ZEBACAR

MCA I, SEM I
A.Y. 2023-24
Subject:- Operating Systems Concepts
By
Prof. Ashok Deokar

Topic No 1.
Overview

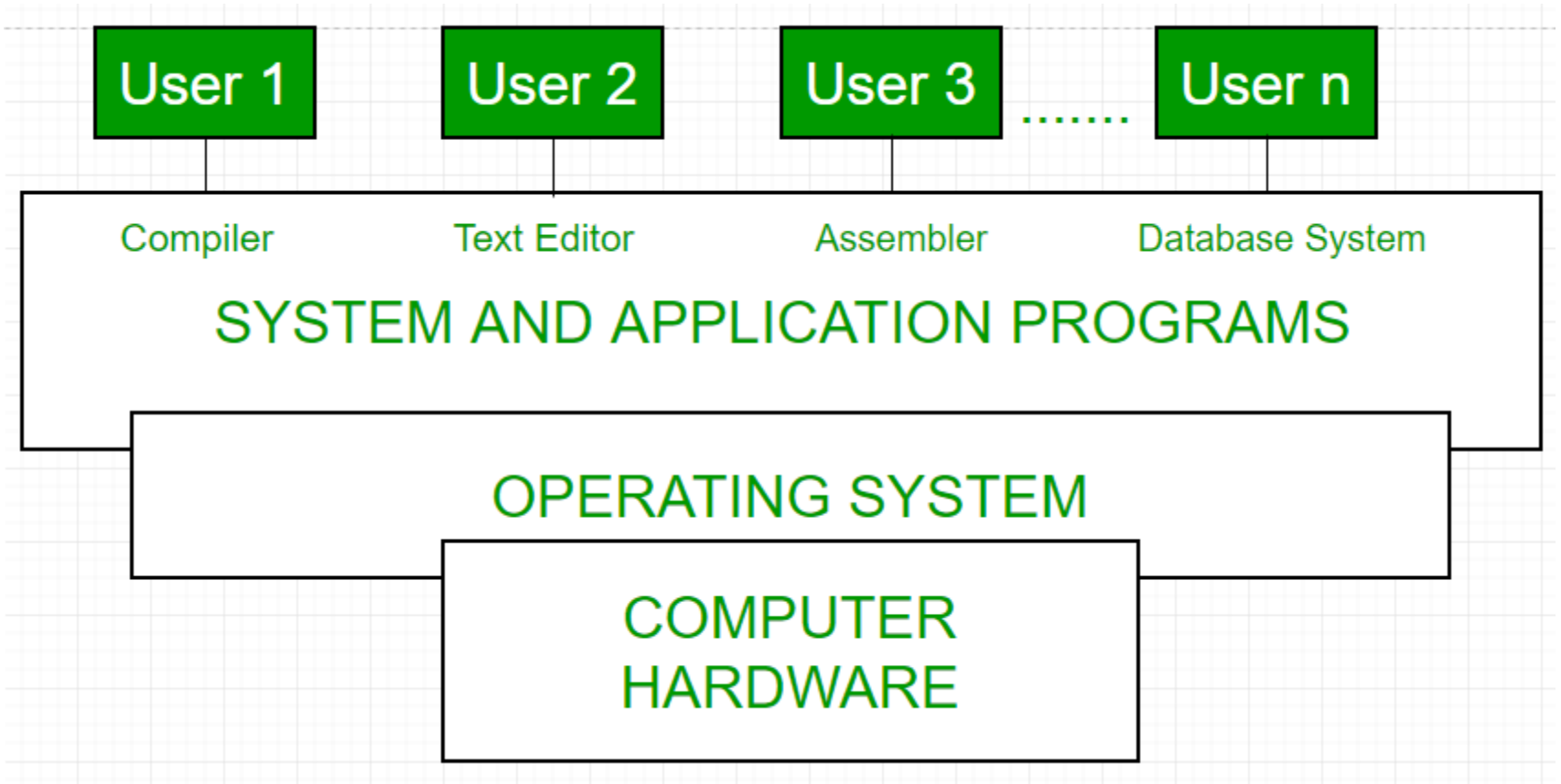
1.1. Overview of operating systems

What is an Operating System?

An operating system is a program on which application programs are executed and acts as a communication bridge (interface) between the user and the computer hardware. The main task of an operating system carries out is the allocation of resources and services, such as the allocation of memory, devices, processors, and information. The operating system also includes programs to manage these resources, such as a traffic controller, a scheduler, a memory management module, I/O programs, and a file system.

Examples of Operating Systems are Windows, Linux, Mac OS, etc.

1.1. Overview of operating systems



1.1. Overview of operating systems

1.2 Functionalities and Characteristics of OS

1. **Resource Management:** The operating system manages and allocates memory, CPU time, and other hardware resources among the various programs and processes running on the computer.
2. **Process Management:** The operating system is responsible for starting, stopping, and managing processes and programs. It also controls the scheduling of processes and allocates resources to them.
3. **Memory Management:** The operating system manages the computer's primary memory and provides mechanisms for optimizing memory usage.
4. **Security:** The operating system provides a secure environment for the user, applications, and data by implementing security policies and mechanisms such as access controls and encryption.

1.2 Functionalities and Characteristics of OS

5. **Job Accounting:** It keeps track of time and resources used by various jobs or users.
6. **File Management:** The operating system is responsible for organizing and managing the file system, including the creation, deletion, and manipulation of files and directories.
7. **Device Management:** The operating system manages input/output devices such as printers, keyboards, mice, and displays. It provides the necessary drivers and interfaces to enable communication between the devices and the computer.
8. **Networking:** The operating system provides networking capabilities such as establishing and managing network connections, handling network protocols, and sharing resources such as printers and files over a network.

1.2 Functionalities and Characteristics of OS

9. **User Interface**: The OS provides a user interface that enables users to interact with the computer system. This can be a Graphical User Interface (GUI), a Command-Line Interface (CLI), or a combination of both.
10. **Backup and Recovery:** The OS provides mechanisms for backing up data and recovering it in case of system failures, errors, or disasters.
11. **Virtualization:** The OS provides virtualization capabilities that allow multiple operating systems or applications to run on a single physical machine. This can enable efficient use of resources and flexibility in managing workloads.
12. **Performance Monitoring:** The operating system provides tools for monitoring and optimizing system performance, including identifying bottlenecks, optimizing resource usage, and analyzing system logs and metrics.

1.2 Functionalities and Characteristics of OS

13. **Time-Sharing:** The operating system enables multiple users to share a computer system and its resources simultaneously by providing time-sharing mechanisms that allocate resources fairly and efficiently.
14. **System Calls:** The operating system provides a set of system calls that enable applications to interact with the operating system and access its resources. System calls provide a standardized interface between applications and the operating system, enabling portability and compatibility across different hardware and software platforms.
15. **Error-detecting Aids:** These contain methods that include the production of dumps, traces, error messages, and other debugging and error-detecting methods.

1.2 Functionalities and Characteristics of OS

Characteristics of Operating System

- **Virtualization:** OS can provide Virtualization capabilities, allowing multiple operating systems or instances of an operating system to run on a single physical machine. This can improve resource utilization and provide isolation between different operating systems or applications.
- **Networking:** OS provide networking capabilities, allowing the computer system to connect to other systems and devices over a network. This can include features such as network protocols, network interfaces, and network security.
- **Scheduling:** OS provide scheduling algorithms that determine the order in which tasks are executed on the system. These algorithms prioritize tasks based on their resource requirements and other factors to optimize system performance.

1.2 Functionalities and Characteristics of OS

Characteristics of Operating System

- **Interproces Communication:** Operating systems provide mechanisms for applications to communicate with each other, allowing them to share data and coordinate their activities.
- **Performance Monitoring:** Operating systems provide tools for monitoring system performance, including CPU usage, memory usage, disk usage, and network activity. This can help identify performance bottlenecks and optimize system performance.
- **Backup and Recovery:** Operating systems provide backup and recovery mechanisms to protect data in the event of system failure or data loss.
- **Debugging:** Operating systems provide debugging tools that allow developers to identify and fix software bugs and other issues in the system.

1.3 Hardware concepts related to OS

Hardware refers to the physical components of a computer system, such as the central processing unit (CPU), memory, hard disk, keyboard, mouse, monitor, and more. Operating system, on the other hand, is a type of software that manages and controls the hardware components of the computer.

1.5 I/O channels

- A channel is an independent hardware component that co-ordinate all I/O to a set of controllers. Computer systems that use I/O channel have special hardware components that handle all I/O operations.
- Channels use separate, independent and low cost processors for its functioning which are called Channel Processors.
- Channel processors are simple, but contains sufficient memory to handle all I/O tasks. When I/O transfer is complete or an error is detected, the channel controller communicates with the CPU using an interrupt, and informs CPU about the error or the task completion.

1.5 I/O channels

I/O Channel is an extension of the DMA concept. It has ability to execute I/O instructions using special-purpose processor on I/O channel and complete control over I/O operations. Processor does not execute I/O instructions itself. Processor initiates I/O transfer by instructing the I/O channel to execute a program in memory.

What is a DMA Controller?

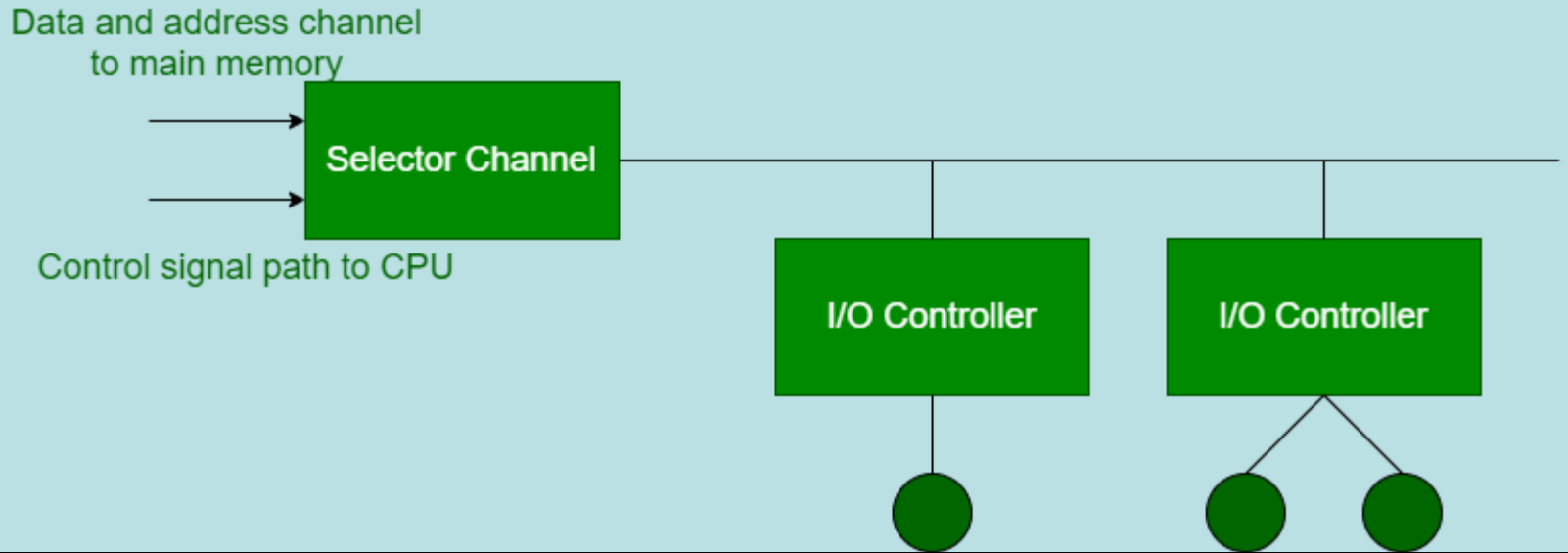
It has the work of transferring the data between Input Output devices and main memory with very less interaction with the processor. The direct Memory Access Controller is a control unit, which has the work of transferring data.

1.5 I/O channels

There are two types of I/O channel

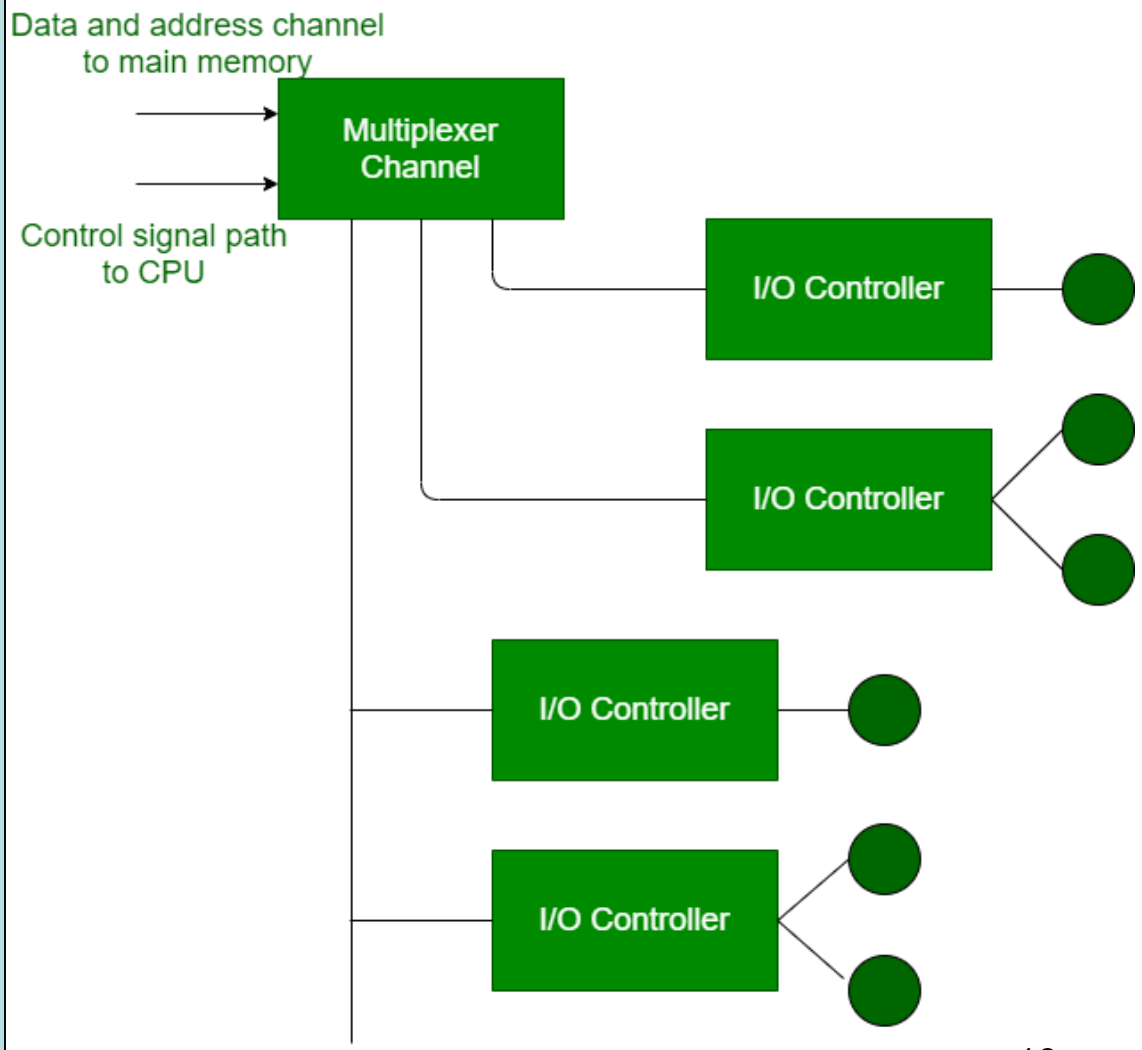
1. Selector channel
2. Multiplexer channel

1. Selector Channel : Selector channel controls multiple high-speed devices. It is dedicated to the transfer of data with one of the devices. In selector channel, each device is handled by a controller or I/O module. It controls the I/O controllers shown in the figure.



1.5 I/O channels

2. Multiplexer Channel : Multiplexer channel is a DMA controller that can handle multiple devices at the same time. It can do block transfers for several devices at once.



1.6 Memory Management

Memory Management:- Memory is the important part of the computer that is used to store the data. Its management is critical to the computer system because the amount of main memory available in a computer system is very limited. At any time, many processes are competing for it. Moreover, to increase performance, several processes are executed simultaneously. For this, we must keep several processes in the main memory, so it is even more important to manage them effectively.

1.6 Memory Management

Role of Memory Management:-

- Memory manager is used to keep track of the status of memory locations, whether it is free or allocated.
- Memory manager permits computers with a small amount of main memory to execute programs larger than the size or amount of available memory. It does this by moving information back and forth between primary memory and secondary memory by using the concept of swapping.
- The memory manager is responsible for protecting the memory allocated to each process from being corrupted by another process.
- Memory managers should enable sharing of memory space between processes. Thus, two programs can reside at the same memory location although at different times.

1.6.1 Memory Management Techniques

Memory Management Techniques are basic techniques that are used in managing the memory in operating system. Memory Management Techniques are basically classified into two categories:

(i) Contiguous Memory allocation

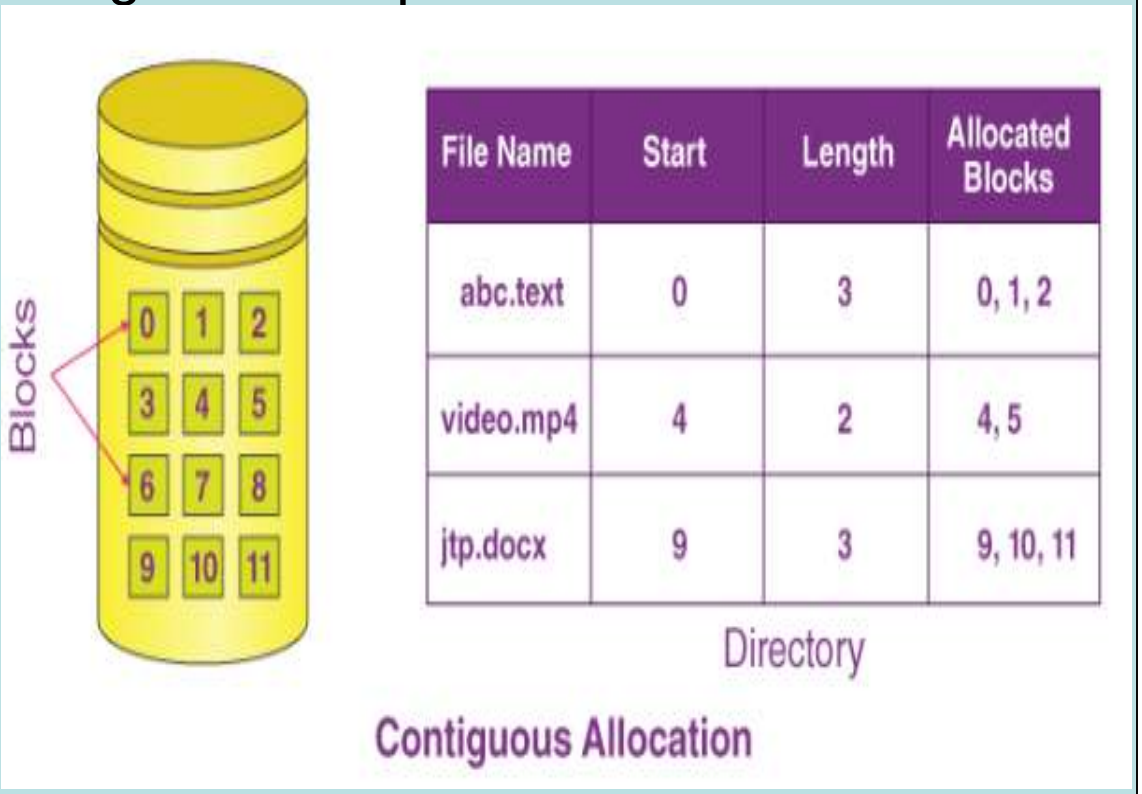
(ii) Non-contiguous Memory allocation

1.6.2 Contiguous & Non-Contiguous Memory allocation

(i) Contiguous Memory allocation

- It is the type of ***memory allocation method.*** When a process requests the memory, a single contiguous section of memory blocks is allotted depending on its requirements.

Three files are there in the directory. The starting block, along with the length of each file, is mentioned in the table. Thus, we can see in this table that all the contiguous blocks get assigned to every file as per their need.

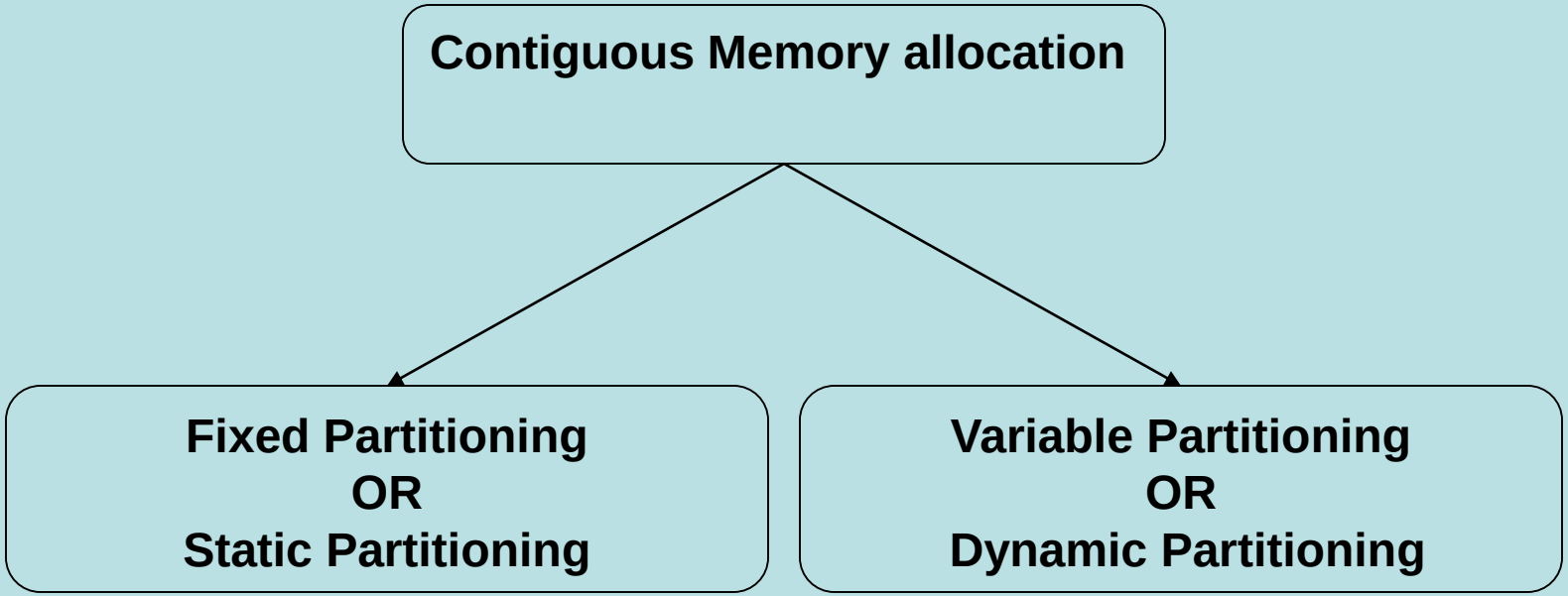


File Name	Start	Length	Allocated Blocks
abc.text	0	3	0, 1, 2
video.mp4	4	2	4, 5
jtp.docx	9	3	9, 10, 11

Directory

Contiguous Allocation

1.6.2 Contiguous & Non-Contiguous Memory allocation



1.6.2 Contiguous & Non-Contiguous Memory allocation

1. Fixed Partitioning OR Static Partitioning

The earliest and one of the simplest technique which can be used to load more than one processes into the main memory is Fixed partitioning or Contiguous memory allocation.

In this technique, the main memory is divided into partitions of equal or different sizes. The operating system always resides in the first partition while the other partitions can be used to store user processes. The memory is assigned to the processes in contiguous way.

1. The partitions cannot overlap.
2. A process must be contiguously present in a partition for the execution.

1.6.2 Contiguous & Non-Contiguous Memory allocation

1. Fixed Partitioning OR Static Partitioning

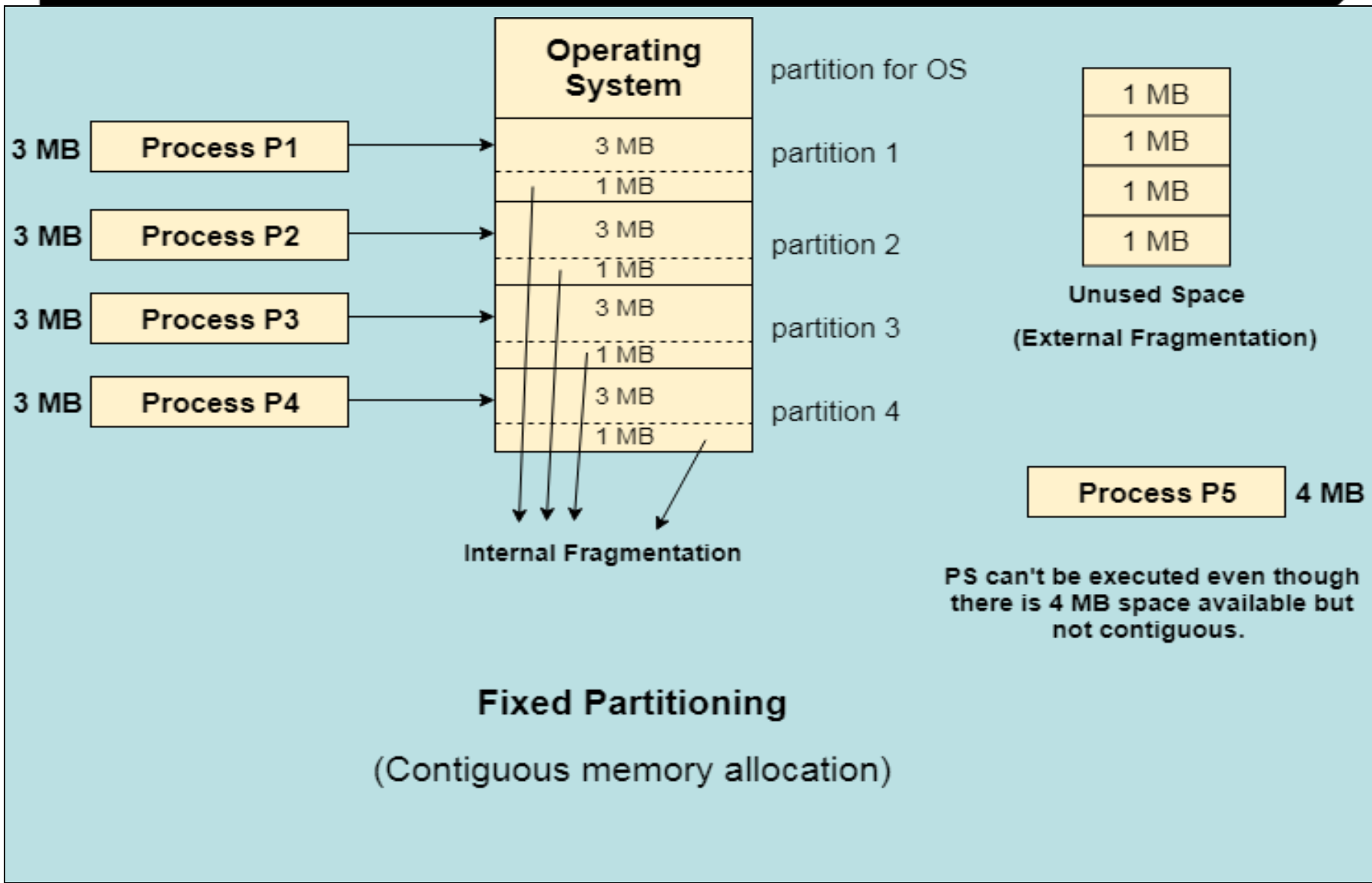
There are various disadvantages of using this technique.

A) Internal Fragmentation

If the size of the process is lesser then the total size of the partition then some size of the partition get wasted and remain unused. This is wastage of the memory and called internal fragmentation.

As shown in the image below, the 4 MB partition is used to load only 3 MB process and the remaining 1 MB got wasted.

1.6.2 Contiguous & Non-Contiguous Memory allocation



1.6.2 Contiguous & Non-Contiguous Memory allocation

1. Fixed Partitioning OR Static Partitioning

B) External Fragmentation

The total unused space of various partitions cannot be used to load the processes even though there is space available but not in the contiguous form.

As shown in the image above, the remaining 1 MB space of each partition cannot be used as a unit to store a 4 MB process(P5). Despite of the fact that the sufficient space is available to load the process, process will not be loaded.

1.6.2 Contiguous & Non-Contiguous Memory allocation

1. Fixed Partitioning OR Static Partitioning

There are various disadvantages of using this technique.

C) Limitation on the size of the process

If the process size is larger than the size of maximum sized partition then that process cannot be loaded into the memory. Therefore, a limitation can be imposed on the process size that is it cannot be larger than the size of the largest partition.

1.6.2 Contiguous & Non-Contiguous Memory allocation

1. Fixed Partitioning OR Static Partitioning

There are various disadvantages of using this technique.

D) Degree of multiprogramming is less

By Degree of multi programming, we simply mean the maximum number of processes that can be loaded into the memory at the same time. In fixed partitioning, the degree of multiprogramming is fixed and very less due to the fact that the size of the partition cannot be varied according to the size of processes.

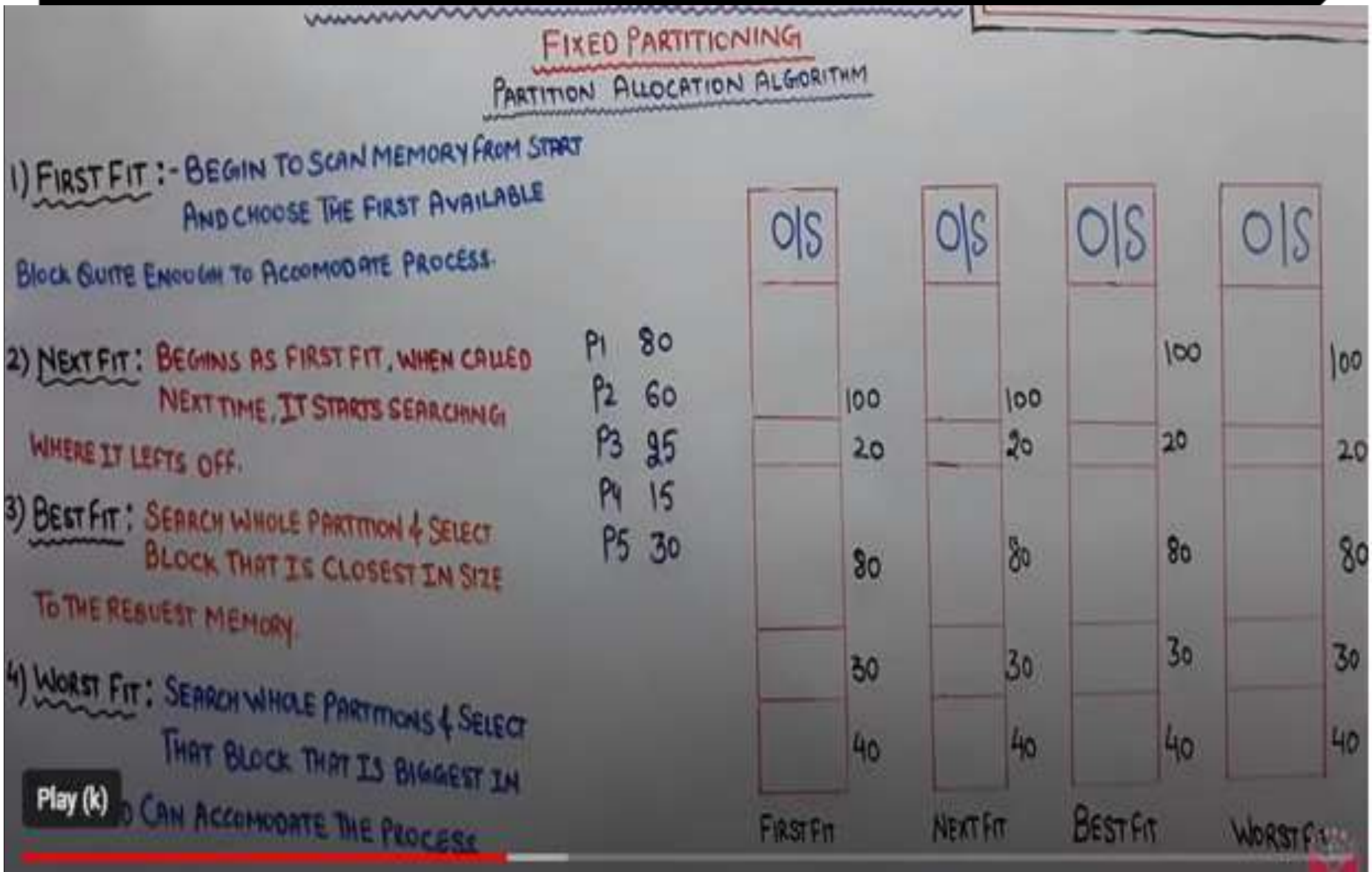
1.6.2 Contiguous & Non-Contiguous Memory allocation

Partition Allocation Methods(algorithm) in Operating System

There are 4 different methods or algorithm to allocate partitions for the process.

1. First Fit Allocation ·
2. Next Fit Allocation.
3. Best Fit Allocation ·
4. Worst Fit Allocation ·

1.6.2 Contiguous & Non-Contiguous Memory allocation



1.6.2 Contiguous & Non-Contiguous Memory allocation

1.6.2 Contiguous & Non-Contiguous Memory allocation

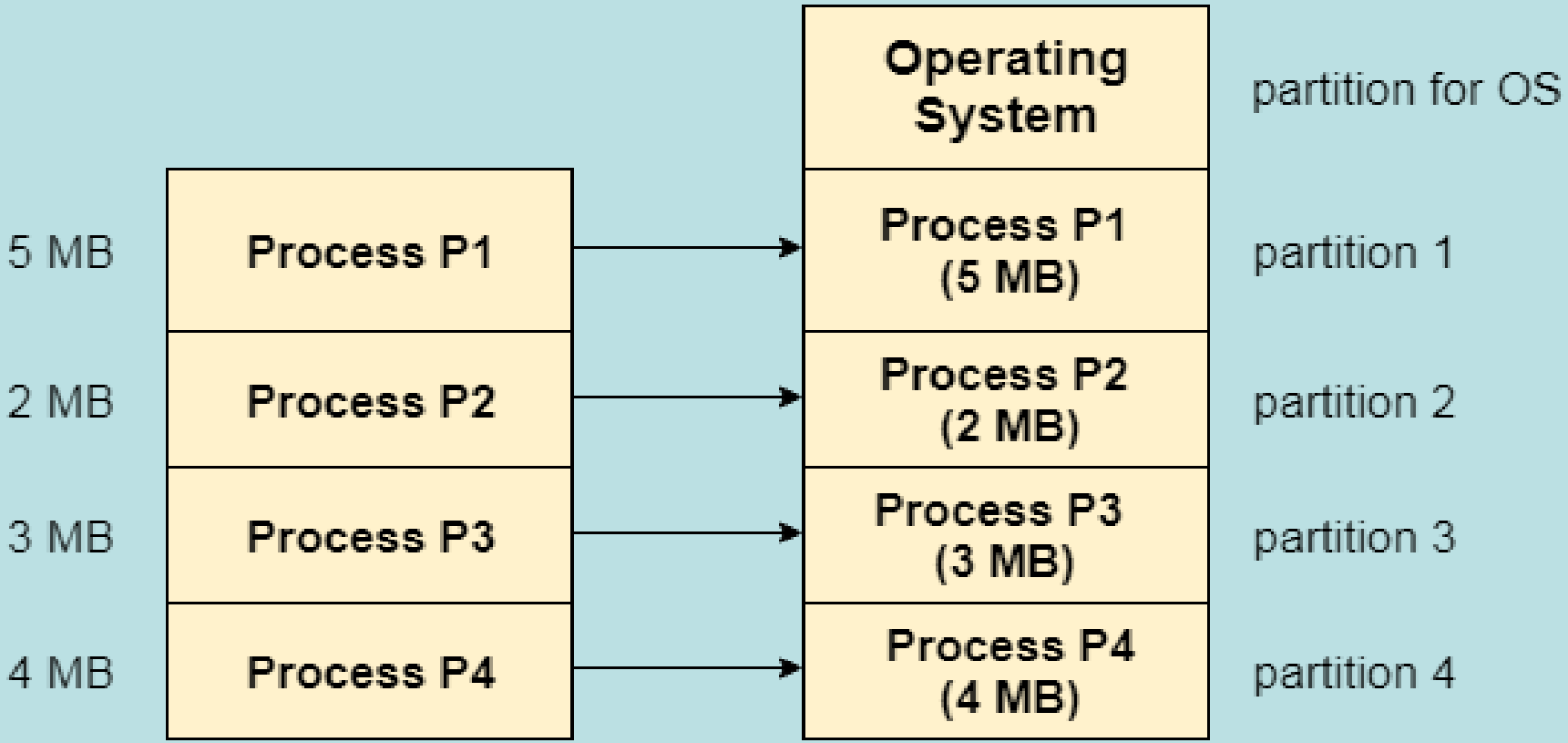
2. Variable Partitioning OR Dynamic Partitioning

Dynamic partitioning tries to overcome the problems caused by fixed partitioning. In this technique, the partition size is not declared initially. It is declared at the time of process loading.

The first partition is reserved for the operating system. The remaining space is divided into parts. The size of each partition will be equal to the size of the process. The partition size varies according to the need of the process so that the internal fragmentation can be avoided.

1.6.2 Contiguous & Non-Contiguous Memory allocation

2. Variable Partitioning OR Dynamic Partitioning



Dynamic Partitioning

(Process Size = Partition Size)

1.6.2 Contiguous & Non-Contiguous Memory allocation

Advantages of Dynamic Partitioning over fixed partitioning

1. No Internal Fragmentation

The fact that the partitions in dynamic partitioning are created according to the need(size) of the process, It is clear that there will not be any internal fragmentation because there will not be any unused remaining space in the partition.

2. No Limitation on the size of the process

In Fixed partitioning, the process with the size greater than the size of the largest partition could not be executed due to the lack of sufficient contiguous memory. In Dynamic partitioning, the process size can't be restricted since the partition size is decided according to the process size.

3. Degree of multiprogramming is dynamic

Due to the absence of internal fragmentation, there will not be any unused space in the partition hence more processes can be loaded in the memory at the same time.

1.6.2 Contiguous & Non-Contiguous Memory allocation

Disadvantages of dynamic partitioning

External Fragmentation

Absence of internal fragmentation doesn't mean that there will not be external fragmentation.

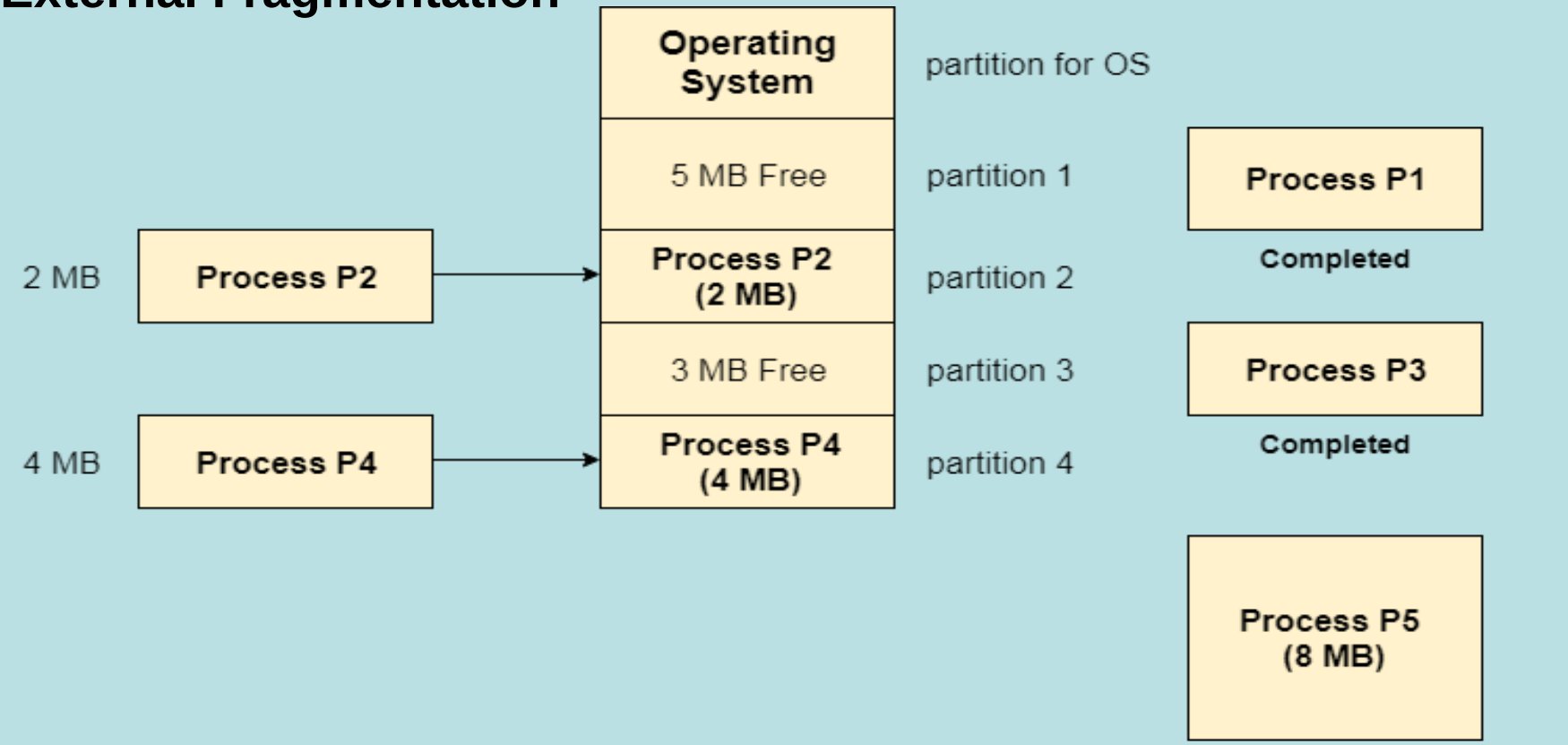
Let's consider three processes P1 (5 MB) and P2 (2 MB) and P3 (3 MB) are being loaded in the respective partitions of the main memory.

After some time P1 and P3 got completed and their assigned space is freed. Now there are two unused partitions (5 MB and 3 MB) available in the main memory but they cannot be used to load a 8 MB process in the memory since they are not contiguously located.

The rule says that the process must be contiguously present in the main memory to get executed. We need to change this rule to avoid external fragmentation.

1.6.2 Contiguous & Non-Contiguous Memory allocation

External Fragmentation



External Fragmentation in Dynamic Partitioning

X

1.6.2 Contiguous & Non-Contiguous Memory allocation

Compaction

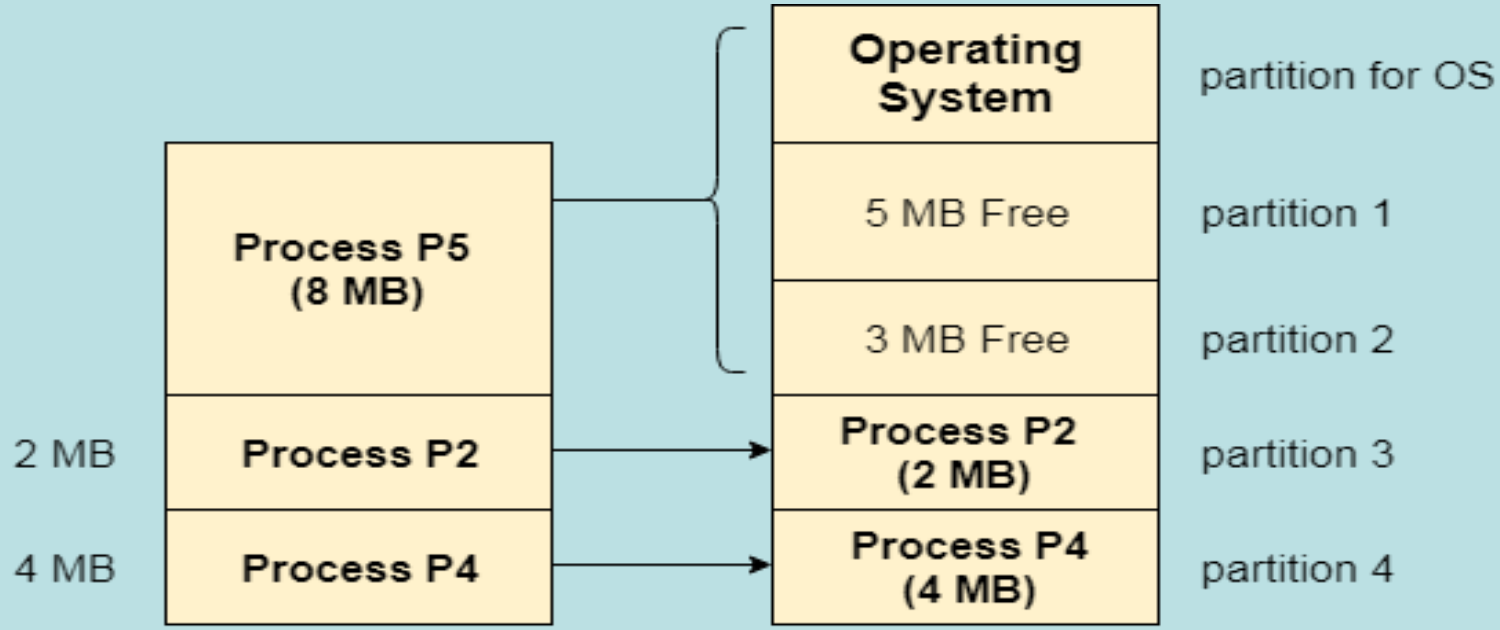
We got to know that the dynamic partitioning suffers from external fragmentation. However, this can cause some serious problems. we need to change the rule which says that the process can't be stored in the different places in the memory.

We can also use compaction to minimize the probability of external fragmentation. In compaction, all the free partitions are made contiguous and all the loaded partitions are brought together.

By applying this technique, we can store the bigger processes in the memory. The free partitions are merged which can now be allocated according to the needs of new processes. This technique is also called defragmentation.

1.6.2 Contiguous & Non-Contiguous Memory allocation

As shown in the image, the process P5, which could not be loaded into the memory due to the lack of contiguous space, can be loaded now in the memory since the free partitions are made contiguous.



Now PS can be loaded into memory because the free space is now made contiguous by compaction

Compaction

1.6.2 Contiguous & Non-Contiguous Memory allocation

Non-Contiguous Memory allocation

It allows a process to obtain multiple memory blocks in various locations in memory based on its requirements. The non-contiguous memory allocation also reduces memory wastage caused by ***internal*** and ***external*** fragmentation because it uses the memory holes created by internal and external fragmentation.

- Non-contiguous memory allocation divides the process into blocks ***(pages or segments)*** that are allocated to different areas of memory space based on memory availability.
-

1.7 Paging

Need of Paging:-

Disadvantage of Dynamic Partitioning

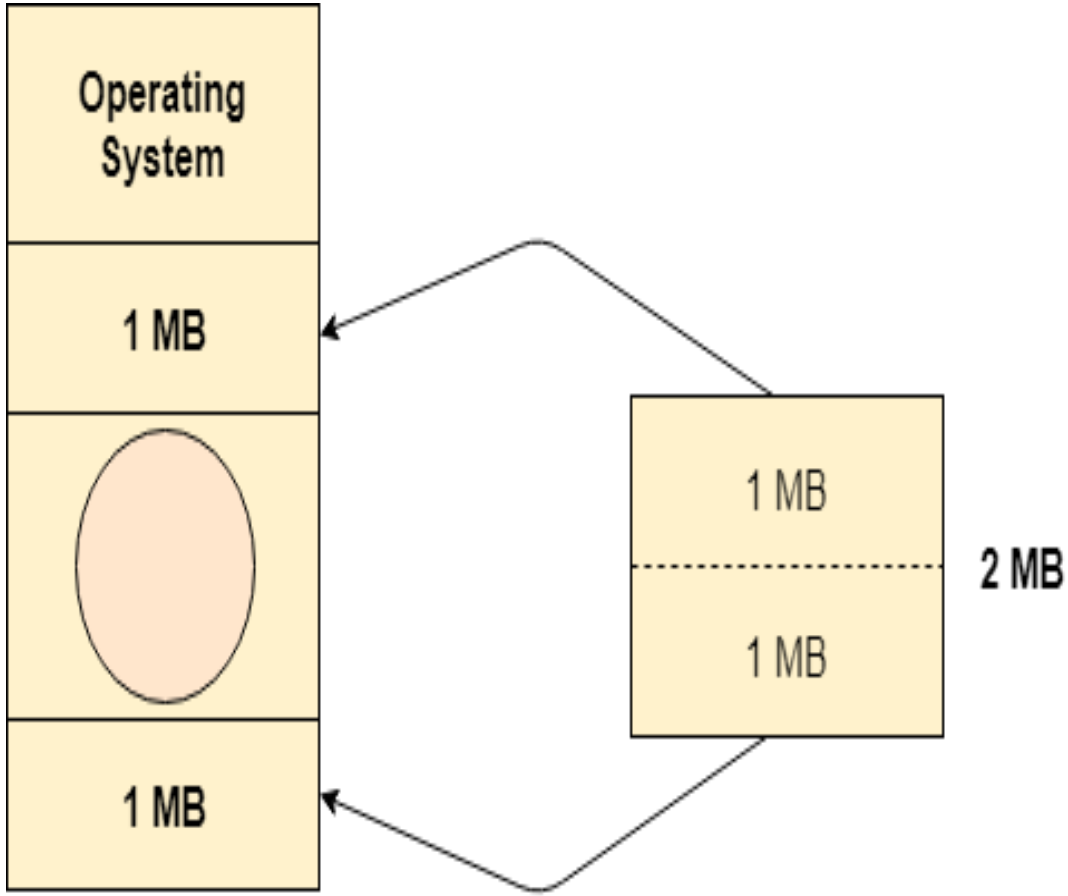
- The main disadvantage of Dynamic Partitioning is External fragmentation. Although, this can be removed by Compaction but as we have discussed earlier, the compaction makes the system inefficient.
- We need to find out a mechanism which can load the processes in the partitions in a more optimal way. Let us discuss a dynamic and flexible mechanism called paging.

1.7 Paging

Need of Paging:-

Lets consider a process P1 of size 2 MB and the main memory which is divided into three partitions. Out of the three partitions, two partitions are holes of size 1 MB each.

P1 needs 2 MB space in the main memory to be loaded. We have two holes of 1 MB each but they are not contiguous.



The process needs to be divided into two parts to get stored at two different places.

1.7 Paging

Need of Paging:-

Although, there is 2 MB space available in the main memory in the form of those holes but that remains useless until it become contiguous. This is a serious problem to address.

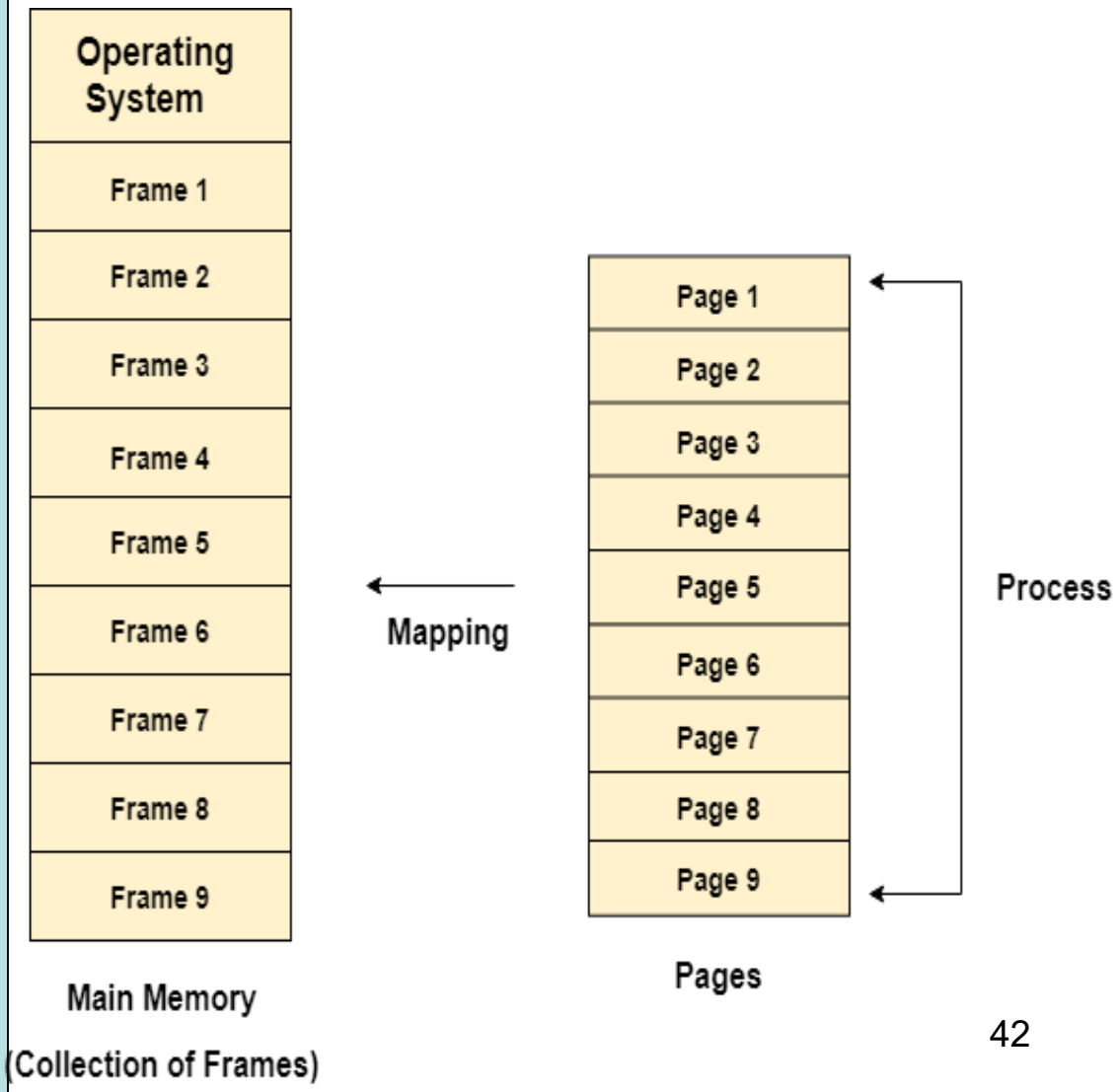
We need to have some kind of mechanism which can store one process at different locations of the memory.

The Idea behind paging is to divide the process in pages so that, we can store them in the memory at different holes

1.7 Paging
1.7.1 Demand Paging

Paging:-

Paging is a storage mechanism used in OS to retrieve processes from secondary storage to the main memory as pages. The primary concept behind paging is to break each process into individual pages. Thus the primary memory would also be separated into frames.

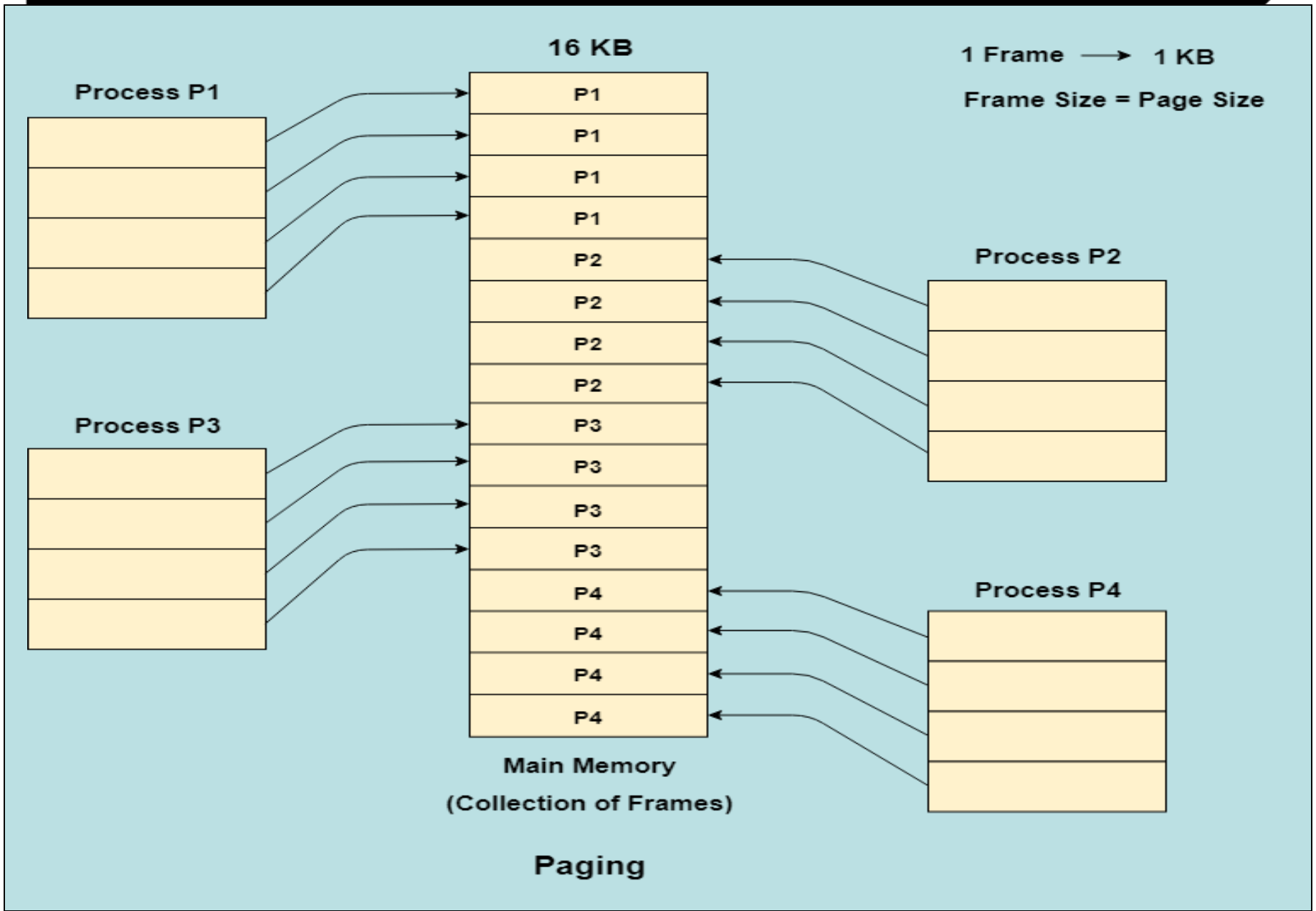


1.7 Paging
1.7.1 Demand Paging

Paging:-

- Let us consider the main memory size 16 Kb and Frame size is 1 KB therefore the main memory will be divided into the collection of 16 frames of 1 KB each.
- There are 4 processes in the system that is P1, P2, P3 and P4 of 4 KB each. Each process is divided into pages of 1 KB each so that one page can be stored in one frame.
- Initially, all the frames are empty therefore pages of the processes will get stored in the contiguous way.
- Frames, pages and the mapping between the two is shown in the image below.

1.7 Paging

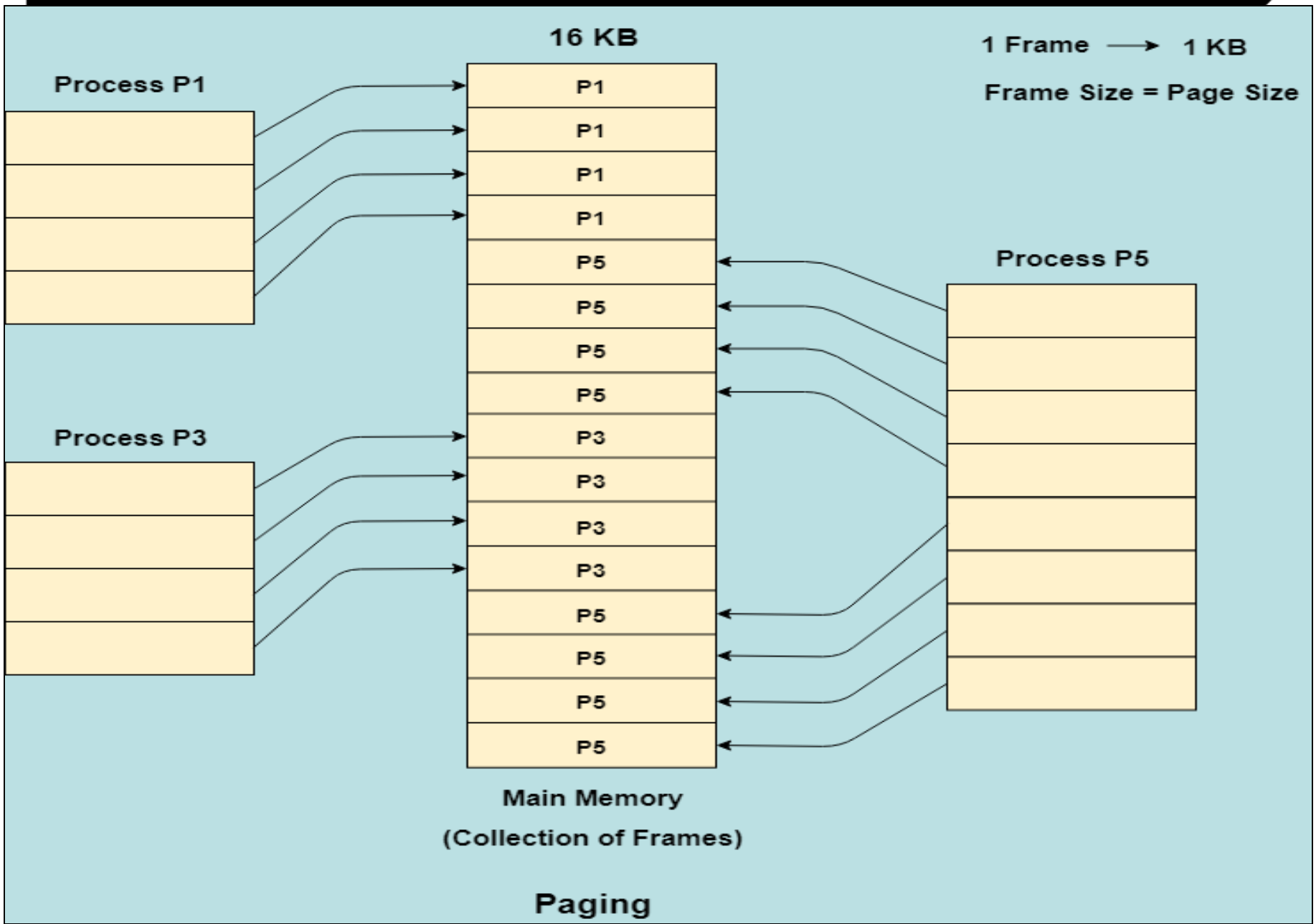

16 KB
1 Frame → 1 KB
Frame Size = Page Size
Process P1
Process P2
Process P3
Process P4
P1
P1
P1
P1
P2
P2
P2
P2
P3
P3
P3
P3
P4
P4
P4
P4
Main Memory
(Collection of Frames)
Paging

1.7 Paging

Paging:-

- Let us consider that, P2 and P4 are moved to waiting state after some time. Now, 8 frames become empty and therefore other pages can be loaded in that empty place. The process P5 of size 8 KB (8 pages) is waiting inside the ready queue.
- we have 8 non contiguous frames available in the memory and paging provides the flexibility of storing the process at the different places. Therefore, we can load the pages of process P5 in the place of P2 and P4.

1.7 Paging


16 KB
1 Frame → 1 KB
Frame Size = Page Size
Process P1
Process P3
Process P5
P1
P1
P1
P1
P5
P5
P5
P5
P3
P3
P3
P3
P5
P5
P5
P5
Main Memory
(Collection of Frames)
Paging

1.7.1 Demand Paging

Demand Paging:-

According to the concept of Virtual Memory, in order to execute some process, only a part of the process needs to be present in the main memory which means that only a few pages will only be present in the main memory at any time. However, deciding, which pages need to be kept in the main memory and which need to be kept in the secondary memory, is going to be difficult because we cannot say in advance that a process will require a particular page at particular time.

Therefore, to overcome this problem, there is a concept called Demand Paging is introduced. It suggests keeping all pages of the frames in the secondary memory until they are required. In other words, it says that do not load any page in the main memory until it is required.

1.7.1 Demand Paging

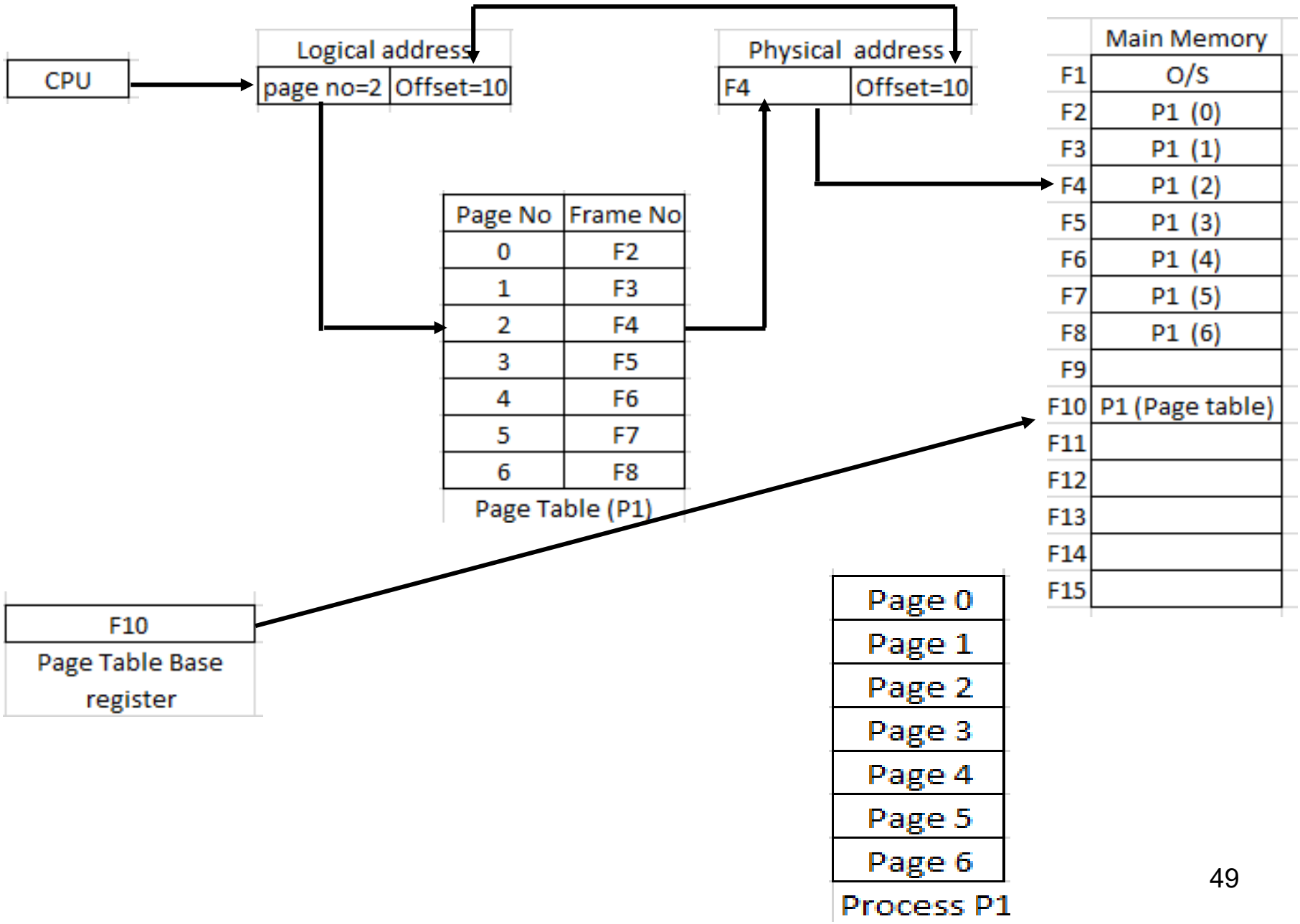
Demand Paging:-

Whenever any page is referred for the first time in the main memory, then that page will be found in the secondary memory. After that, it may or may not be present in the main memory depending upon the page replacement algorithm **What is a Page Fault?**

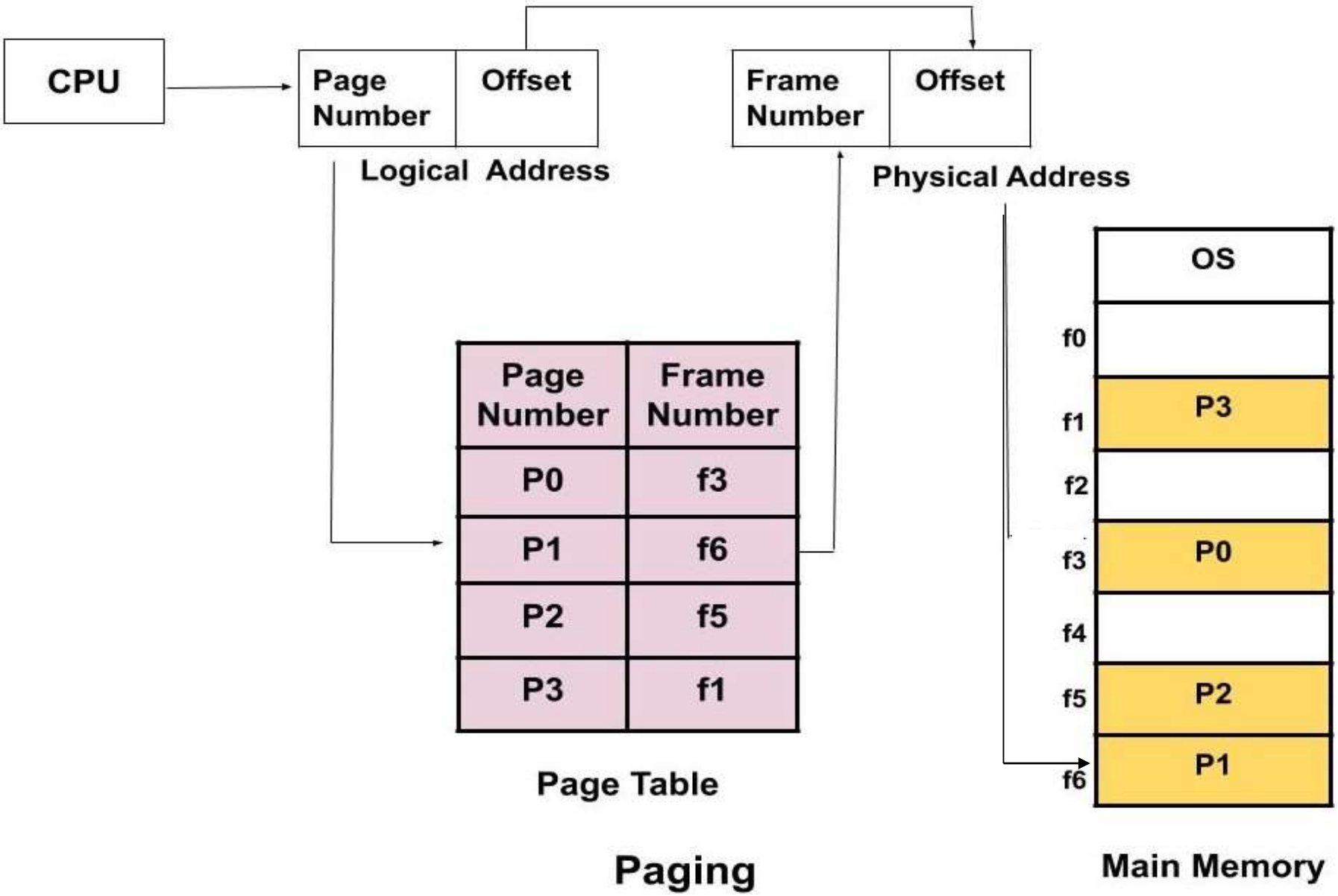
If the referred page is not present in the main memory then there will be a miss and the concept is called Page miss or page fault.

The CPU has to access the missed page from the secondary memory. If the number of page fault is very high then the effective access time of the system will become very high.

Conversion of Logical Address to Physical Address



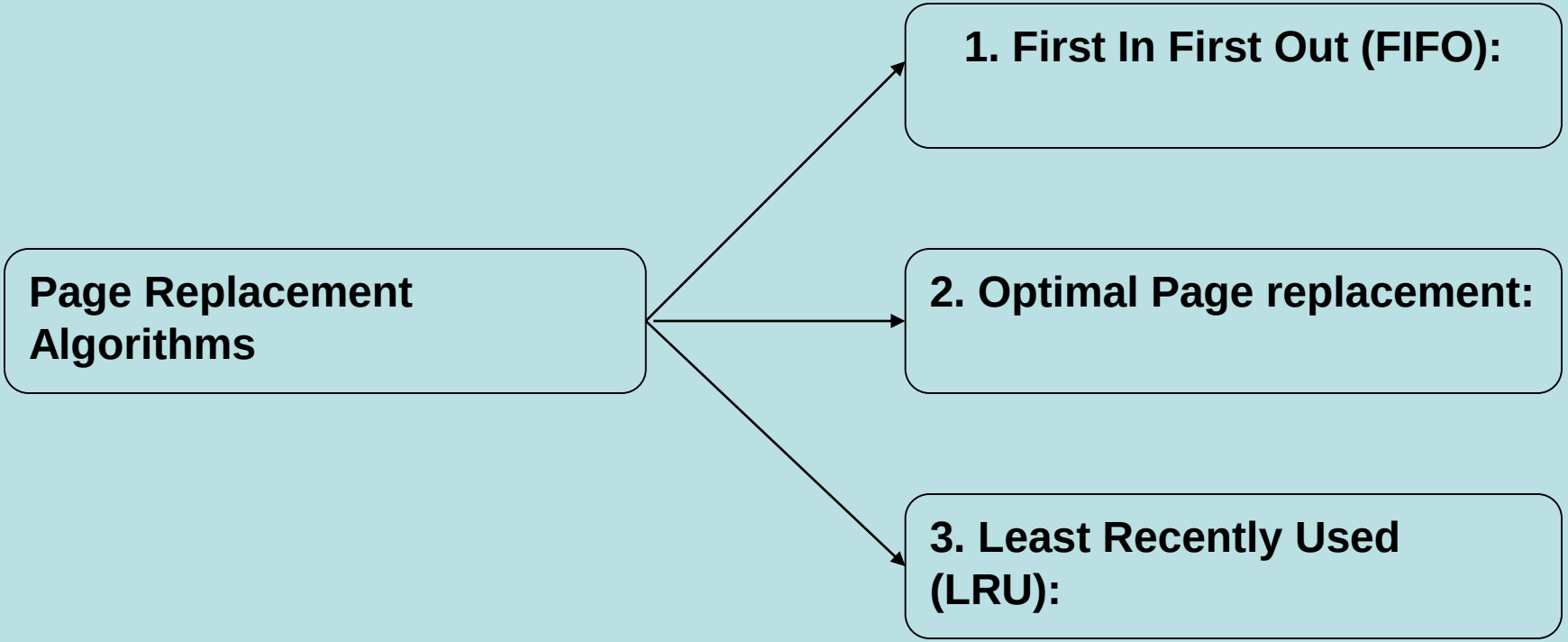
Conversion of Logical address to Physical address



1.7.2 Page Replacement Concept

When a page that is residing in virtual memory is requested by a process for its execution, the Operating System needs to decide which page will be replaced by this requested page. This process is known as page replacement.

There are three page replacement algorithms.



1.7.2 Page Replacement Concept

1. First In First Out (FIFO):

This is the first basic algorithm of Page Replacement Algorithms. This algorithm is basically dependent on the number of frames used. Then each frame takes up the certain page and tries to access it. When the frames are filled then the actual problem starts.

- If the page to be searched is found among the frames then, this process is known as **Page Hit**.
- If the page to be searched is not found among the frames then, this process is known as **Page Fault**.
- When Page Fault occurs this problem arises, then the First In First Out Page Replacement Algorithm comes into picture.

1.7.2 Page Replacement Concept

1. First In First Out (FIFO):

The First In First Out (FIFO) Page Replacement Algorithm removes the Page in the frame which is allotted long back. This means the useless page which is in the frame for a longer time is removed and the new page which is in the ready queue and is ready to occupy the frame is allowed by the First In First Out Page Replacement.

1.7.2 Page Replacement Concept

1. First In First Out (FIFO):

Example:- Consider the reference string 6, 1, 1, 2, 0, 3, 4, 6, 0, 2, 1, 2, 1, 2, 0, 3, 2, 1, 2, 0 for a memory with three frames and calculate number of page faults by using FIFO (First In First Out) Page replacement algorithms.

6, 1, 1, 2, 0, 3, 4, 6, 0, 2, 1, 2, 1, 2, 0, 3, 2, 1, 2, 0

S. no	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F3				2	2	2	4	4	4	2	2	2	2	2	2	2	2	2	2	0
F2		1	1	1	1	3	3	3	0	0	0	0	0	0	0	3	3	3	3	3
F1	6	6	6	6	0	0	0	6	6	6	1	1	1	1	1	1	1	1	1	1
Hit (H)/ Fault (F)	F	F	H	F	F	F	F	F	F	F	F	H	H	H	H	F	H	H	H	F

- Number of Page Hits = 8
- Number of Page Faults = 12

1.7.2 Page Replacement Concept

2. Optimal Page replacement:

The OPTIMAL Page Replacement Algorithms works on a certain principle. The principle is:

- Replace the Page which is not used in the Longest Dimension of time in future
- This principle means that after all the frames are filled then, see the future pages which are to occupy the frames. Go on checking for the pages which are already available in the frames. Choose the page which is at last.

1.7.2 Page Replacement Concept

2. Optimal Page replacement:

Example:-Consider the reference string 6, 1, 1, 2, 0, 3, 4, 6, 0, 2, 1, 2, 1, 2, 0, 3, 2, 1, 2, 0 for a memory with three frames and calculate number of page faults by using Optimal Page replacement

Sr No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F3				2	2	3	4	4	4	2	2	2	2	2	2	2	2	2	2	2
F2		1	1	1	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3
F1	6	6	6	6	6	6	6	6	6	6	1	1	1	1	1	1	1	1	1	0
Hit(H)/ Fault(F)	F	F	H	F	F	F	F	H	H	F	F	H	H	H	H	F	H	H	H	F

- Number of Page Hits = 10
- Number of Page Faults = 10

1.7.2 Page Replacement Concept

3. Least Recently Used (LRU):

The Least Recently Used (LRU) Page Replacement Algorithms works on a certain principle. The principle is:

- Replace the page with the page which is less dimension of time recently used page in the **past.**

Example:- Suppose Reference String is: 6, 1, 1, 2, 0, 3, 4, 6, 0

- The pages with page numbers 6, 1, 2 are in the frames occupying the frames.
- Now, we need to allot a space for the page numbered 0.
- Now, we need to travel back into the past to check which page can be replaced.
- 6 is the oldest page which is available in the Frame.
- So, replace 6 with the page numbered 0.

1.7.2 Page Replacement Concept

3. Least Recently Used (LRU):

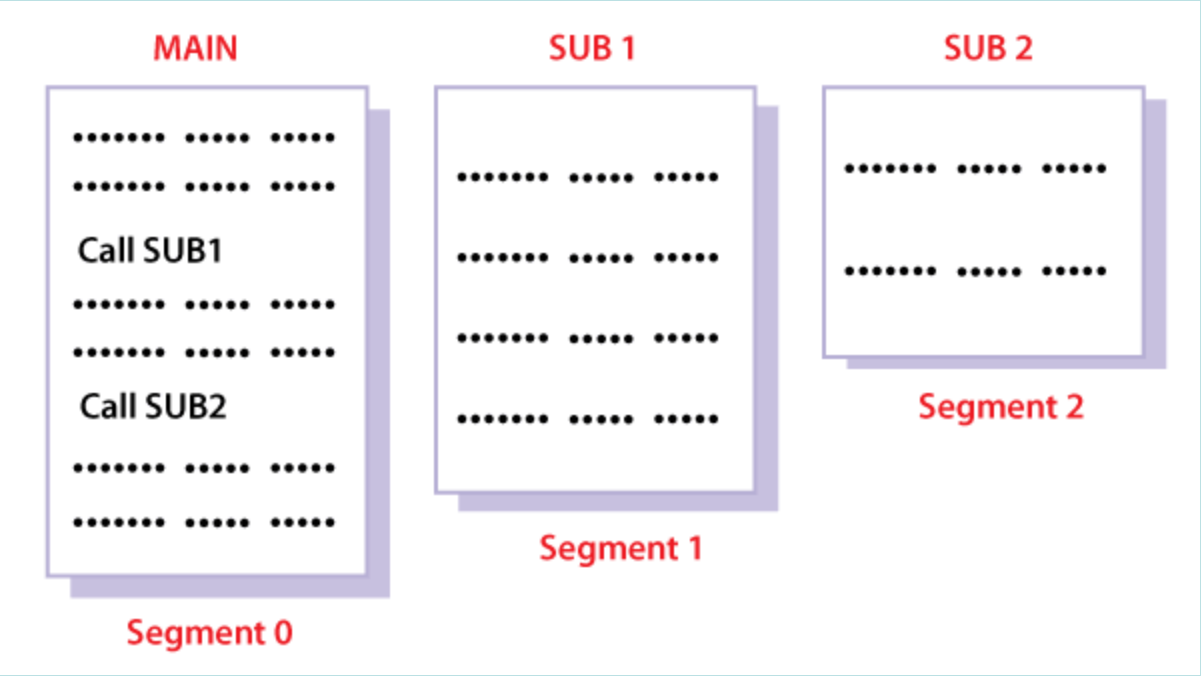
Example:-Consider the reference string 6, 1, 1, 2, 0, 3, 4, 6, 0, 2, 1, 2, 1, 2, 0, 3, 2, 1, 2, 0 for a memory with three frames and calculate number of page faults by using Optimal Page replacement

Sr No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
F3				2	2	2	4	4	4	2	2	2	2	2	2	3	3	3	3	3
F2		1	1	1	1	3	3	3	0	0	0	0	0	0	0	0	2	2	2	2
F1	6	6	6	6	0	0	0	6	6	6	1	1	1	1	1	1	1	1	1	0
Hit(H)/ Fault(F)	F	F	H	F	F	F	F	F	F	F	F	H	H	H	H	F	F	H	H	F

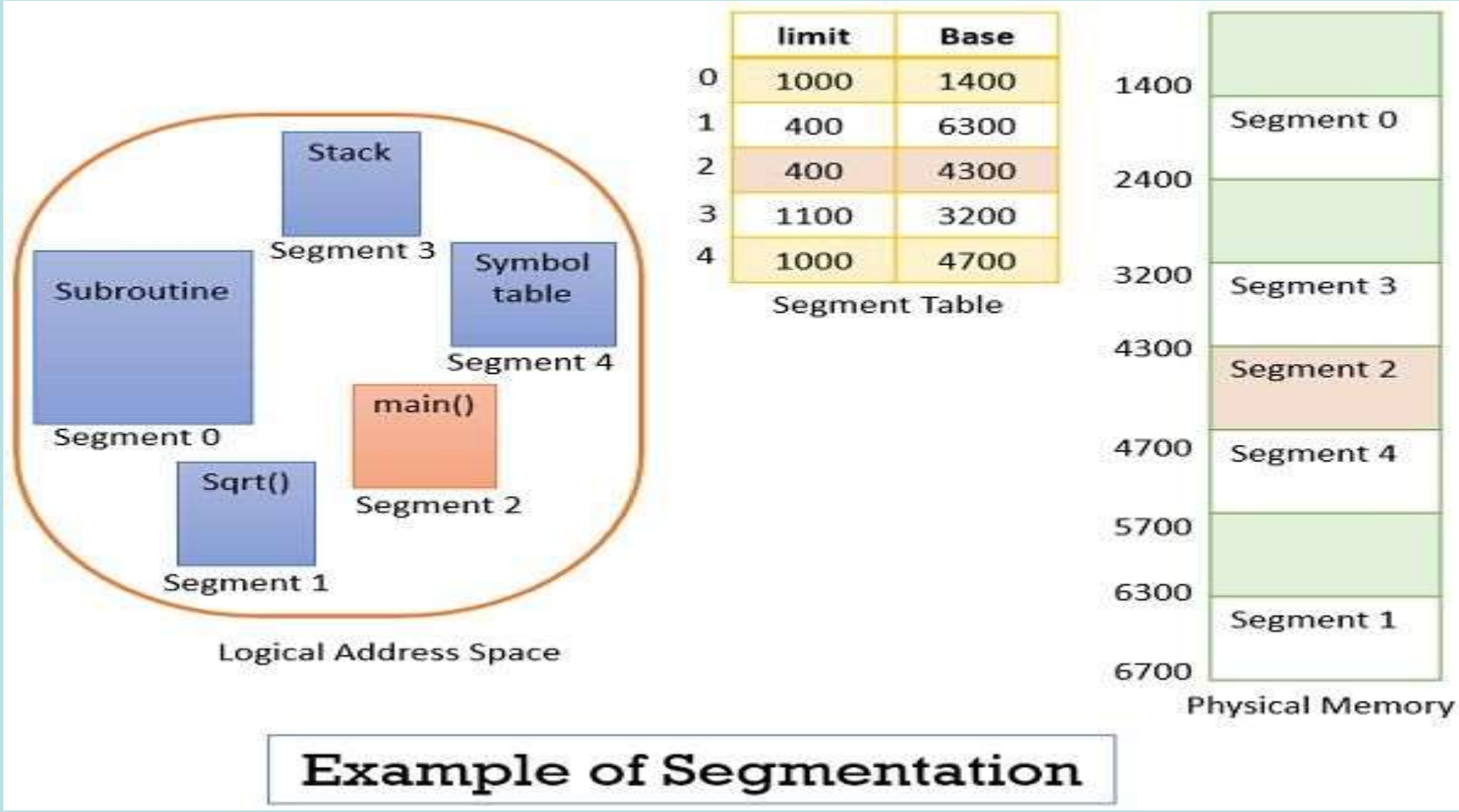
- Number of Page Hits = 07
- Number of Page Faults = 13

1.8 Segmentation - Segment with paging

It is better to have segmentation which divides the process into the segments. Each segment contains the same type of functions such as the main function can be included in one segment and the library functions can be included in the other segment.

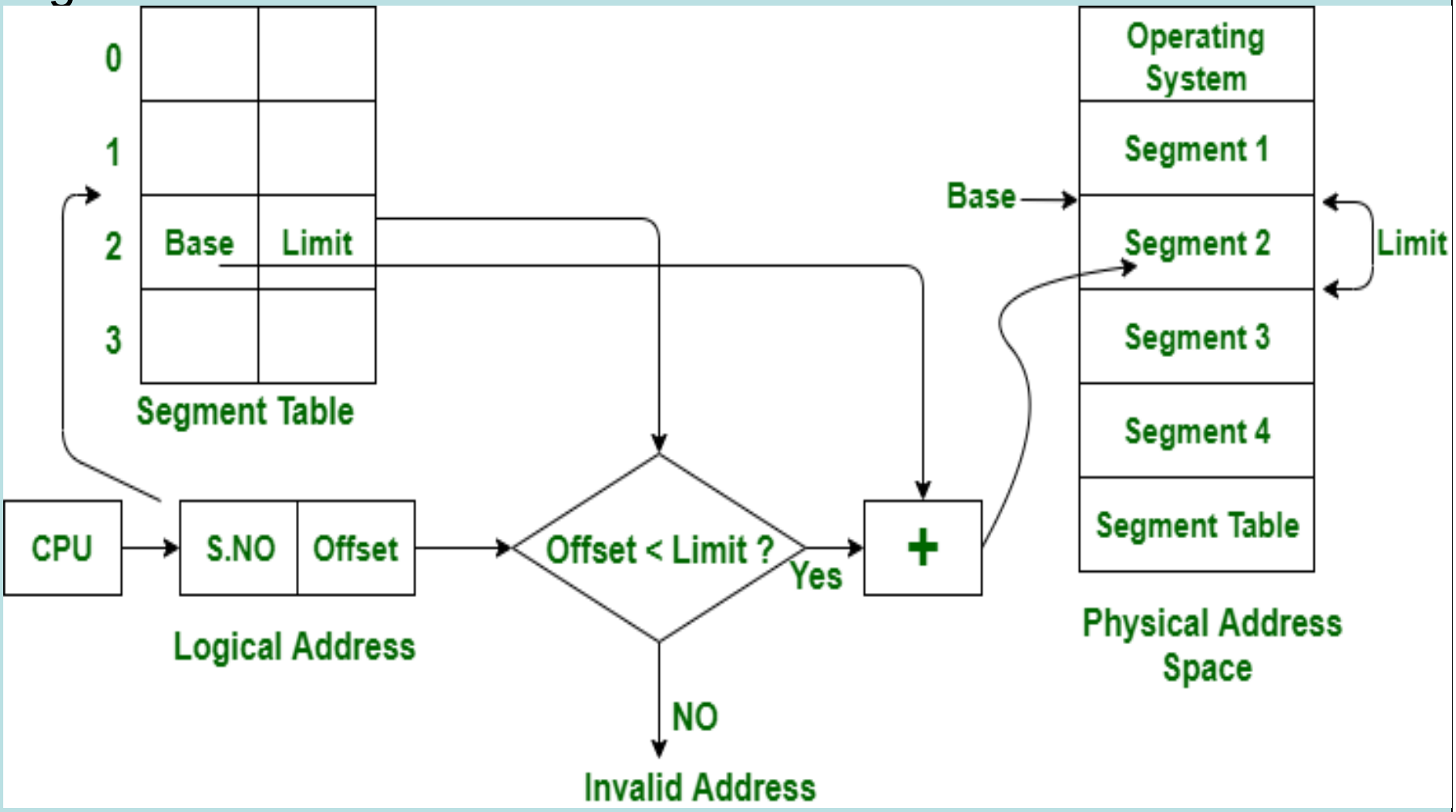


1.8 Segmentation - Segment with paging



1.8 Segmentation - Segment with paging

Conversion of logical address into physical address in segmentation:-



1.6.2 Contiguous & Non-Contiguous Memory allocation

Advantages of Non-Contiguous Memory allocation

1. It has the advantage of reducing memory waste, but it increases overhead because of the address translation.

Disadvantages of Non-Contiguous Memory allocation

1. The downside of this memory allocation is that the access is slow because you must reach the other nodes using pointers and traverse them.

1.6.2 Contiguous & Non-Contiguous Memory allocation

Contiguous Memory Allocation	Non-Contiguous Memory Allocation
It allocates only a single memory contiguous block to the process.	It separates the process into numerous blocks, each of which is assigned to a different memory address space.
It is very faster in execution in comparison to non-contiguous memory allocation.	It is slower in execution in comparison to contiguous memory allocation.
There is no overhead of address translation during execution because the process is stored in contiguous memory space in contiguous memory allocation.	There is an overhead of address translation during process execution because the process blocks are scattered across the memory space.
Operating system keeps a table that lists all available and occupied partitions in the contiguous memory allocation.	In the non-contiguous memory allocation, each process must keep a table that primarily contains each block's base addresses acquired by the memory.

1.6.2 Contiguous & Non-Contiguous Memory allocation

Contiguous Memory Allocation	Non-Contiguous Memory Allocation
The Operating System can better control contiguous memory allocation	The Non-Contiguous Memory Allocation is difficult for the Operating System to manage.
Contiguous memory allocation contains two memory allocations: single partition and multi-partition.	It contains Paging and Segmentation.
The memory space is partitioned into fixed-sized partitions in the contiguous memory allocation, and each partition is only assigned to one process.	It is broken into several blocks, which are then placed in different areas of the memory based on available memory space.
Wastage of memory	No wastage of memory
Swapped-in processes are placed in the initially allotted space in the contiguous memory allocation.	Swapped-in processes in non-contiguous memory allocation can be organized in any location in memory.
Both internal and exterior fragmentation	The non-Contiguous memory allocation

1.6.3 Logical & Physical Memory – Conversion of Logical to Physical address

Logical Memory(address):- The **logical address** is a virtual address created by the CPU of the computer system. The logical address of a program is generated when the program is running. A group of several logical address is referred to a **logical address space**. The logical address is basically used as a reference to access the physical memory locations.

In computer systems, a hardware device named **memory management unit** (MMU) is used to map the logical address to its corresponding physical address. However, the logical address of a program is visible to the computer user.

1.6.3 Logical & Physical Memory – Conversion of Logical to Physical address

Physical Memory(address):-

The **physical address** of a computer program is one that represents a location in the memory unit of the computer. The physical address is not visible to the computer user. The MMU of the system generates the physical address for the corresponding logical address.

The physical address is accessed through the corresponding logical address because a user cannot directly access the physical address. For running a computer program, it requires a physical memory space. Therefore, the logical address has to be mapped with the physical address before the execution of the program.

1.9 Virtual Memory Concept

Virtual Memory Concept

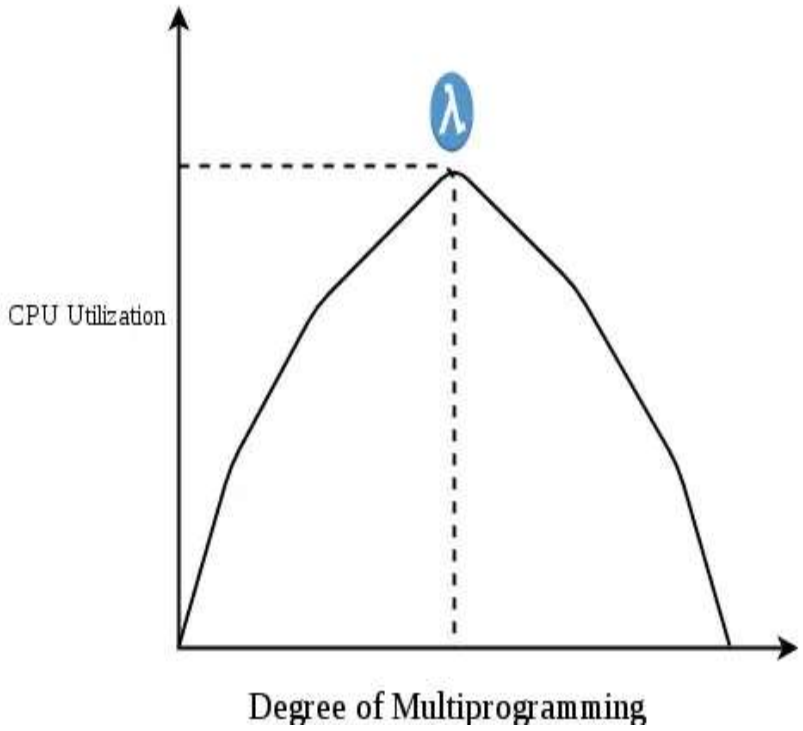
Virtual memory is a mechanism used to manage memory using hardware and software. It helps in running multiple applications with low main memory and increases the degree of multiprogramming in systems. It is commonly implemented using demand paging.

Virtual memory is a part of the system's secondary memory that acts and gives us an illusion of being the main memory. When your computer's physical memory is full, virtual memory is required.

Virtual memory in OS works mainly by transferring processes between the computer's RAM and hard disk depending on the requirements. A Memory Management unit (MMU) facilitates this purpose.

1.10 Thrashing

The initial degree of multiprogramming up to some extent of point(lambda), the CPU utilization is very high and the system resources are utilized 100%. But if we further increase the degree of multiprogramming the CPU utilization will drastically fall down and the system will spend more time only on the page replacement and the time taken to complete the execution of the process will increase. This situation in the system is called thrashing.



1.10 Thrashing

High Degree of Multiprogramming: If the number of processes keeps on increasing in the memory then the number of frames allocated to each process will be decreased. So, fewer frames will be available for each process. Due to this, a page fault will occur more frequently and more CPU time will be wasted in just swapping in and out of pages and the utilization will keep on decreasing.

Example: Let free frames = 400

Case 1: Number of processes = 100 Then, each process will get 4 frames.

Case 2: Number of processes = 400, Each process will get 1 frame. Case 2 is a condition of thrashing, as the number of processes is increased, frames per process are decreased. Hence CPU time will be consumed just by swapping pages.

1.10 Thrashing

High Degree of Multiprogramming: If the number of processes keeps on increasing in the memory then the number of frames allocated to each process will be decreased. So, fewer frames will be available for each process. Due to this, a page fault will occur more frequently and more CPU time will be wasted in just swapping in and out of pages and the utilization will keep on decreasing.

Example: Let free frames = 400

Case 1: Number of processes = 100 Then, each process will get 4 frames.

Case 2: Number of processes = 400, Each process will get 1 frame. Case 2 is a condition of thrashing, as the number of processes is increased, frames per process are decreased. Hence CPU time will be consumed just by swapping pages.

Thank You