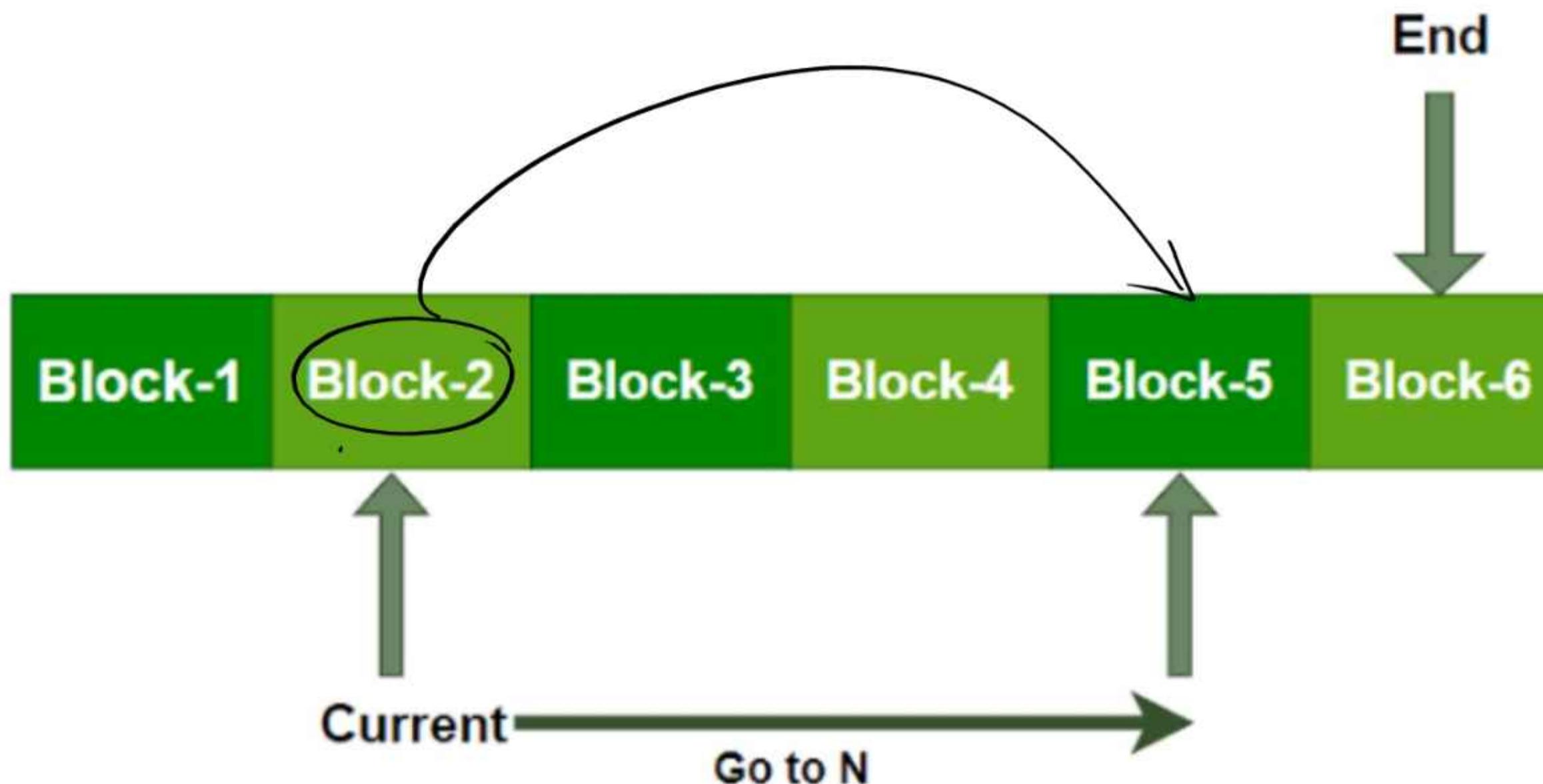


## 2. Direct Access (प्रत्यक्ष पहुँच विधि):

- इस विधि में आप किसी भी रिकॉर्ड या डेटा को सीधे एक्सेस कर सकते हैं, बिना सभी रिकॉर्ड पढ़े।
- In this method you can access any record or data directly without reading all the records.



## Direct Access Method





## Advantages (फायदे):

- तेज़ और तुरंत एक्सेस
- Fast and instant access.
- Random रिकॉर्ड को पढ़ना और अपडेट करना आसान
- Easy to read and update random records

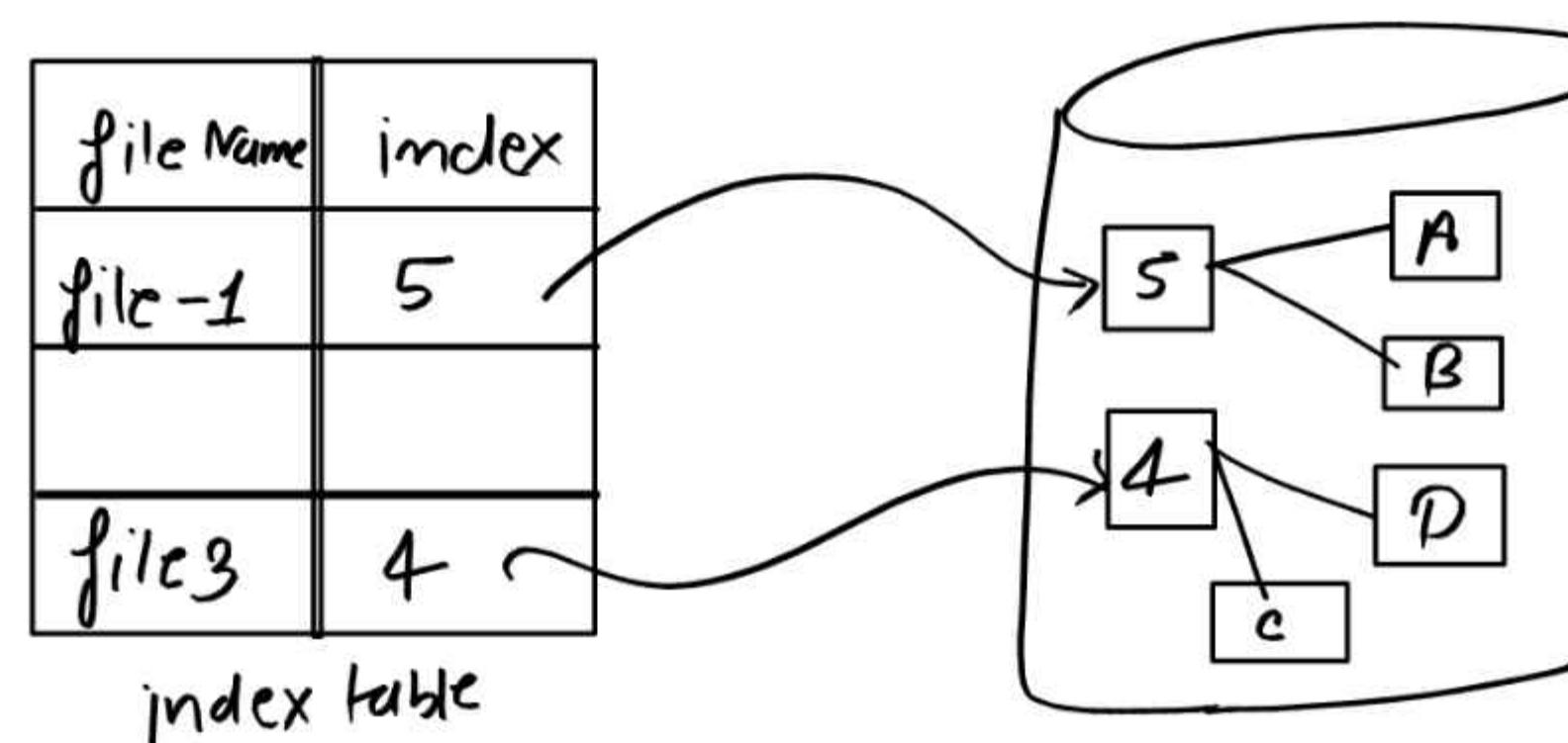
## Disadvantages (कमियाँ):

- फाइल को Structured तरीके से स्टोर करना जरूरी होता है
- Files need to be stored in a structured manner
- कोडिंग और प्रोग्रामिंग अधिक जटिल हो सकती है
- Coding and programming can be more complex



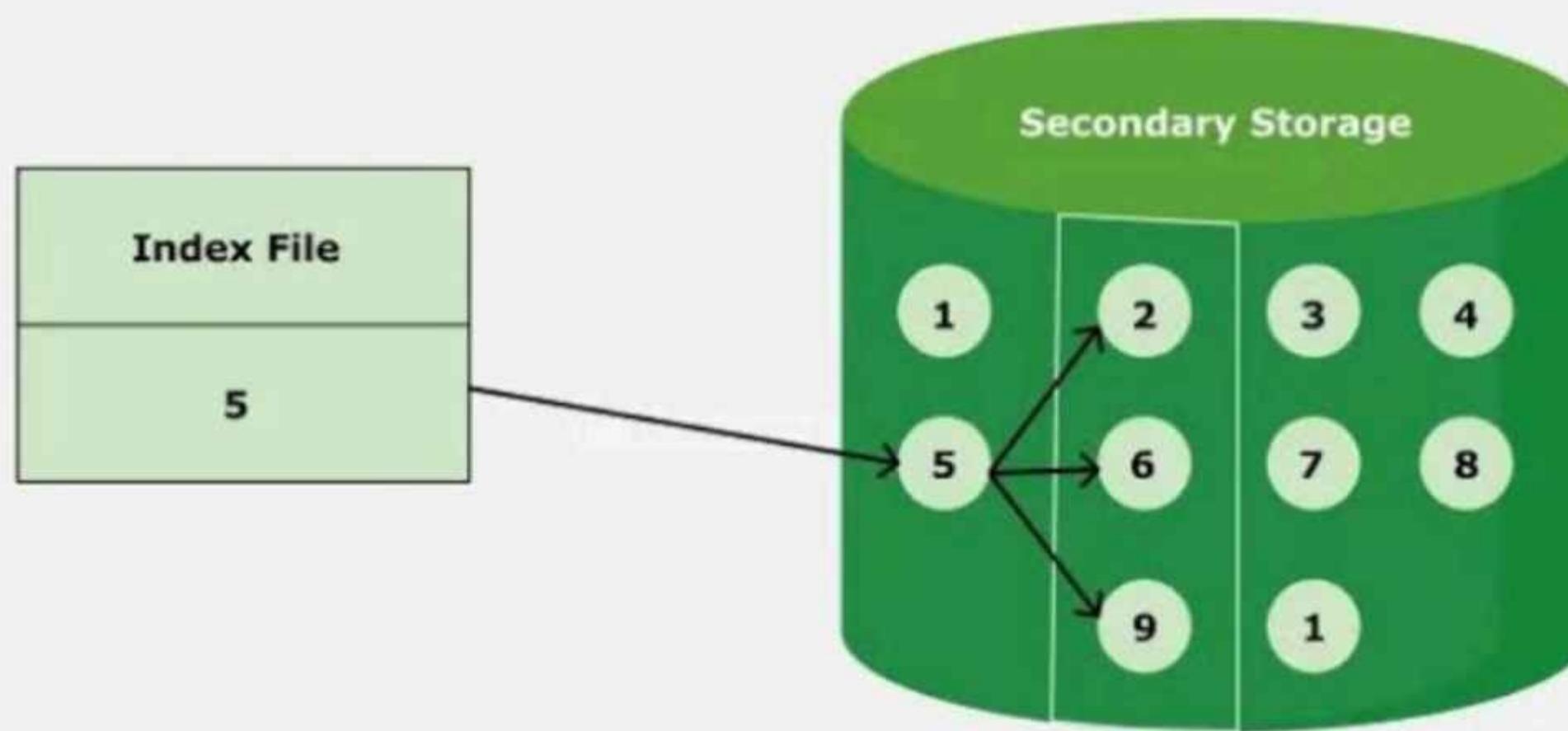
### 3. Indexed Access (सूचकांकित पहुँच विधि):

- इस विधि में एक Index Table बनाई जाती है, जिसमें यह जानकारी होती है कि किस डेटा का पता (location) कहाँ है। इससे हम सीधे डेटा तक पहुँच सकते हैं।
- In this method an Index Table is created which contains the information about the location of which data. With this we can access the data directly.





## Index Access Method





## Advantages (फायदे):

- Search करने में तेज़
- Fast to search
- Direct और Sequential दोनों Access को Support करता है
- Supports both direct and sequential access

## Disadvantages (कमियाँ):

- Index को अलग से Maintain करना पड़ता है
- Index has to be maintained separately
- Index फाइल भी Storage लेती है
- Index file also takes up storage
- Index Update करने की ज़रूरत होती है
- Index needs to be updated

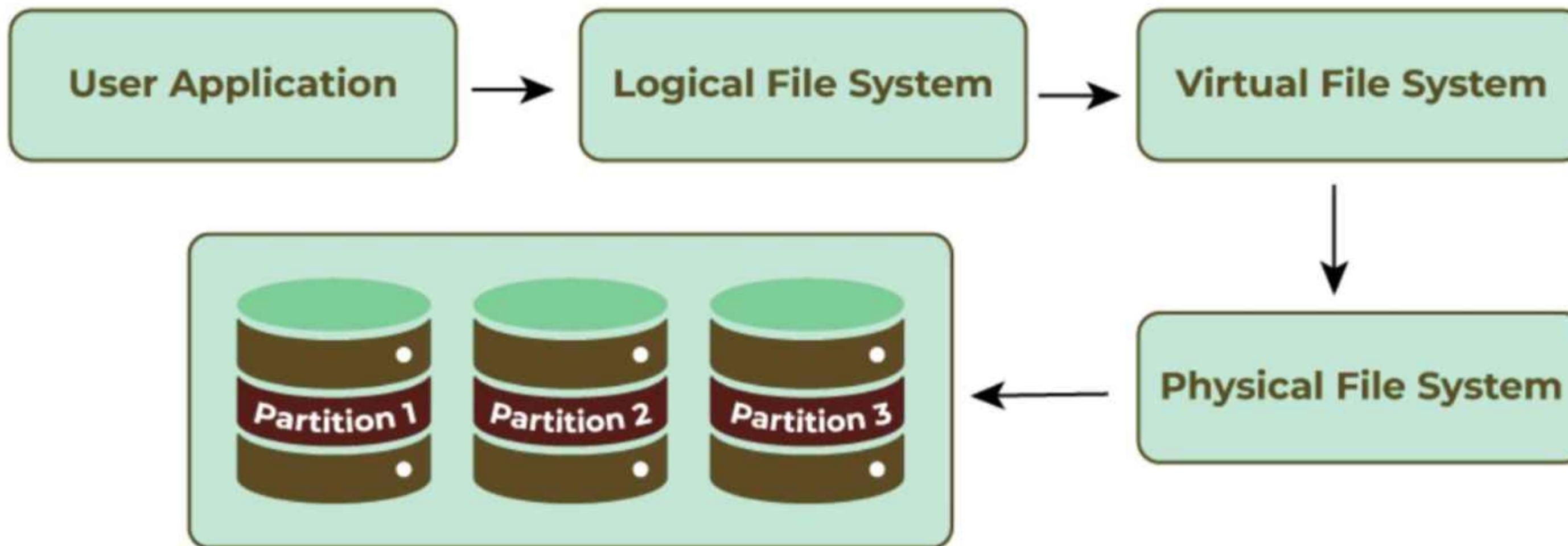


## फाइल सिस्टम (File System):

- फाइल सिस्टम किसी भी ऑपरेटिंग सिस्टम (Operating System) का एक महत्वपूर्ण हिस्सा है, जो स्टोरेज डिवाइस जैसे हार्ड ड्राइव, SSD और USB ड्राइव पर डेटा को संगठित, संग्रहित और प्रबंधित करने का संरचित तरीका प्रदान करता है।
- A file system is an important part of any operating system, providing a structured way to organize, store, and manage data on storage devices such as hard drives, SSDs, and USB drives.



## The Architecture of a File System





### 1. User Application (उपयोगकर्ता एप्लिकेशन):

- वे प्रोग्राम या सॉफ्टवेयर जो फाइल ऑपरेशन जैसे पढ़ना (read), लिखना (write), या हटाना (delete) अनुरोध करते हैं।
- Programs or software that request file operations such as read, write, or delete.

### 2. Logical File System (लॉजिकल फाइल सिस्टम):

- यह मेटाडेटा, फाइल के नाम, डायरेक्टरी और एक्सेस परमिशन को प्रबंधित करता है।
- It manages metadata, file names, directories, and access permissions.

### 3. Virtual File System (VFS) (वर्चुअल फाइल सिस्टम):

- यह एक ब्रिज की तरह कार्य करता है, जिससे विभिन्न फाइल सिस्टम्स को एक ही इंटरफेस के तहत काम करने की अनुमति मिलती है।
- It acts as a bridge, allowing different file systems to work under a single interface.

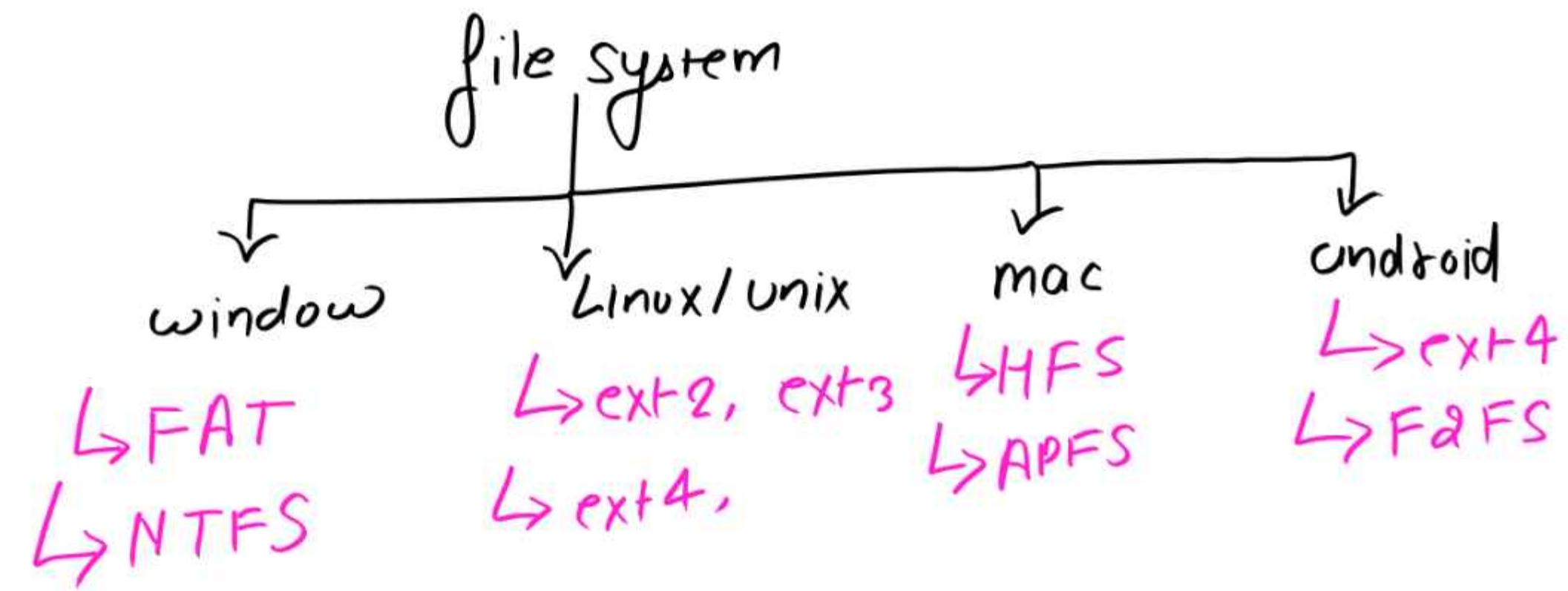


#### 4. Physical File System (फिजिकल फाइल सिस्टम):

- यह डिस्क पर डेटा ब्लॉक्स का वास्तविक भंडारण (storage) संभालता है।
- It handles the actual storage of data blocks on disk.

#### 5. Partition 1, Partition 2, Partition 3 (पार्टिशन 1, पार्टिशन 2, पार्टिशन 3):

- स्टोरेज डिवाइस के विभाजन, जहाँ फाइलें भौतिक रूप से संग्रहीत (physically stored) होती हैं।
- Partitions of a storage device where files are physically stored.





## Popular File Systems (लोकप्रिय फाइल सिस्टम्स):

### 1. FAT (File Allocation Table) - फाइल एलोकेशन टेबल:

- FAT एक पुराना फाइल सिस्टम है जिसका उपयोग शुरुआती Windows और कुछ अन्य ऑपरेटिंग सिस्टम में किया जाता था।
- FAT is an older file system used in early Windows and some other operating systems.
- यह फाइलों को ट्रैक करने के लिए एक टेबल (File Allocation Table) का उपयोग करता है, जो यह बताती है कि फाइल के कौन से हिस्से डिस्क पर कहाँ संग्रहीत हैं।
- It uses a table (File Allocation Table) to track files, which tells which parts of a file are stored where on the disk.



## मुख्य बिंदु (Key Points):

- सरल और हल्का फाइल सिस्टम।
- Simple and lightweight file system.
- छोटे स्टोरेज डिवाइस जैसे पेन ड्राइव, मेमोरी कार्ड आदि में उपयोगी।
- Useful for small storage devices like pen drives, memory cards, etc.
- आधुनिक सुरक्षा और परमिशन फीचर्स की कमी होती है।
- Lacks modern security and permissions features.



## 2. NTFS (New Technology File System) - न्यू टेक्नोलॉजी फाइल सिस्टम:

- NTFS एक आधुनिक फाइल सिस्टम है जिसे Microsoft ने Windows OS के लिए विकसित किया। यह उन्नत सुरक्षा, डेटा रिकवरी और बड़ी फाइलों के लिए बेहतर सपोर्ट प्रदान करता है।
- NTFS is a modern file system developed by Microsoft for the Windows operating system. It offers advanced security, data recovery, and better support for large files.

### मुख्य बिंदु (Key Points):

- फाइल और फोल्डर परमिशन को सपोर्ट करता है (Access Control).
- Supports file and folder permissions (Access Control).
- डेटा कम्प्रेशन और एन्क्रिप्शन (Encryption) की सुविधा देता है।
- Provides data compression and encryption.
- बड़ी फाइलों और हार्ड ड्राइव के लिए उपयुक्त।
- Suitable for large files and hard drives.

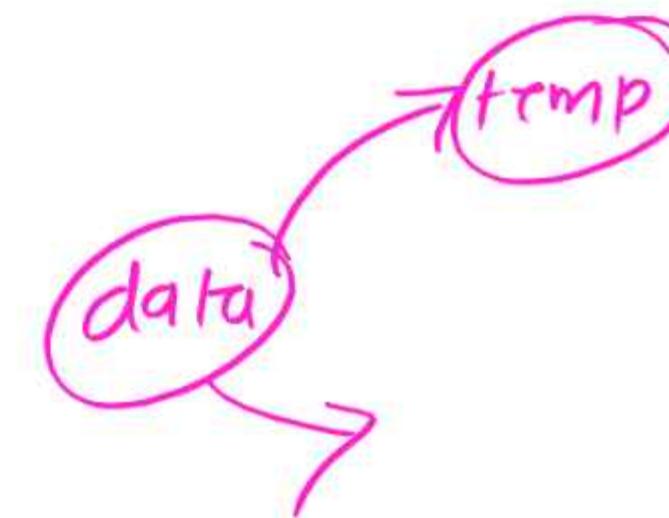


### 3. ext (Extended File System) - एक्सटेंडेड फाइल सिस्टम:

- ext फाइल सिस्टम का उपयोग Linux और Unix आधारित ऑपरेटिंग सिस्टम में किया जाता है। इसके कई संस्करण हैं - ext2, ext3, ext4, जिनमें से ext4 सबसे आधुनिक और तेज़ है।
- The ext file system is used in Linux and Unix-based operating systems. There are several versions of it - ext2, ext3, and ext4, of which ext4 is the most modern and fastest.

#### मुख्य बिंदु (Key Points):

- Linux OS का डिफॉल्ट फाइल सिस्टम।
- The default file system of the Linux OS.
- जर्नलिंग (Journaling) फीचर से डेटा लॉस कम होता है।
- Journaling feature reduces data loss.
- बड़ी फाइलों और डिस्क पार्टिशन को सपोर्ट करता है।
- Supports large files and disk partitions.



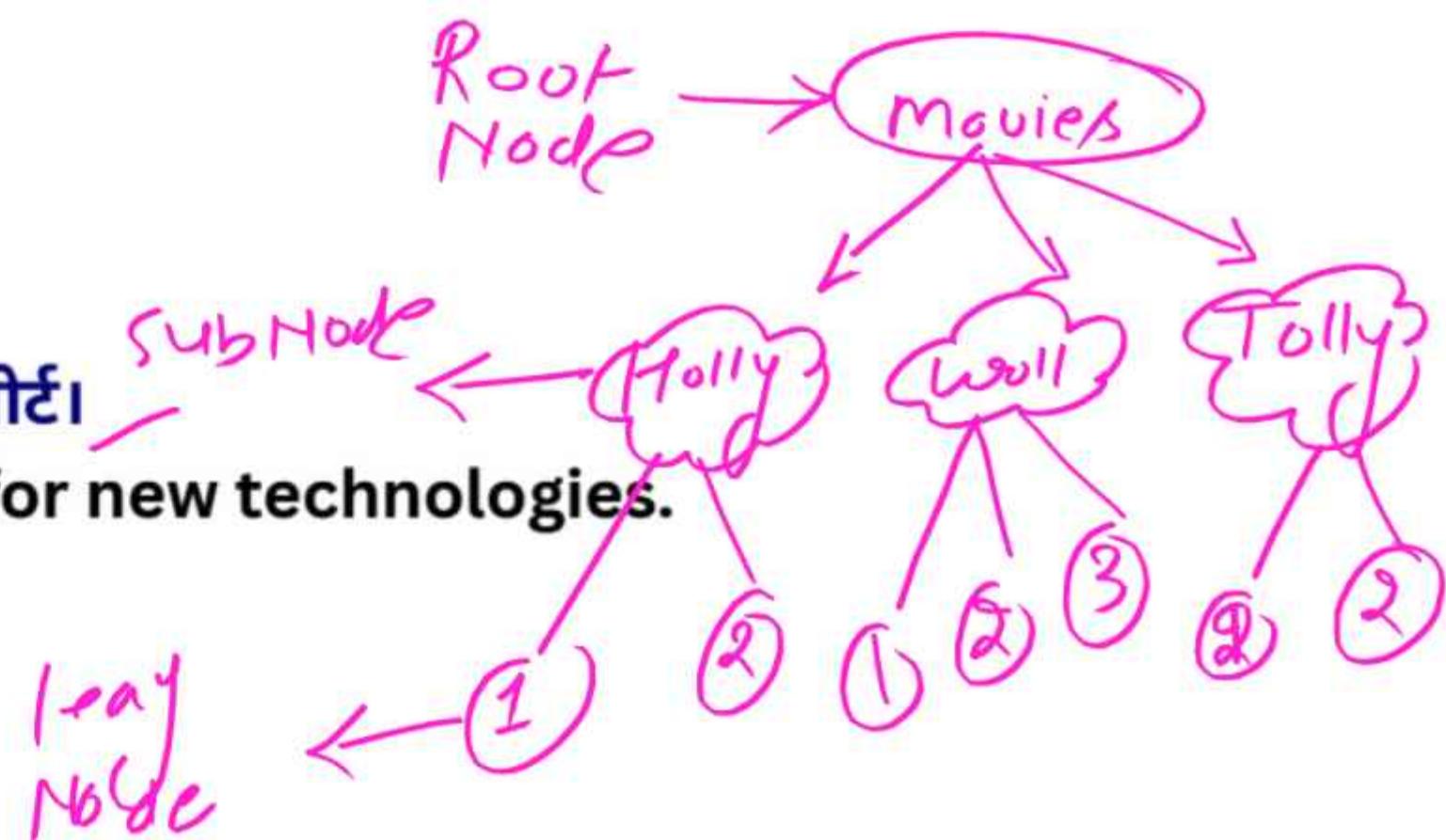


#### 4. HFS (Hierarchical File System) - हाइरार्किकल फाइल सिस्टम:

- HFS एक फाइल सिस्टम है जिसे Apple ने पुराने macOS (Macintosh OS) के लिए बनाया था। यह डेटा को एक हायरार्किकल स्ट्रक्चर (फोल्डर के अंदर फोल्डर) में संग्रहीत करता है।
- HFS is a file system developed by Apple for the older macOS (Macintosh OS). It stores data in a hierarchical structure (folders within folders).

#### मुख्य बिंदु (Key Points):

- macOS के शुरुआती संस्करणों में उपयोग होता था।
- Used in early versions of macOS.
- सरल संरचना लेकिन नई तकनीकों के लिए सीमित सपोर्ट।
- Simple architecture but limited support for new technologies.
- अब इसे APFS ने बदल दिया है।
- Now replaced by APFS.





## 5. APFS (Apple File System) – एप्पल फाइल सिस्टम;

- APFS Apple द्वारा विकसित एक आधुनिक और सुरक्षित फाइल सिस्टम है, जिसका उपयोग macOS, iOS, iPadOS, और watchOS में किया जाता है। यह speed, security और efficiency के लिए डिज़ाइन किया गया है।
- APFS is a modern and secure file system developed by Apple and used in macOS, iOS, iPadOS, and watchOS. It's designed for speed, security, and efficiency.

### मुख्य बिंदु (Key Points):

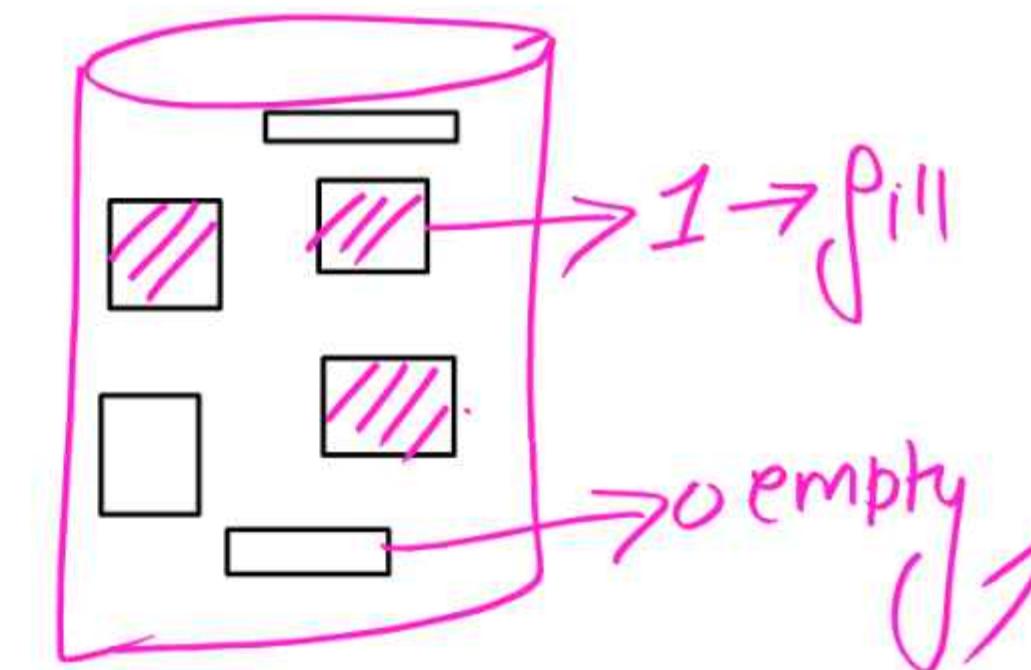
- डेटा एन्क्रिप्शन और स्नैपशॉट फीचर सपोर्ट करता है।
- Supports data encryption and snapshot features.
- SSD (Solid State Drives) के लिए optimized।
- Optimized for SSDs (Solid State Drives).
- macOS High Sierra (2017) से डिफॉल्ट फाइल सिस्टम।
- Default file system since macOS High Sierra (2017).



~~not~~ imp

## Free Space Management (मुक्त स्थान प्रबंधन):

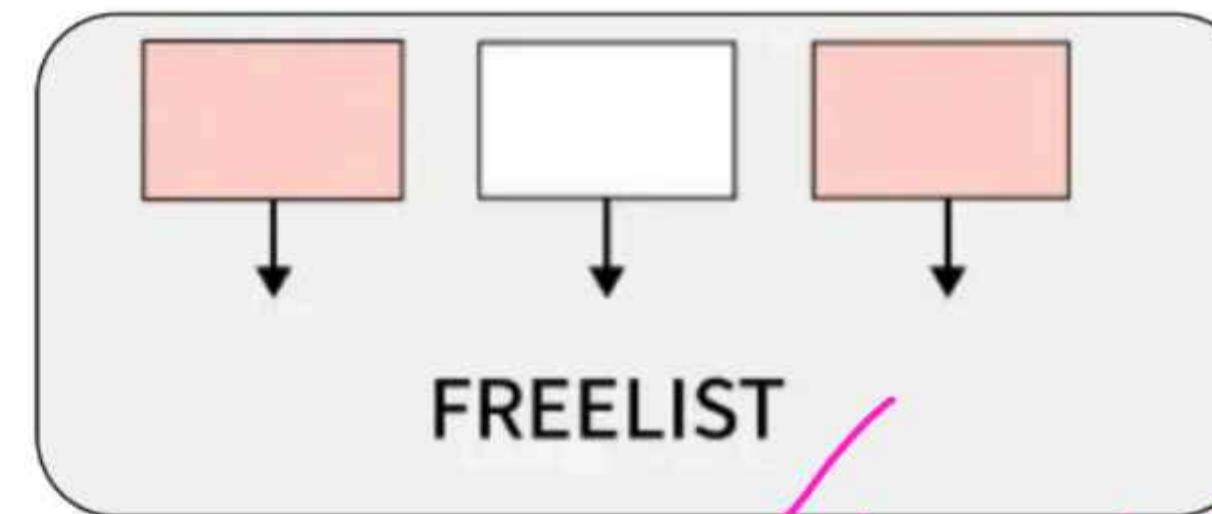
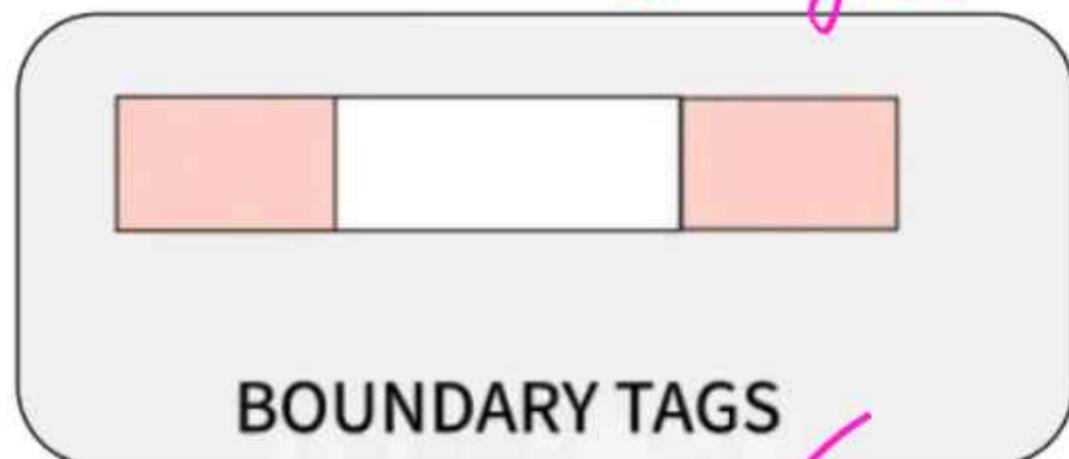
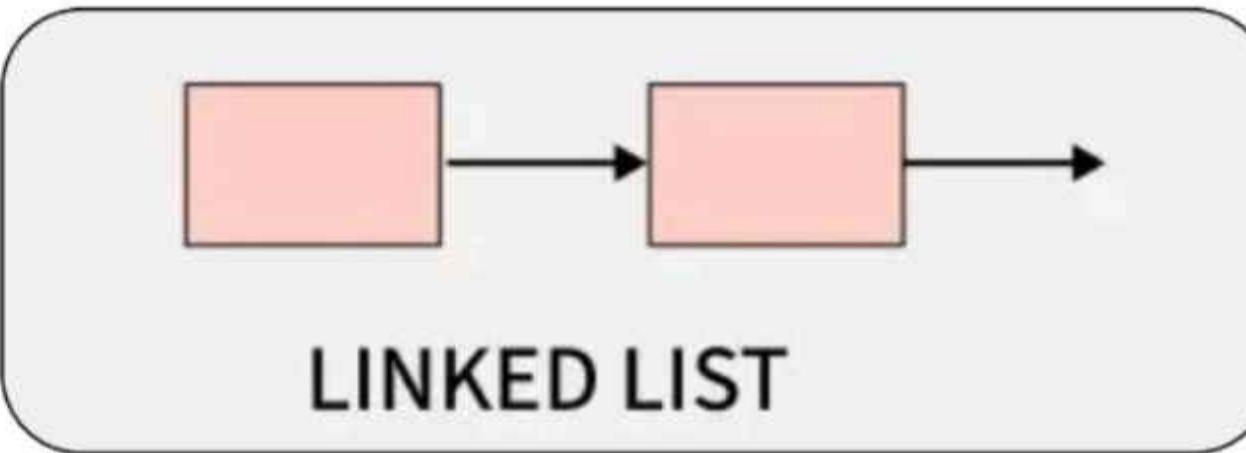
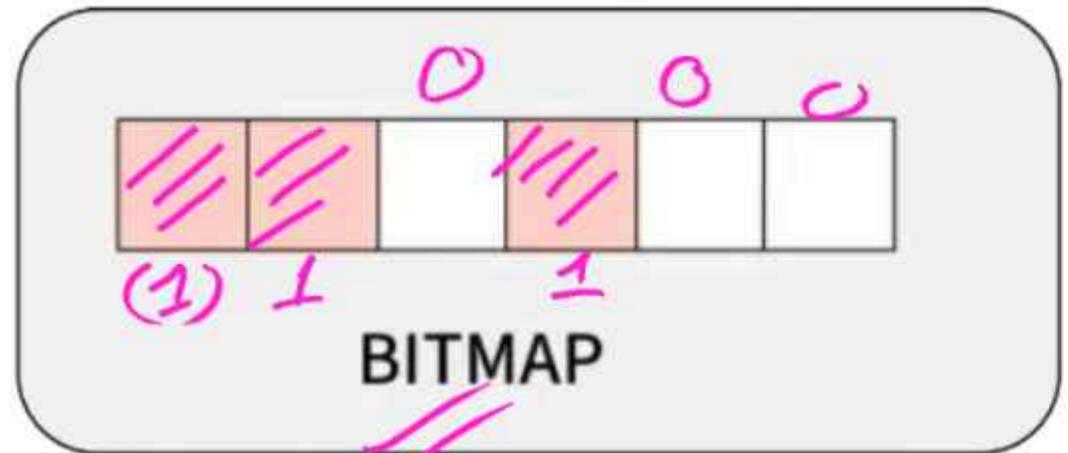
- Free Space Management वह प्रक्रिया है जिसके द्वारा हार्ड डिस्क या अन्य सेकेंडरी स्टोरेज डिवाइस पर उपलब्ध (unallocated) खाली स्थान का प्रबंधन किया जाता है।
- Free Space Management is the process of managing the available (unallocated) free space on a hard disk or other secondary storage device.
- इसका मुख्य उद्देश्य है – डिस्क के खाली ब्लॉक्स को ट्रैक करना और जब भी नई फ़ाइल बनती है, तो उसी खाली स्थान का पुनः उपयोग करना।
- Its main purpose is to track free blocks on the disk and reuse that free space whenever a new file is created.





## Free Space Management (मुक्त स्थान प्रबंधन):

① Tail → [ ]



array, linked-list



## 1. Bit Map या Bit Vector (बिट मैप या बिट वेक्टर)

- इस विधि (approach) में Bitmap या Bit Vector का उपयोग किया जाता है, जो bits की एक श्रृंखला (series) होती है। प्रत्येक bit डिस्क के एक ब्लॉक (disk block) का प्रतिनिधित्व करती है।
- This approach uses a bitmap or bit vector, which is a series of bits. Each bit represents a disk block.
- प्रत्येक bit दो मान (values) ले सकती है:
- Each bit can take two values:
  - 0 → यह दर्शाता है कि ब्लॉक खाली (Free) है।
  - 0 → This indicates that the block is free.
  - 1 → यह दर्शाता है कि ब्लॉक उपयोग में (Allocated) है।
  - 1 → This indicates that the block is allocated.



### लाभ (Advantages):

- खाली ब्लॉक्स को जल्दी से खोजा जा सकता है।
- Empty blocks can be found quickly.
- यह विधि सरल और आसान होती है।
- This method is simple and easy.

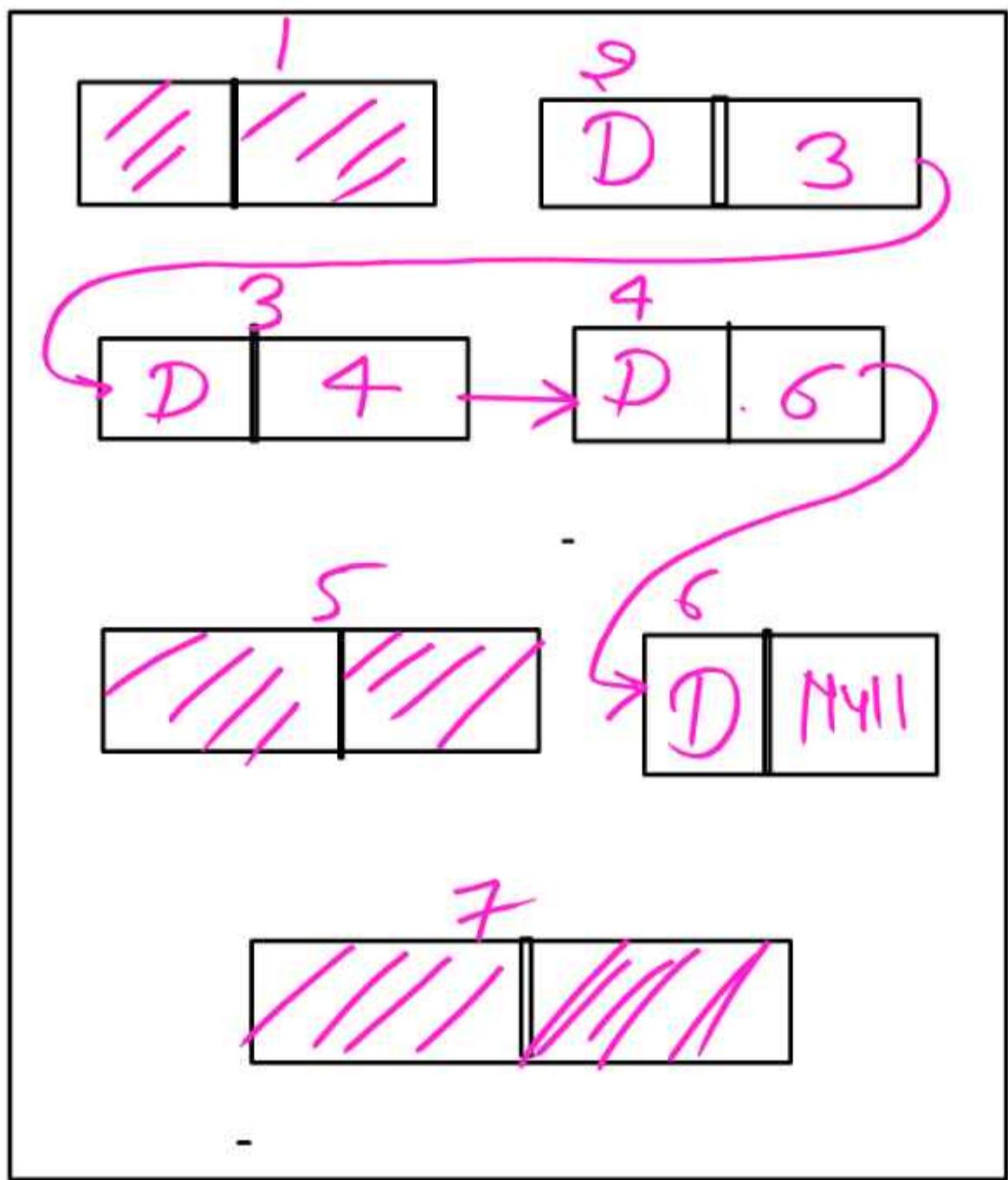
### हानियाँ (Disadvantages):

- बड़े डिस्क के लिए अधिक मेमोरी की आवश्यकता होती है।
- Larger disks require more memory.
- bitmap को बार-बार अपडेट करना पड़ता है।
- The bitmap needs to be updated frequently.



## 2. Linked List Approach (लिंक्ड लिस्ट विधि):

- इस विधि में सभी खाली ब्लॉक्स (Free Blocks) को एक लिंक्ड लिस्ट (Linked List) के रूप में जोड़ा जाता है। प्रत्येक खाली ब्लॉक में अगले खाली ब्लॉक (Next Free Block) का पता (Address) संग्रहित किया जाता है।
- In this method, all free blocks are added to a linked list. Each free block stores the address of the next free block.
- जब कोई ब्लॉक Allocate (उपयोग) किया जाता है या Free (खाली) किया जाता है, तो यह लिस्ट डायनामिक रूप से (Dynamically) अपडेट होती रहती है।
- This list is updated dynamically when a block is allocated (used) or freed (emptied).





### लाभ (Advantages):

- खाली ब्लॉक्स की जानकारी स्टोर करने के लिए अतिरिक्त मेमोरी बहुत कम लगती है।
- Storing information about empty blocks requires very little additional memory.
- लिस्ट को आसानी से अपडेट किया जा सकता है।
- The list can be easily updated.

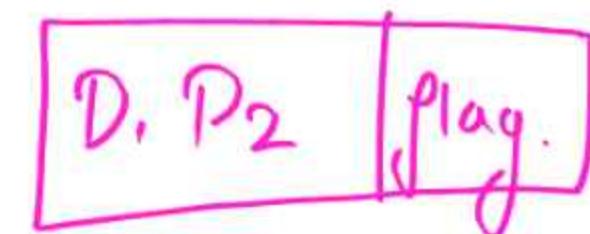
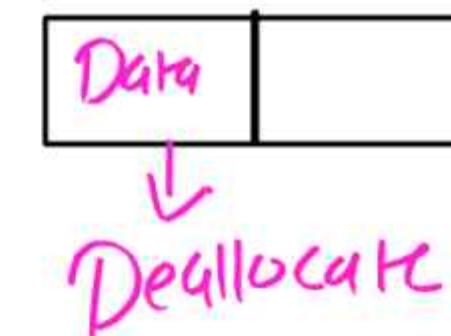
### हानियाँ (Disadvantages):

- किसी विशेष ब्लॉक तक पहुँचने के लिए लिस्ट को क्रमवार (Sequentially) खोजना पड़ता है।
- To access a particular block, the list has to be searched sequentially.
- यह विधि रैंडम एक्सेस (Random Access) को सपोर्ट नहीं करती।
- This method does not support random access.



### 3. Boundary Tag Method (बाउंड्री टैग विधि):

- इस विधि में प्रत्येक ब्लॉक के साथ एक “boundary tag” (सीमा टैग) जुड़ा होता है, जो उस ब्लॉक का आकार (size) और यह बताता है कि ब्लॉक free (खाली) है या allocated (उपयोग में)।
- In this method, each block is associated with a "boundary tag", which indicates the size of the block and whether the block is free or allocated.
- जब किसी ब्लॉक को मुक्त (deallocate) किया जाता है, तो पास-पास के खाली ब्लॉकों को आपस में जोड़ (merge) दिया जाता है ताकि fragmentation (खंडन) कम किया जा सके। यह विधि memory management systems में बहुत उपयोगी होती है।
- When a block is deallocated, adjacent free blocks are merged to reduce fragmentation. This method is very useful in memory management systems.





## Advantages (फायदे):

### 1. Fragmentation कम करता है:

- पास-पास के खाली ब्लॉक को जोड़कर मेमोरी का बेहतर उपयोग करता है।
- Makes better use of memory by combining adjacent empty blocks.

### 2. Coalescing आसान बनाता है:

- आस-पास के free blocks को पहचानना और जोड़ना सरल होता है।
- It is easy to identify and connect nearby free blocks.

### 3. Efficient Memory Utilization:

- मेमोरी को डायनामिक रूप से मैनेज करना आसान हो जाता है।
- It becomes easier to manage memory dynamically.



### Disadvantages (नुकसान):

#### 1. थोड़ा अतिरिक्त ओवरहेड (Slight additional overhead):

- हर ब्लॉक में टैग जानकारी रखने से थोड़ी अतिरिक्त मेमोरी लगती है। ✓
- Keeping tag information in each block takes up a little extra memory.

#### 1. जटिल संरचना (Complex structure):

- यह विधि bitmap या linked list की तुलना में अधिक complex होती है। ✓
- This method is more complex than a bitmap or linked list. ✓

#### 1. Implementation कठिन (Implementation difficult):

- कोडिंग और प्रबंधन के लिए अतिरिक्त लॉजिक की आवश्यकता होती है। ✓
- Requires additional logic for coding and management. ✓



#### 4. Free List (फ्री लिस्ट विधि):

- इस विधि में सभी खाली ब्लॉक्स (Free Blocks) की एक सूची (List) बनाई जाती है। यह सूची या तो array के रूप में होती है या linked list के रूप में। फ्री लिस्ट की प्रत्येक एंट्री (entry) सीधे डिस्क पर उपलब्ध खाली ब्लॉक के पते (address) को दर्शाती है। इससे सिस्टम को यह पता चलता है कि अगला खाली ब्लॉक कहाँ है और नई फाइल कहाँ स्टोर की जा सकती है।
- This method creates a list of all free blocks. This list can be either an array or a linked list. Each entry in the free list directly represents the address of an available free block on the disk. This lets the system know where the next free block is and where a new file can be stored.



## फायदे(Advantages):

- तेज़ अलोकेशन क्योंकि खाली ब्लॉक्स की जानकारी पहले से मौजूद होती है।
- Faster allocation because information about empty blocks is already present
- पढ़ना और प्रबंधित करना आसान।
- Easier to read and manage.

## नुकसान (Disadvantages):

- सूची को स्टोर करने के लिए अतिरिक्त मेमोरी की आवश्यकता।
- Additional memory required to store the list
- समय के साथ खंडन (Fragmentation) की समस्या हो सकती है
- Fragmentation may occur over time.



## → Efficiency और Performance (कुशलता और प्रदर्शन):

### Efficiency (कुशलता):

- किसी सिस्टम, प्रोग्राम या प्रक्रिया की कुशलता यह बताती है कि वह उपलब्ध संसाधनों (resources) का कितना अच्छा उपयोग करता है।
- The efficiency of a system, program, or process describes how well it uses available resources.
  - यह CPU, मेमोरी, डिस्क और नेटवर्क जैसे संसाधनों की बचत और सही उपयोग को दर्शाती है।
  - It reflects the efficiency and efficient use of resources such as CPU, memory, disk, and network.
  - कुशल सिस्टम कम संसाधन खर्च करता है और अधिक कार्य कर सकता है।
  - An efficient system consumes fewer resources and can perform more tasks.



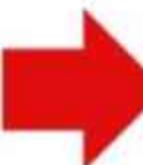
### Performance (प्रदर्शन):

- प्रदर्शन यह बताता है कि कोई सिस्टम, प्रोग्राम या प्रक्रिया कितनी जल्दी और सही तरीके से कार्य पूरा करता है।
- Performance describes how quickly and accurately a system, program, or process completes a task.
  - इसमें Execution Time, Response Time, Throughput और Latency जैसे मानदंड शामिल होते हैं।
  - This includes parameters such as execution time, response time, throughput, and latency.
  - अच्छा प्रदर्शन मतलब सिस्टम तेज़, reliable और users के लिए संतोषजनक है।
  - Good performance means the system is fast, reliable, and satisfactory to users.



### Directory System :

- Directory System एक ऐसी फाइल होती है जिसमें दूसरी फाइलों और डायरेक्टरी (फोल्डर) की जानकारी रखी जाती है। यह फाइलों को व्यवस्थित (Organize) करने और आसानी से खोजने की सुविधा देता है।
- Directory System is a file in which information about other files and directories (folders) is kept. It allows to organize the files and search them easily.
- Directory मतलब Folder – जैसे आपके मोबाइल में “Documents”, “Images” या “Downloads” नाम के फोल्डर होते हैं, जिनमें फाइलें रखी जाती हैं। इन्हें व्यवस्थित करने और जल्दी एक्सेस करने के लिए Directory System का उपयोग किया जाता है।
- Directory means Folder - Like in your mobile there are folders named "Documents", "Images" or "Downloads" in which files are kept. Directory System is used to organize them and access them quickly.



## Directory Operations:

### 1. Create (डायरेक्टरी बनाना)

- नई डायरेक्टरी बनाना ताकि उसमें फाइलें या अन्य डायरेक्टरी रखी जा सकें।
- Create a new directory to place files or other directories into.

#### Command:

- Windows में: Right Click → New → Folder
- Linux में: mkdir folder\_name

### 2. Delete (डायरेक्टरी हटाना)

- किसी पुरानी या खाली डायरेक्टरी को सिस्टम से हटाना।
- Removing any old or empty directories from the system.

#### Command:

- Windows में: Right click → Delete
- Linux में: rmdir folder\_name



### 3. Open (डायरेक्टरी खोलना):

- किसी डायरेक्टरी को खोलकर उसमें रखी फाइलों/फोल्डरों को देखना या उन पर कार्य करना।
- Opening a directory and viewing or working on the files/folders contained in it.

#### Command:

- Double-click करके “My Documents” खोलना।

### 4. Close (डायरेक्टरी बंद करना)

- खुली हुई डायरेक्टरी को बंद करना, जिससे वो RAM से हट जाए।
- Closing the open directory to remove it from RAM.

### 5. Rename (नाम बदलना)

- किसी डायरेक्टरी या फाइल का नाम बदलना।
- Renaming a directory or file.
- Windows में: Right click → Rename or Linux में: mv oldname newname

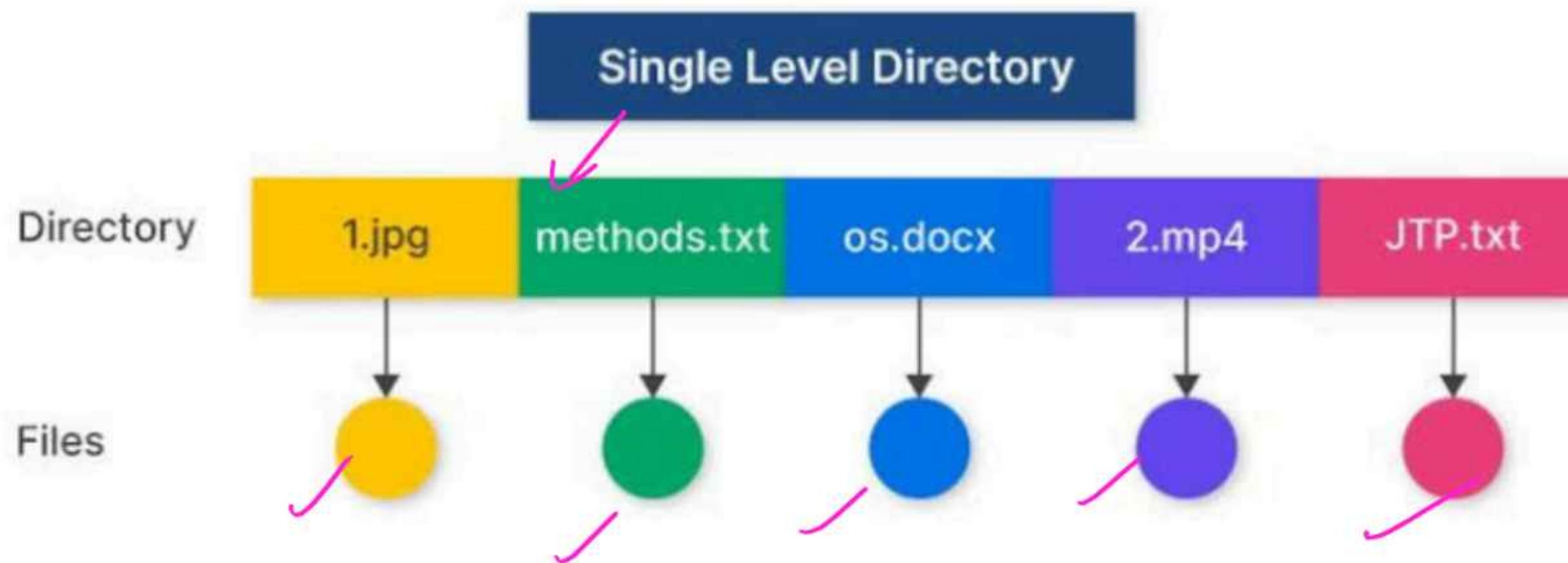
Ankit.c → Addition.c



## → Types of Directory Structure:

### 1. Single-Level Directory (एक-स्तरीय डायरेक्टरी):

- इस संरचना में सभी फाइलें एक ही डायरेक्टरी (Root Directory) में होती हैं, जिसमें सभी यूज़र की फाइलें रखी जाती हैं।
- In this structure all the files are located in a single directory (root directory), in which all the user's files are kept.





## Advantage (फायदे):

### 1. सरल और आसान (Simple and easy):

- एक ही डायरेक्टरी में सब कुछ होता है, जिससे समझना आसान होता है।
- Everything is in one directory, which makes it easy to understand.

### 2. फाइल एक्सेस तेज़ (Fast file access):

- फाइल को जल्दी ढूँढ सकते हैं क्योंकि सब एक ही जगह है।
- You can find files quickly because everything is in one place.



## Disadvantage (नुकसान):

### 1. नाम टकराव (Name collision):

- दो फाइलों का नाम एक जैसा नहीं हो सकता।
- Two files cannot have the same name.

### 2. बड़ा सिस्टम अव्यवस्थित (Large system unorganized):

- अगर बहुत सारी फाइलें हों तो सिस्टम गड़बड़ हो जाता है।
- If there are too many files, the system becomes chaotic.

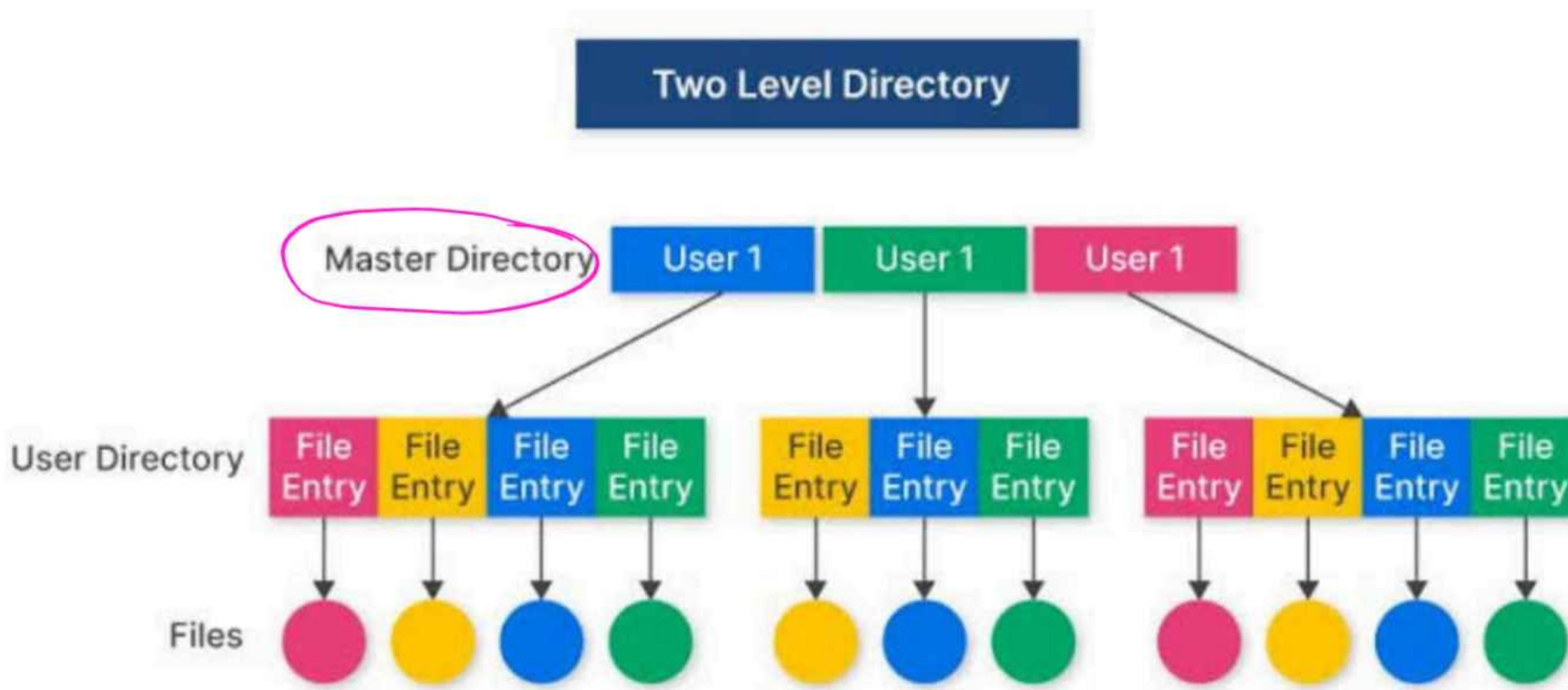
### 3. कम सुरक्षा (Low security)

- कोई भी यूज़र किसी की भी फाइल देख सकता है।
- Any user can view anyone's files.



## 2. Two-Level Directory (दो-स्तरीय डायरेक्टरी):

- इस संरचना में हर यूजर के लिए अलग-अलग डायरेक्टरी होती है, और हर यूजर की फाइलें उसकी व्यक्तिगत डायरेक्टरी में रखी जाती हैं।
- In this structure there is a separate directory for each user, and each user's files are kept in his or her personal directory.





## Advantage (फायदे):

### 1. हर यूज़र की फाइलें अलग (Separate files for each user):

- फाइलें यूज़र के नाम से अलग डायरेक्टरी में होती हैं।
- Everything is in one directory, which makes it easy to understand.

### 2. नाम टकराव नहीं(No name collisions):

- एक ही नाम की फाइल अलग यूज़र के पास हो सकती है।
- You can find files quickly because everything is in one place.

### 3. थोड़ी सुरक्षा (Some security):

- यूज़र्स की फाइलें प्राइवेट रह सकती हैं।
- Users' files can remain private.



## Disadvantage (नुकसान):

### 1. शेयर करना कठिन (Difficult to share):

- एक यूजर दूसरे यूजर की फाइल सीधे एक्सेस नहीं कर सकता।
- one user cannot directly access another user's files.

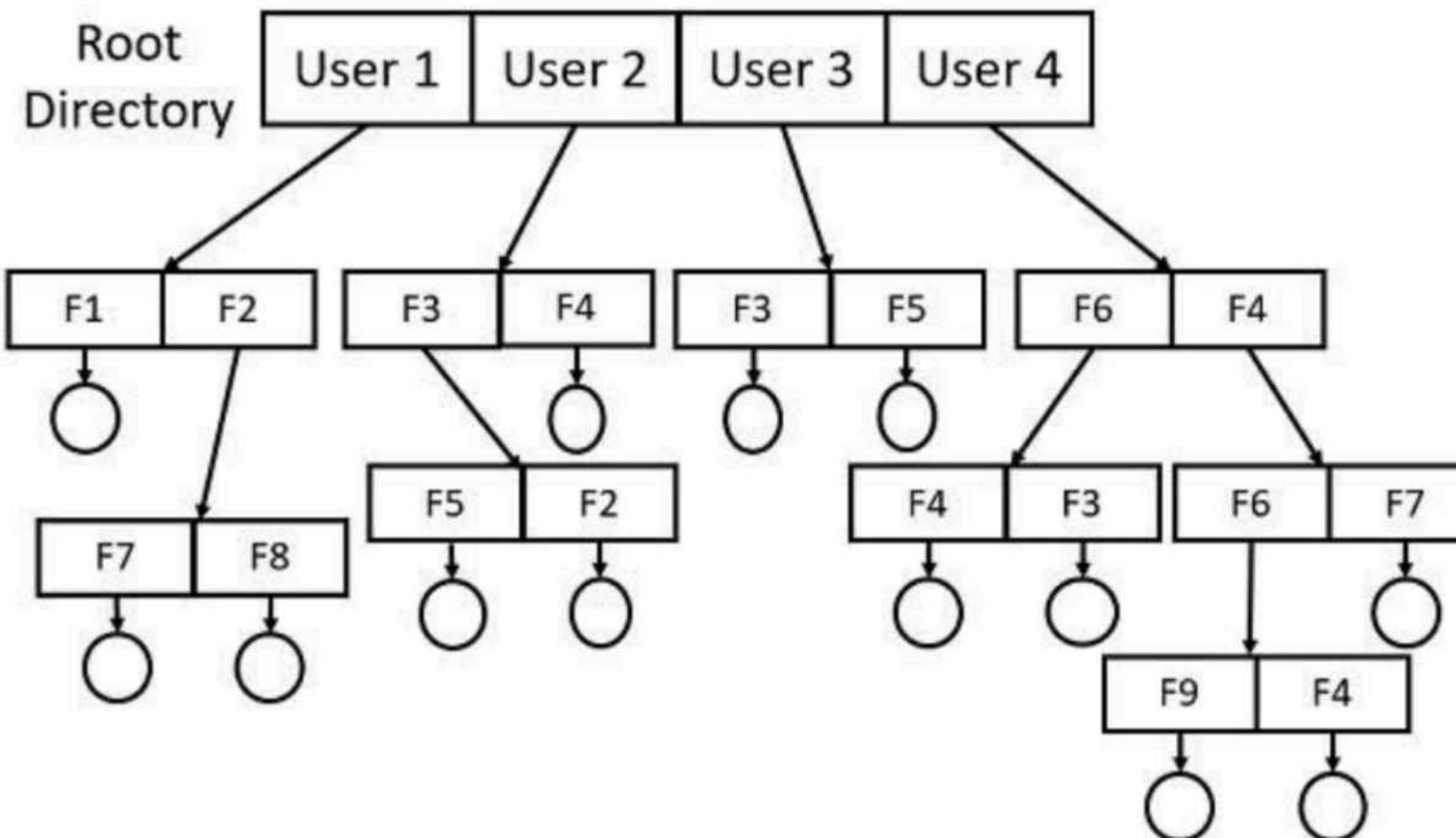
### 2. थोड़ी सीमित संरचना (Slightly limited structure):

- सब डायरेक्टरी नहीं बनाई जा सकती।
- subdirectories cannot be created.



### 3. Tree Directory / Hierarchical Directory (वृक्ष संरचना)

- यह सबसे सामान्य और उपयोगी संरचना है जिसमें एक Root Directory होती है और उसके नीचे कई Sub-directories और फाइलें होती हैं। यह पेड़ जैसी दिखती है।
- This is the most common and useful structure which consists of a root directory and many sub-directories and files beneath it. It looks like a tree.





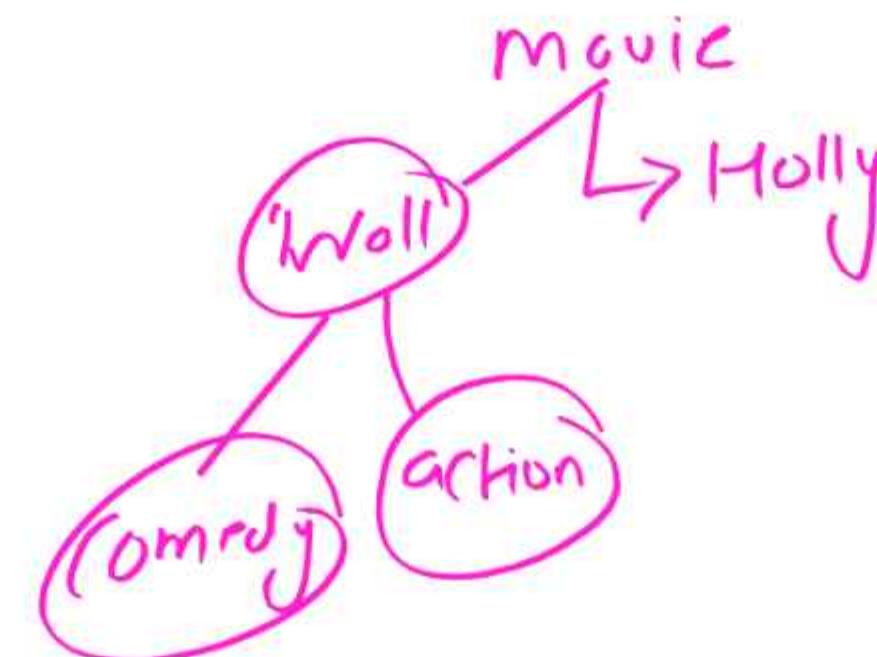
## Advantage (फायदे):

### 1. फाइलों का अच्छा संगठन (Good file organization):

- फोल्डर और सब-फोल्डर बनाकर सही से फाइलें रख सकते हैं।
- You can organize files properly by creating folders and subfolders.

### 2. नेविगेशन आसान (Easy navigation):

- हर फाइल का एक स्पष्ट रास्ता (path) होता है।
- Every file has a clear path.





## Disadvantage (नुकसान):

### 1. थोड़ा जटिल (A little complicated):

- बहुत सारी सब-डायरेक्टरी बनने पर ढूँढना कठिन हो सकता है।
- May be hard to find if there are a lot of sub-directories created.

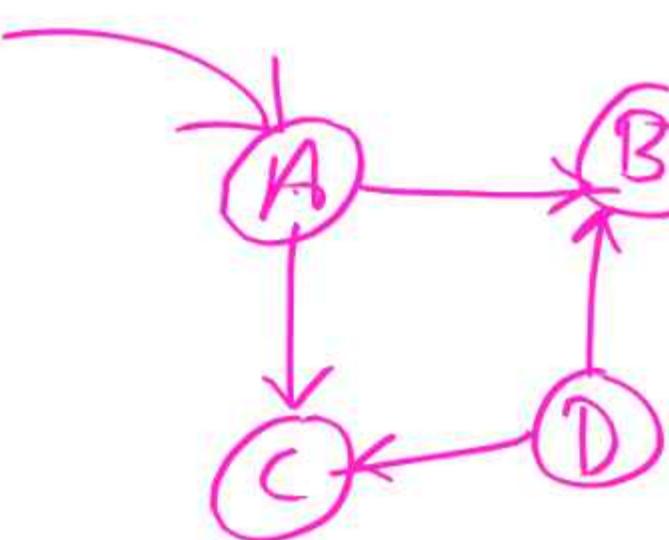
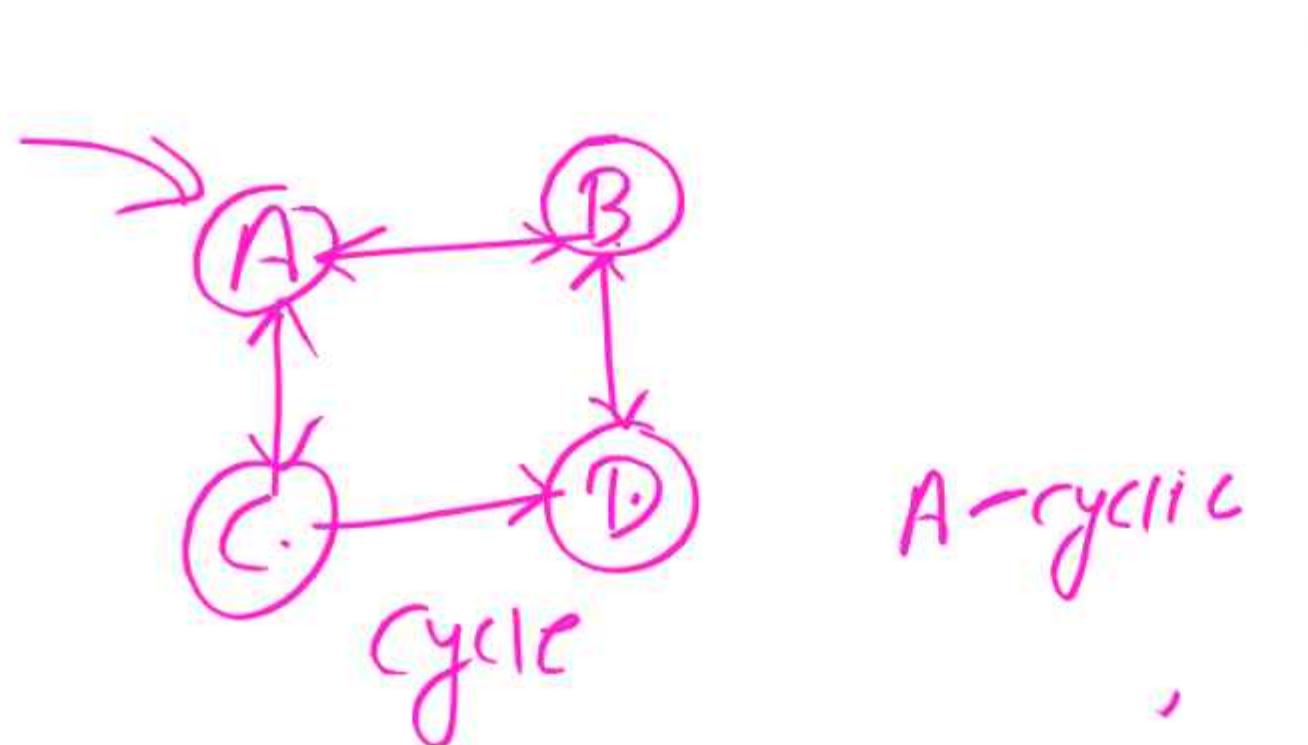
### 2. ज्यादा ज्ञान की ज़रूरत (More knowledge required):

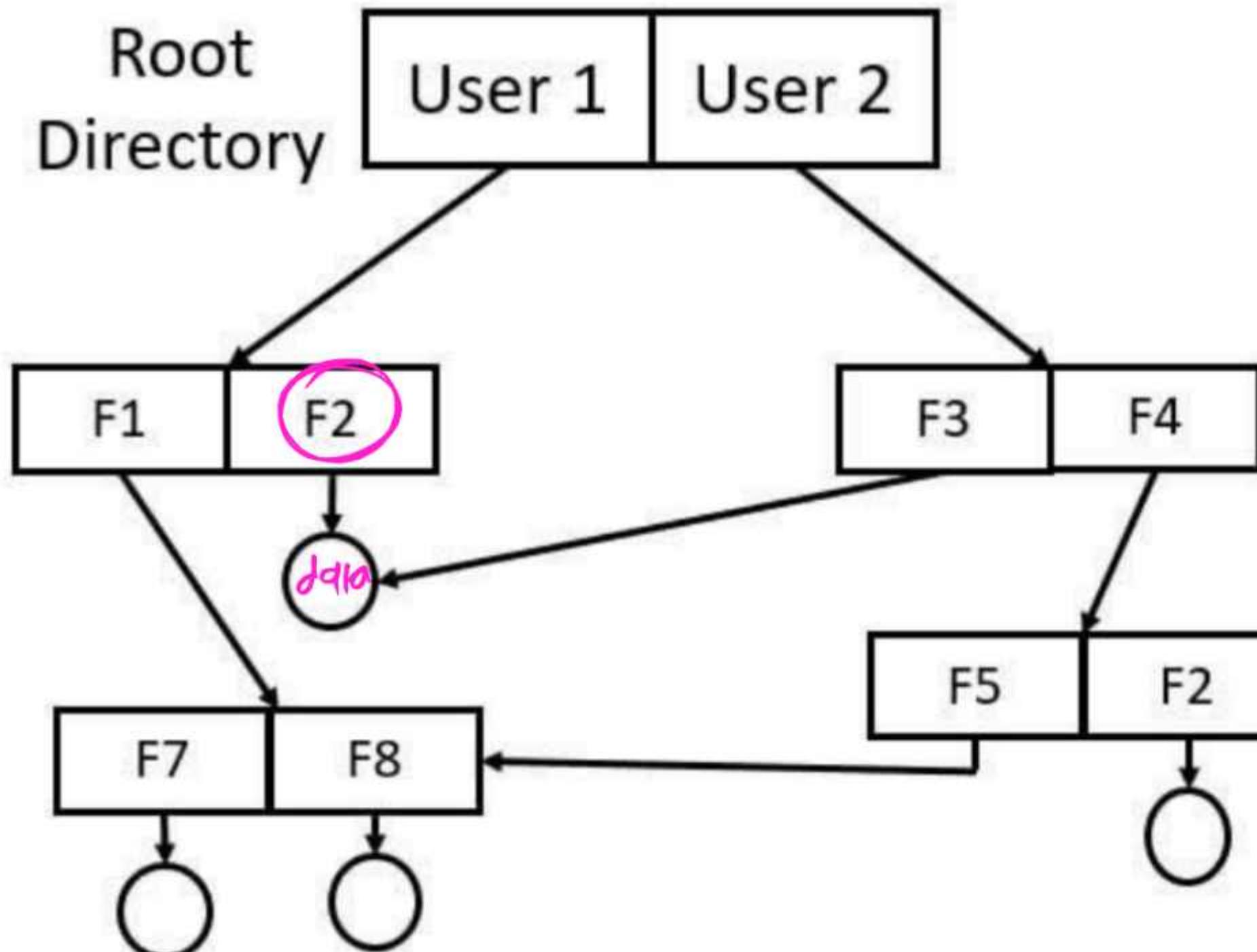
- नए यूज़र्स के लिए समझना थोड़ा कठिन हो सकता है।
- May be a little difficult for new users to understand.



#### 4. Acyclic Graph Directory (गैर-चक्रीय ग्राफ डायरेक्टरी):

- इस संरचना में एक ही फाइल या डायरेक्टरी को कई डायरेक्टरी द्वारा साझा (shared) किया जा सकता है। लेकिन यह लूप (Loop) नहीं बनाता, इसलिए इसे “Acyclic” कहते हैं।
- In this structure a single file or directory can be shared by multiple directories. But it does not create a loop, so it is called "Acyclic".







## Advantage (फायदे):

### 1. फाइल शेयरिंग आसान (Easy file sharing):

- एक ही फाइल को कई डायरेक्टरी से एक्सेस किया जा सकता है।
- The same file can be accessed from multiple directories.

### 2. स्पेस की बचत (More knowledge required):

- एक ही कॉपी को बार-बार सेव नहीं करना पड़ता।
- No need to save the same copy again and again.

### 3. साझेदारी के लिए अच्छा (Good for sharing):

- teamwork में उपयोगी होता है।
- Useful in teamwork.



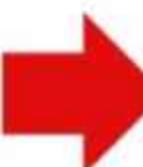
## Disadvantage (नुकसान):

### 1. लिंक टूटने की समस्या (Problem of broken links):

- अगर लिंक गलत हो गया तो फाइल खो सकती है।
- If the link goes wrong, the file can be lost.

### 2. कंप्यूटर को समझाना कठिन (Difficult to explain to the computer):

- कौन-सी असली फाइल है और कौन लिंक ये समझाना मुश्किल हो सकता है।
- It can be difficult to tell which is the original file and which is the link.



## File System Mounting:

- Mounting का मतलब है किसी file system (जैसे hard disk partition, pen drive, CD/DVD, network drive) को operating system के directory structure से जोड़ना ताकि user और applications उस file system की files को access कर सकें।
- Mounting means attaching a file system (such as a hard disk partition, pen drive, CD/DVD, network drive) to the directory structure of the operating system so that users and applications can access the files in that file system.
- जब भी कोई नया storage device connect होता है, उसे use करने से पहले mount करना ज़रूरी होता है।
- Whenever a new storage device is connected, it must be mounted before use.



## How Mount Generally Works:

### 1. Identifying the Filesystem (फाइल सिस्टम पहचानना):

- Mount करने से पहले आपको ये पहचानना होता है कि कौन-सा filesystem mount करना है। ये हो सकता है:
- Before mounting, you need to identify which filesystem you want to mount.

This could be:

- Hard drive का local partition
- USB drive
- Network share
- या फिर virtual filesystem (जैसे cloud storage services में होता है)।



## 2. Choosing a Mount Point (Mount Point चुनना):

- आपको ये decide करना होता है कि system की directory structure में कहाँ पर उस filesystem की files को accessible बनाना है। इसे mount point कहते हैं।
- You need to decide where in the system's directory structure you want to make the files of that filesystem accessible. This is called a mount point.

## 3. Mounting the Filesystem (फाइल सिस्टम को mount करना):

- जब आपने filesystem identify कर लिया और mount point चुन लिया, तो आप एक command या system call का use करके filesystem को उस जगह mount करते हो।
- Once you have identified the filesystem and selected a mount point, you use a command or system call to mount the filesystem at that location.
- किस command से mount करना है, ये आपके operating system पर depend करता है।
- Which command to use to mount depends on your operating system
  - Example: Linux में mount command का use किया जाता है
  - The mount command is used in Linux.



#### 4. Accessing the Filesystem (फ़ाइल सिस्टम को access करना):

- Filesystem mount होने के बाद, आप उसकी files को mount point के through access कर सकते हो।
- Once the filesystem is mounted, you can access its files through the mount point.

#### 5. Unmounting (Unmount करना):

- जब filesystem का use खत्म हो जाता है, तो आप उसे unmount कर सकते हो। इससे वो filesystem directory tree से detach हो जाएगा और उसकी files inaccessible हो जाएँगी।
- When a filesystem is no longer in use, you can unmount it. This will detach it from the filesystem directory tree and make its files inaccessible.



## Mounts के प्रकार:

- Mounts को उनकी विशेषताओं (characteristics) और कार्यक्षमताओं (functionalities) के आधार पर विभिन्न प्रकारों में बँटा जा सकता है। मुख्यतः mounts के दो प्रकार होते हैं:
- Mounts can be divided into different types based on their characteristics and functionalities. There are two main types of mounts:

### 1. Physical Mounts (भौतिक Mounts):

- इसमें storage devices को भौतिक रूप से जोड़ना शामिल है, जैसे: Hard drives, USB flash drives.
- This involves physically connecting storage devices, such as: hard drives, USB flash drives.
- जब कोई device system से connect किया जाता है, तो operating system (कार्य प्रणाली) उस नए जुड़े हुए device को पहचान लेता है और उसे किसी विशेष mount point पर attach कर देता है।
- When a device is connected to the system, the operating system recognizes the newly connected device and attaches it to a specific mount point.



## 2. Virtual Mounts :

- भौतिक mounts के विपरीत, virtual mounts में किसी physical hardware की आवश्यकता नहीं होती। इसमें दूरस्थ (remote) file systems को network connection के माध्यम से mount किया जाता है।
- Unlike physical mounts, virtual mounts do not require any physical hardware.  
Remote file systems are mounted via a network connection.
- Virtual mounts के लिए सामान्यतः उपयोग किए जाने वाले protocols हैं:
- Commonly used protocols for virtual mounts are
  - NFS (Network File System)
  - SMB/CIFS (Server Message Block / Common Internet File System)
  - SSHFS (SSH File System)



## Advantages (फायदे):

- एक ही जगह से multiple storage devices को access किया जा सकता है।
- Multiple storage devices can be accessed from a single location.
- Local और Remote दोनों file systems को use करना आसान हो जाता है।
- Both local and remote file systems become easier to use.
- Files और directories को आसानी से organize और manage किया जा सकता है।
- Files and directories can be easily organized and managed.



### Disadvantages (नुकसान):

- अगर device को बिना unmount किए remove कर दिया जाए तो data corrupt हो सकता है।
- Data may be corrupted if the device is removed without unmounting it.
- Network-based mounts (जैसे NFS/SMB) slow हो सकते हैं और security risk भी होते हैं।
- Network-based mounts (such as NFS/SMB) can be slow and also pose a security risk.



## → Definition of file sharing:

- फाइल शेयरिंग वह प्रक्रिया है जिसमें दो या अधिक कंप्यूटर्स या यूज़र्स के बीच इलेक्ट्रॉनिक फाइलें जैसे डॉक्यूमेंट, म्यूज़िक, वीडियो, इमेज, या सॉफ्टवेयर साझा (share) की जाती हैं।
- File sharing is the process of sharing electronic files such as documents, music, videos, images, or software between two or more computers or users.





## Importance of File Sharing (फाइल शेयरिंग का महत्व):

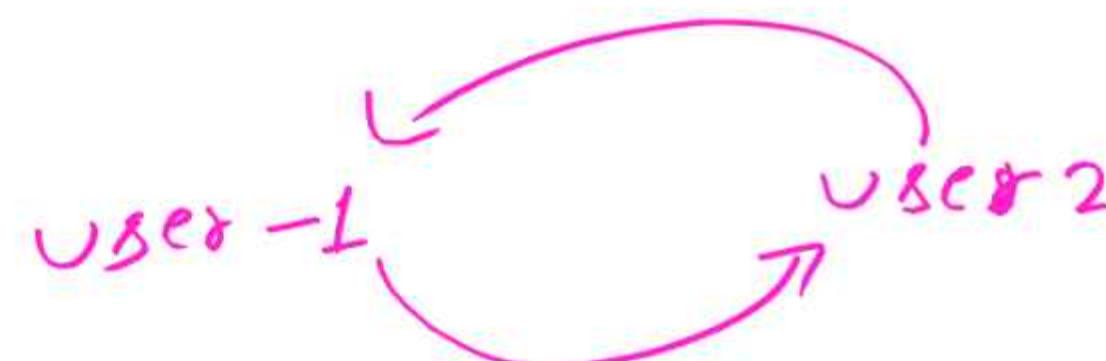
- सहयोग और संचार आसान बनाता है
- Makes collaboration and communication easier
- फाइलें जल्दी साझा की जा सकती हैं
- Files can be shared quickly
- भौतिक बैठकों की आवश्यकता घटाता है
- Reduces the need for physical meetings
- रिमोट वर्क को समर्थन देता है
- Supports remote work
- समय, और पैसे की बचत करता है
- Saves time & money.



## फाइल शेयरिंग के प्रकार (Types of File Sharing):

### 1. Peer-to-Peer (P2P) File Sharing (पीयर-टू-पीयर फाइल शेयरिंग):

- P2P फाइल शेयरिंग में फाइलें सीधे एक यूज़र से दूसरे यूज़र तक भेजी जाती हैं, इसमें किसी सेंट्रल सर्वर की आवश्यकता नहीं होती। हर यूज़र (peer) नेटवर्क का एक हिस्सा बन जाता है और वह फाइल भेज भी सकता है और प्राप्त भी कर सकता है।
- In P2P file sharing, files are sent directly from one user to another, without the need for a central server. Each user (peer) becomes part of the network and can send and receive files.





## कैसे काम करता है (How it works):

- जब कोई यूज़र कोई फाइल शेयर करता है, तो वह फाइल छोटे हिस्सों (chunks) में बाँट दी जाती है।
- When a user shares a file, the file is divided into smaller chunks.
- जो यूज़र उस फाइल को डाउनलोड करना चाहता है, वह उन हिस्सों को नेटवर्क में कई अन्य यूज़र्स से एक साथ डाउनलोड करता है। इससे डाउनलोड स्पीड बढ़ जाती है।
- A user who wants to download the file downloads those chunks simultaneously from multiple other users on the network. This increases download speeds.

## उदाहरण:

- BitTorrent, uTorrent, LimeWire आदि P2P सॉफ्टवेयर हैं।



### फायदे (Advantages):

- तेज़ डाउनलोड स्पीड (क्योंकि फाइल कई जगह से एक साथ आती है)।
- Faster download speeds (because the file comes from multiple locations simultaneously).
- सर्वर की जरूरत नहीं होती, इसलिए लागत कम होती है।
- No server is required, so costs are lower ✓
- नेटवर्क उपयोगकर्ताओं के बीच डेटा शेयरिंग आसान होती है।
- Data sharing between network users is easier. ✓



## नुकसान (Disadvantages):

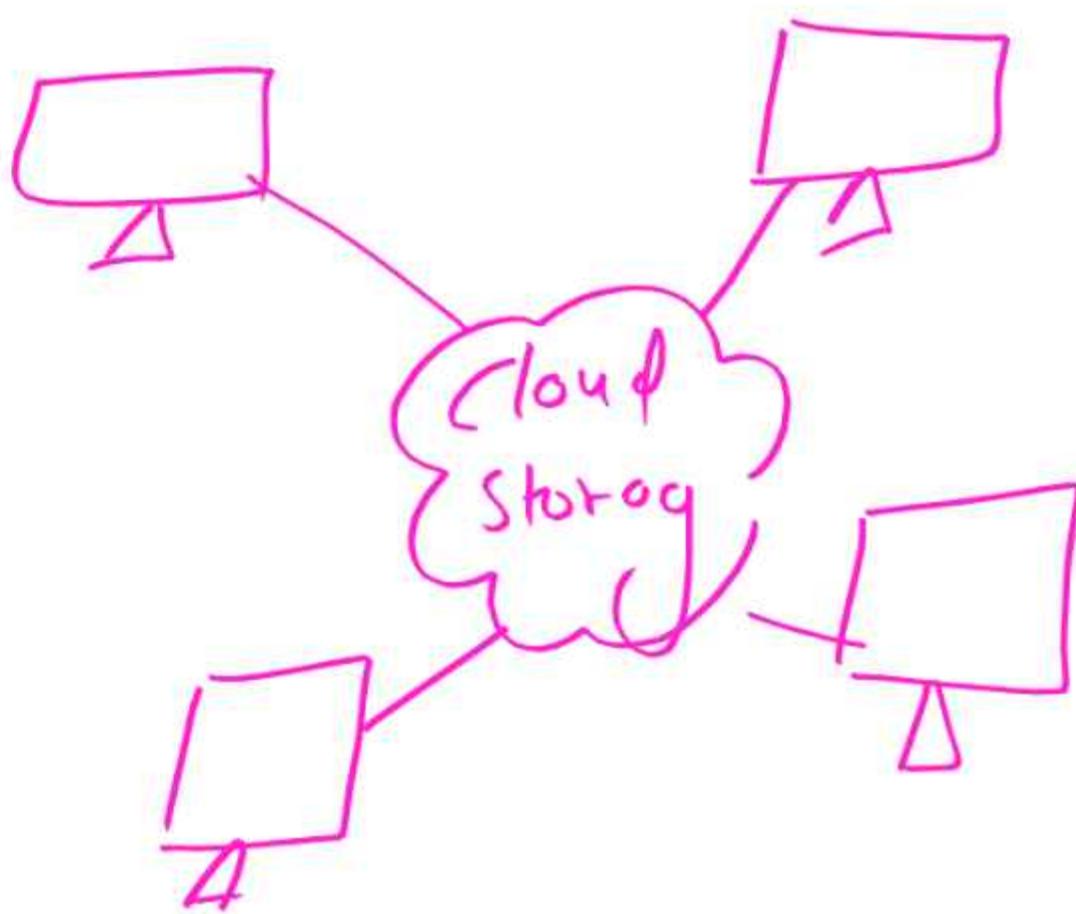
- सुरक्षा का खतरा (क्योंकि फाइलें अनजान यूज़र्स से आती हैं)।
- Security risks (because files come from unknown users).
- वायरस या मैलवेयर फैलने का जोखिम।
- Risk of spreading viruses or malware.
- कॉपीराइटेड फाइलें शेयर करने से कानूनी दिक्कतें हो सकती हैं।
- Sharing copyrighted files can lead to legal issues.



## 2. Cloud-Based File Sharing (क्लाउड-बेस्ड फाइल शेयरिंग):

- इस विधि में फाइलें क्लाउड सर्वर (Remote Server) पर स्टोर होती हैं और यूज़र्स उन्हें इंटरनेट के माध्यम से एक्सेस, डाउनलोड या अपलोड कर सकते हैं।
- In this method, files are stored on a cloud server (remote server) and users can access, download or upload them through the internet.

↳ 2UX7  
↳ any device  
↳ low cost  
↳ Remote location access.





### कैसे काम करता है (How it works.):

- यूज़र अपनी फाइल किसी क्लाउड सर्विस (जैसे Google Drive) पर अपलोड करता है।
- A user uploads their file to a cloud service (such as Google Drive).
- फिर वह फाइल का शेयर लिंक बनाकर दूसरों को भेज सकता है।
- They can then create a share link to the file and send it to others.
- सभी यूज़र्स इंटरनेट से जुड़कर उस फाइल को देख या डाउनलोड कर सकते हैं।
- All users connected to the Internet can view or download the file.

### उदाहरण:

- Google Drive, Dropbox, Microsoft OneDrive, iCloud



## फायदे (Advantages):

- कहीं से भी फाइल एक्सेस की जा सकती है।
- Files can be accessed from anywhere.
- कई लोग एक साथ एक ही फाइल पर काम कर सकते हैं (Collaboration)।
- Multiple people can work on the same file simultaneously (Collaboration).
- डेटा ऑटोमेटिकली बैकअप रहता है।
- Data is automatically backed up.
- ईमेल से बड़े फाइल भेजने की जरूरत नहीं होती।
- There's no need to send large files by email.



## नुकसान (Loss):

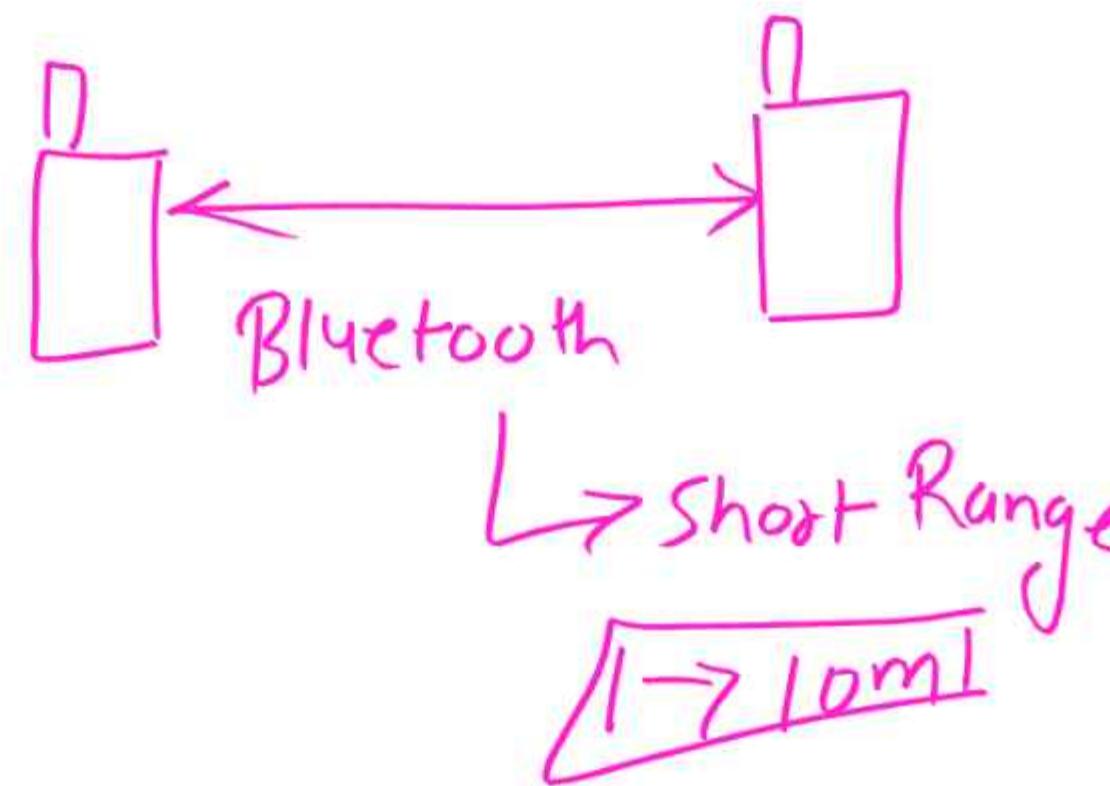
- इंटरनेट कनेक्शन जरूरी है।
- An internet connection is required.
- अगर पासवर्ड या लिंक गलत हाथों में चला जाए तो प्राइवेसी खतरे में।
- Privacy is at risk if the password or link falls into the wrong hands.
- फ्री अकाउंट्स में स्टोरेज लिमिट होती है।
- Free accounts have storage limits.

→ (15GB) → free  
↓ Paid



### 3. Direct File Transfer (डायरेक्ट फाइल ट्रांसफर):

- इस विधि में दो डिवाइसों के बीच सीधा कनेक्शन (Bluetooth, Wi-Fi Direct, या NFC) बनाकर फाइल ट्रांसफर की जाती है। यह विधि छोटी दूरी (Short Range) के लिए उपयुक्त है।
- This method transfers files by establishing a direct connection (Bluetooth, Wi-Fi Direct, or NFC) between two devices. This method is suitable for short distances.





### फायदे (Advantages.):

- इंटरनेट की जरूरत नहीं।
- तेज़ और आसान ट्रांसफर।
- व्यक्तिगत फाइल शेयरिंग के लिए अच्छा तरीका।
- No internet required.
- Fast and easy transfers.
- Great for personal file sharing.

### ~~djs~~ फायदे (Advantages.):

- केवल सीमित दूरी तक काम करता है।
- Only works over a limited distance.
- एक समय में सीमित डिवाइस कनेक्ट हो सकते हैं।
- A limited number of devices can be connected at a time.
- बड़ी फाइलों के लिए स्पीड कम हो सकती है।
- Speeds may be slower for larger files.



#### 4. Removable Media File Sharing (रिमूवेबल मीडिया फाइल शेयरिंग):

- इस विधि में भौतिक स्टोरेज डिवाइस जैसे ~~USB~~ Pen Drive, CD/DVD, या ~~External Hard~~ Disk का उपयोग किया जाता है। यूजर फाइल को इन डिवाइसों में कॉपी करता है और फिर दूसरे को देकर शेयर करता है।
- This method uses physical storage devices such as USB pen drives, CD/DVDs, or external hard disks. The user copies files to these devices and then shares them with others.

##### कैसे काम करता है (How it works.):

- फाइल को USB या हार्ड ड्राइव में कॉपी करें।
- Copy the file to a USB or hard drive.
- डिवाइस को दूसरे सिस्टम में लगाएं।
- Plug the device into another system.
- फाइल को ट्रांसफर करें।
- Transfer the file.



### फायदे:

- इंटरनेट की कोई आवश्यकता नहीं।
- बहुत तेज़ ट्रांसफर (USB 3.0 या SSD होने पर)।
- बड़ी फाइलें भी आसानी से ट्रांसफर की जा सकती हैं।

### Advantages:

- No internet connection required.
- Extremely fast transfers (with USB 3.0 or SSD).
- Even large files can be transferred easily.

### नुकसान (Disadvantages):

- डिवाइस खोने का खतरा।
- Danger of losing the device.
- वायरस फैलने की संभावना।
- Possibility of spreading viruses.
- एक समय में केवल एक व्यक्ति को फाइल दी जा सकती है।
- Only one person can be given a file at a time.



## File Sharing Protection Measures:

### 1. Encryption (एन्क्रिप्शन):

- एन्क्रिप्शन वह प्रक्रिया है जिसमें डेटा को एक कोड़ेड भाषा में बदल दिया जाता है, जिसे केवल अधिकृत उपयोगकर्ता (Authorized Users) अपने Decryption Key के जरिए ही एक्सेस कर सकते हैं।
- Encryption is the process in which data is converted into a coded language, which can only be accessed by authorized users through their decryption key.
- यह फाइलों को अनधिकृत पहुँच से बचाता है।
- This protects files from unauthorized access.
- अगर फाइल शेयरिंग के दौरान डेटा को इंटरसेप्ट किया भी जाए, तो भी वह गोपनीय (Confidential) रहता है।
- Even if data is intercepted during file sharing, it remains confidential.

Hello → Encryption → (@# \$0)



## 2. Password Protection (पासवर्ड सुरक्षा):

- पासवर्ड सुरक्षा में फाइल को पासवर्ड के जरिए लॉक किया जाता है।
- Password protection locks a file with a password.
- फाइल तक पहुँचने के लिए पासवर्ड डालना अनिवार्य होता है।
- A password is required to access the file.
- यह सुनिश्चित करता है कि केवल अधिकृत उपयोगकर्ता ही फाइल को देख या संशोधित (Modify) कर सकते हैं।
- This ensures that only authorized users can view or modify the file.

## 3. Secure File Transfer Protocols (सुरक्षित फाइल ट्रांसफर प्रोटोकॉल):

- सुरक्षित फाइल ट्रांसफर प्रोटोकॉल वे नियम और तकनीकें हैं जिनके माध्यम से इंटरनेट पर फाइलें सुरक्षित तरीके से भेजी और प्राप्त की जाती हैं। इनका मुख्य उद्देश्य फाइल ट्रांसफर के दौरान अनधिकृत पहुँच और डेटा चोरी को रोकना है।



- Secure file transfer protocols are rules and techniques for securely sending and receiving files over the Internet. Their primary purpose is to prevent unauthorized access and data theft during file transfers.

### प्रमुख प्रोटोकॉल (Common Secure Protocols):

#### (a).SFTP (Secure File Transfer Protocol):

- यह SSH (Secure Shell) के जरिए फाइल ट्रांसफर करता है।
- It transfers files via SSH (Secure Shell).
- फाइल को एन्क्रिप्ट करके भेजता है।
- It sends files encrypted.
- सुरक्षित और विश्वसनीय तरीका है।
- It's a secure and reliable method.

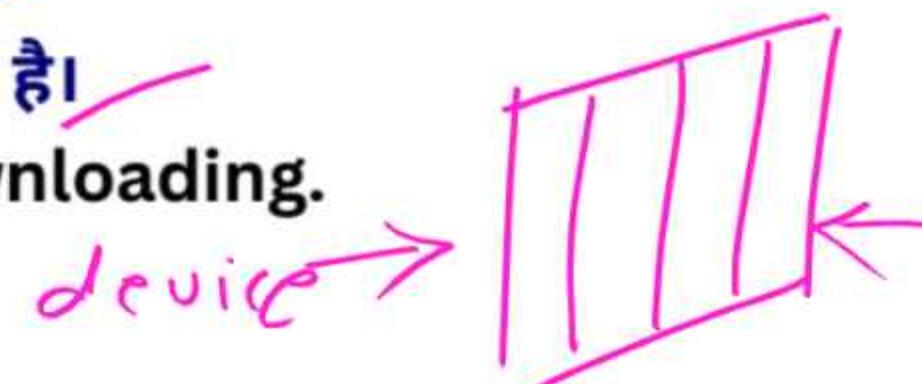


### (b). HTTPS (Hypertext Transfer Protocol Secure):

- वेब आधारित फाइल ट्रांसफर के लिए इस्तेमाल होता है।
- Used for web-based file transfers.
- यह डेटा को एन्क्रिप्ट करके ब्राउज़र और सर्वर के बीच भेजता है।
- It encrypts data between the browser and the server.
- ऑनलाइन फाइल शेयरिंग और डाउनलोडिंग के लिए सुरक्षित तरीका है।
- It is a secure method for online file sharing and downloading.

### 4. Firewall Protection (फायरवॉल सुरक्षा):

- फायरवॉल एक सुरक्षा तंत्र (security mechanism) है जो कंप्यूटर या नेटवर्क पर आने-जाने वाले डेटा ट्रैफिक की निगरानी, नियंत्रण और फ़िल्टरिंग करता है।
- A firewall is a security mechanism that monitors, controls, and filters data traffic flowing to and from a computer or network.





- यह सुनिश्चित करता है कि केवल अधिकृत यूज़र्स और सुरक्षित डेटा ही सिस्टम तक पहुँचे और अनधिकृत पहुँच, हैकिंग, वायरस या मालवेयर को रोका जा सके
- This ensures that only authorized users and secure data have access to the system and prevents unauthorized access, hacking, viruses, or malware.

Operating system → complete

→ Playstore

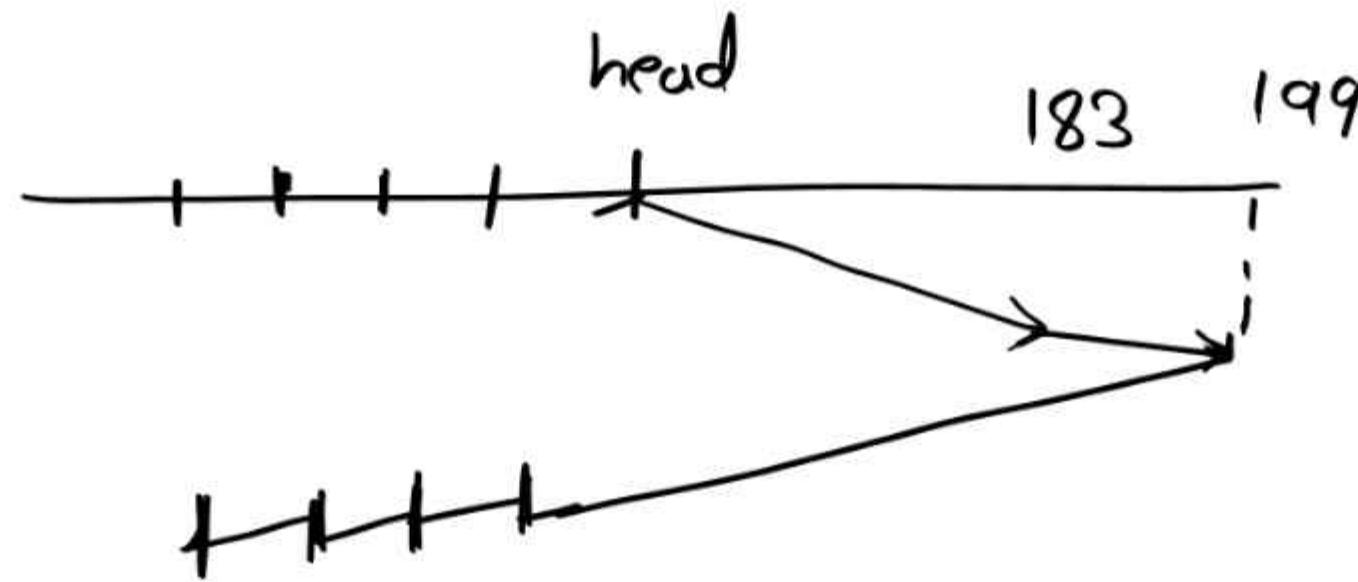
↳ Semeste add application

↳ Regi + login

↳ free (folder)

↳ {  
    1-Semester  
    3→semester  
    5→semester

SCAN - algo.



C- SCAN : =>

