Third Year Engineering

21BTIS501-Operating System

Class - T.Y. (SEM-I)

Unit - IV

Memory Management

AY 2023-2024 **SEM-I**

Unit-IV Syllabus

- Background of Memory Management, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation, Virtual Memory, Background, Demand Paging, Copy~on-Write, Page Replacement, Allocation of Frames, Thrashing.

Why Memory Management is Required?

- Allocate and de-allocate memory before and after process execution.
- To keep track of used memory space by processes.
- To minimize fragmentation issues.
- To proper utilization of main memory.
- To maintain data integrity while executing of process.

What is Memory Management in an Operating System?

- It is technique of controlling and managing the functionality of RAM.
- It is used for achieving better concurrency, system performance, and memory utilization.
- Used to moves processes from primary memory to secondary memory and vice versa.
- To keeps track of available memory, memory allocation, and unallocated.

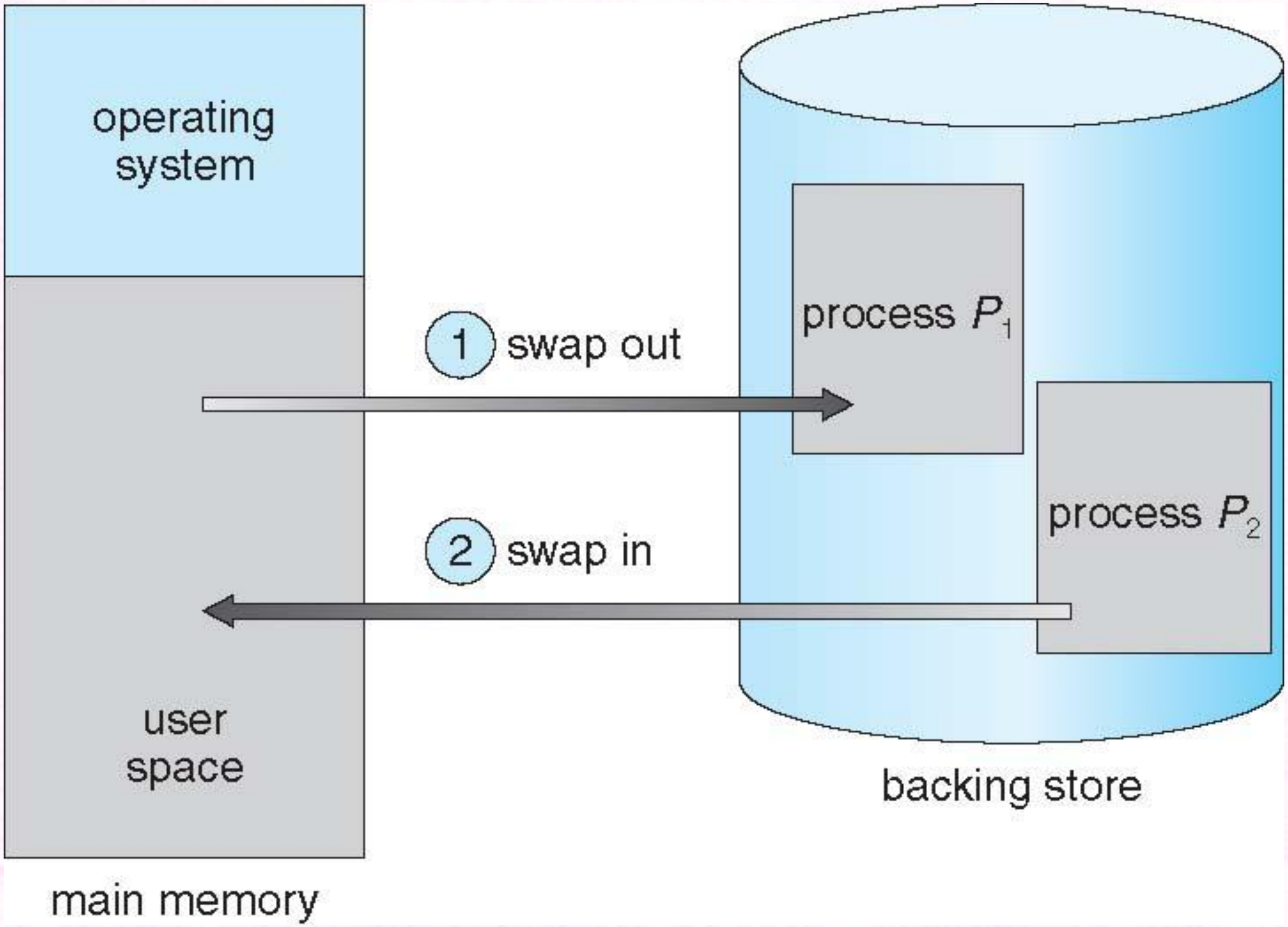
Swapping in an Operating System

- A process can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution.
  - Total physical memory space of processes can exceed physical memory
- **Backing store** – fast disk large enough to accommodate copies of all memory images for all users; must provide direct access to these memory images
- **Roll out, roll in** – swapping variant used for priority-based scheduling algorithms; lower-priority process is swapped out so higher-priority process can be loaded and executed.

Swapping in an Operating System

- Major part of swap time is transfer time; total transfer time is directly proportional to the amount of memory swapped.
- System maintains a ready queue of ready-to-run processes which have memory images on disk.
- Modified versions of swapping are found on many systems (i.e., UNIX, Linux, and Windows)
  - Swapping normally disabled
  - Started if more than threshold amount of memory allocated
  - Disabled again once memory demand reduced below threshold

Schematic View of Swapping



Advantages of Swapping

- Efficient Memory Utilization.
- Better System Performance.
- Reduced Process Blocking.
- Flexibility.

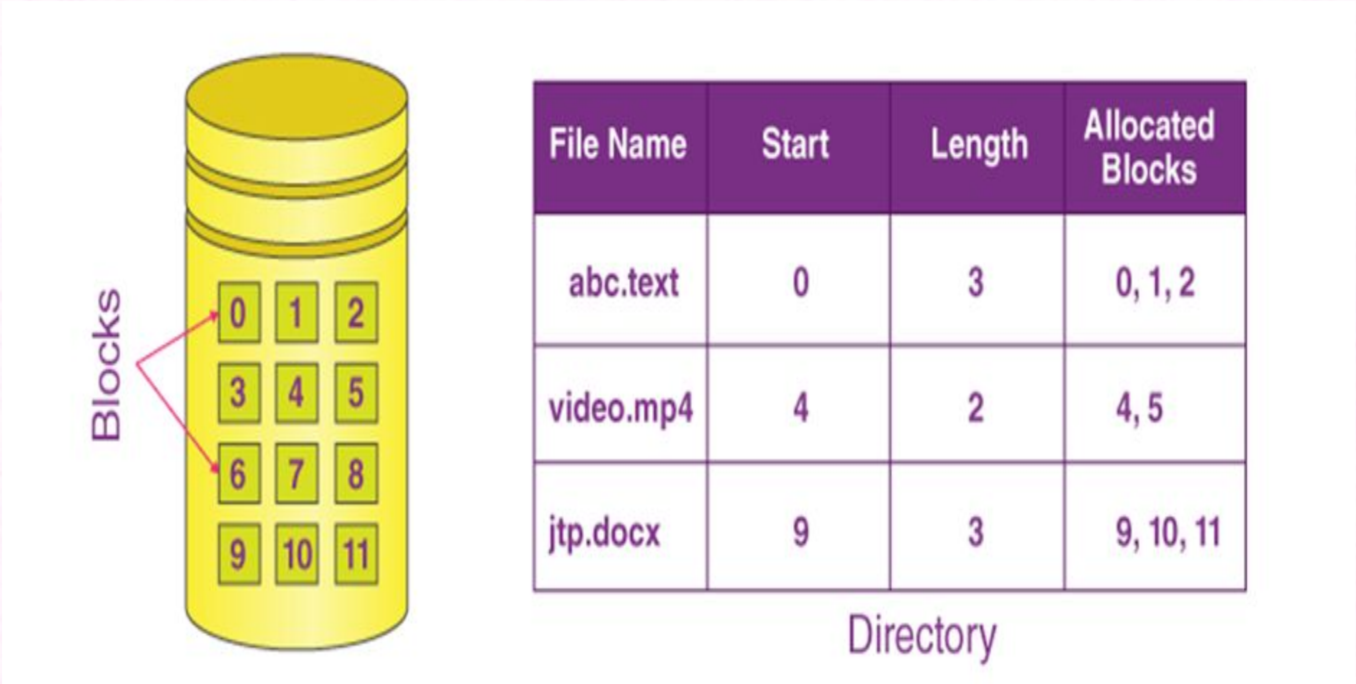
Disadvantages of Swapping

- Performance Overhead
- Storage Overhead
- Data Integrity Issues
- Increased Disk Activity

Contiguous Memory Allocation

- Here, all the processes are stored in contiguous memory locations
- All the available memory space remains together in one place. It means freely available memory partitions are not expanded here and there across the whole memory space.
- When any user process request for the memory a single section of the contiguous memory block is given to that process according to its need.

Contiguous Memory Allocation



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

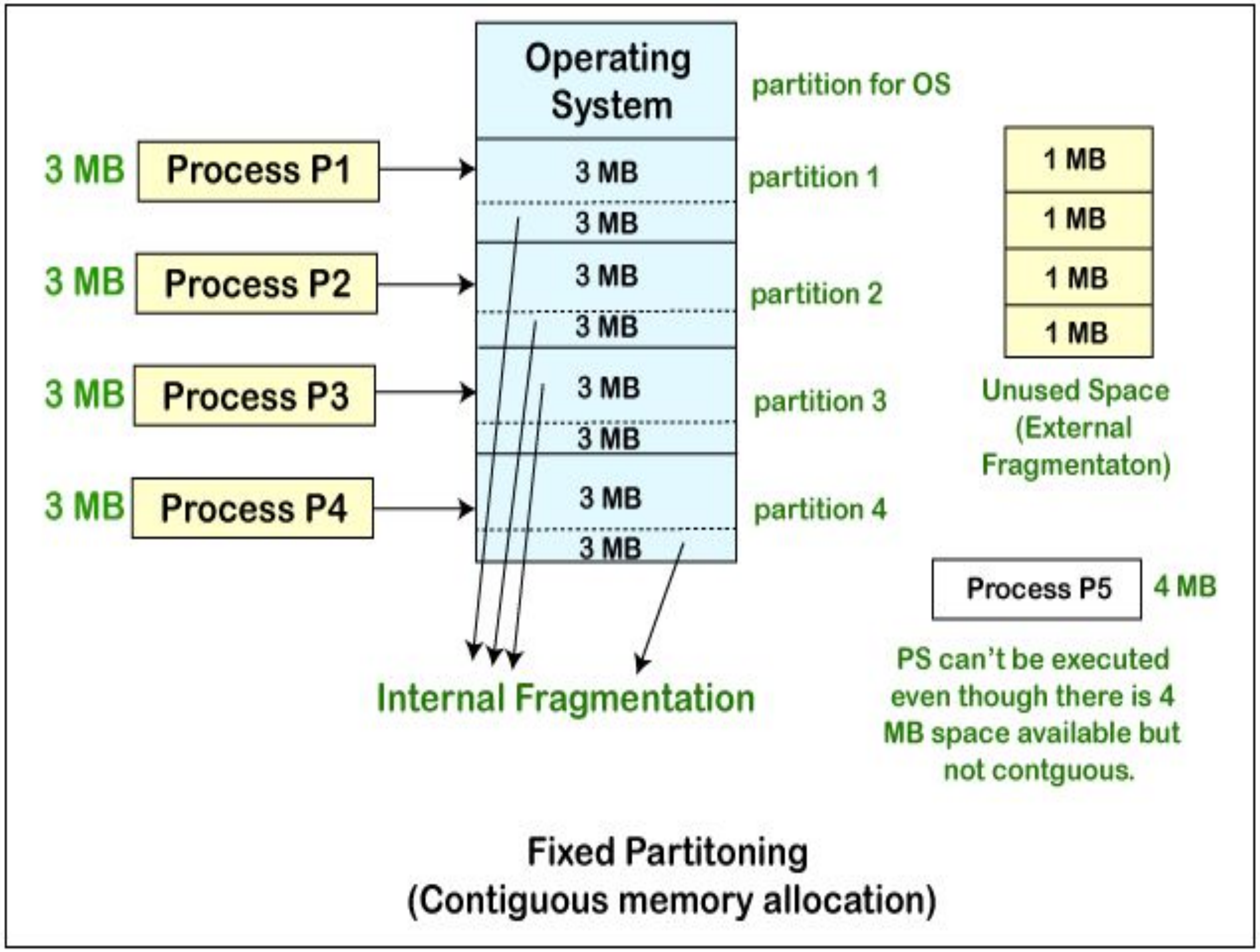
Contiguous Memory Allocation

- This allocation can be done in two ways:

1. Fixed-size Partition Scheme
2. Variable-size Partition Scheme

1. Fixed-size Partition Scheme

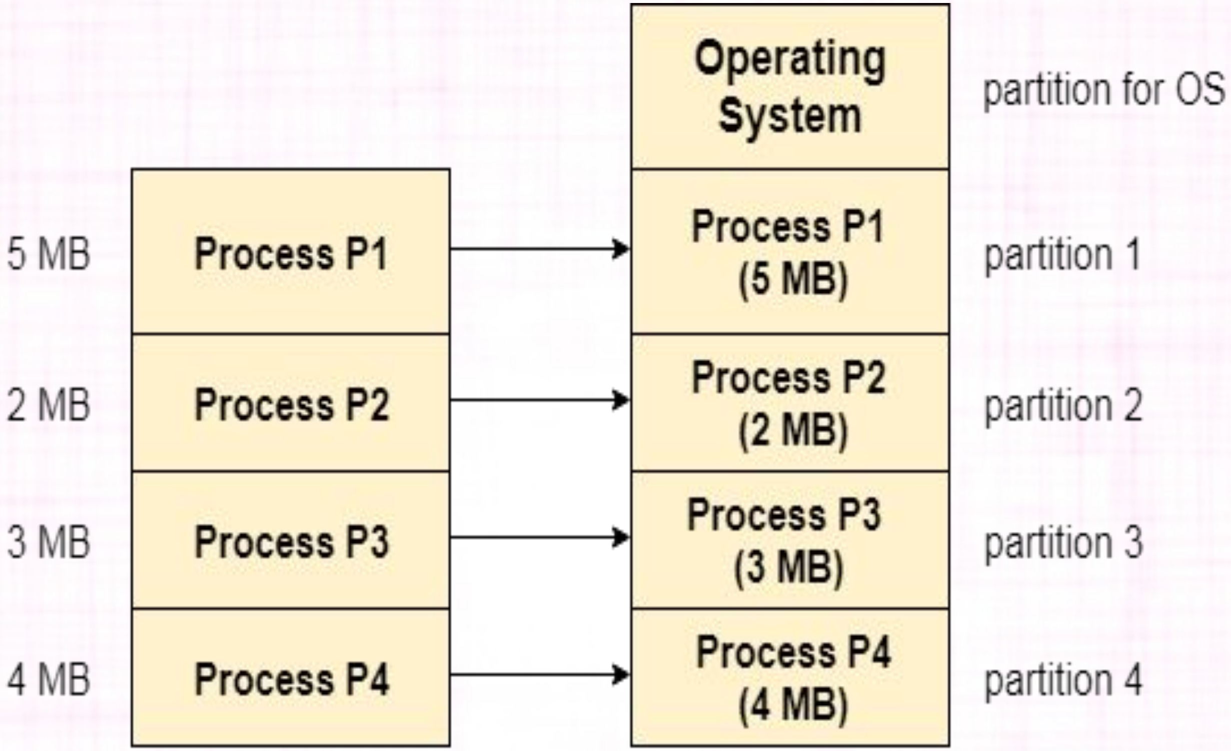
- In the fixed sized partition the system divides memory into fixed size partition (may or may not be of the same size)
- Another name for this is static partitioning.
- Disadvantage- Internal Fragmentation



Fixed Partitoning
(Contiguous memory allocation)

2. Variable-size Partition Scheme

- Dynamic partitioning is another name for this. The scheme allocation in this type of partition is done dynamically.
- Here, the size of every partition isn’t declared initially.
- Only once we know the process size, will we know the size of the partitions.
- But in this case, the size of the process and the partition is equal; thus, it helps in preventing internal fragmentation.



Dynamic Partitioning

(Process Size = Partition Size)

Memory Allocation

- Memory allocation is a process by which computer programs are assigned memory or space.
- Here, main memory is divided into two types of partitions

1. **Low Memory** – Operating system resides in this type of memory.
2. **High Memory**– User processes are held in high memory.

Partition Allocation

- Memory is divided into different blocks or partitions.
- Each process is allocated according to the requirement.
- Partition allocation is an ideal method to avoid internal fragmentation.

Different Placement Algorithm

- Below are the various placement algorithms schemes :

1. First Fit: In this type fit, the partition is allocated, which is the first sufficient block from the beginning of the main memory.
2. Best Fit: It allocates the process to the partition that is the first smallest partition among the free partitions.

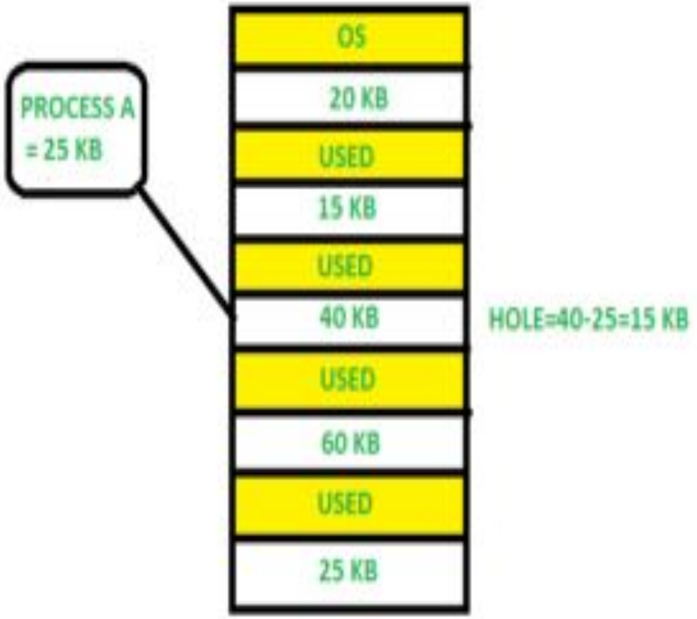
Partition Allocation

3. Worst Fit: It allocates the process to the partition, which is the largest sufficient freely available partition in the main memory.

4. Next Fit: It is mostly similar to the first Fit, but this Fit, searches for the first sufficient partition from the last allocation point.

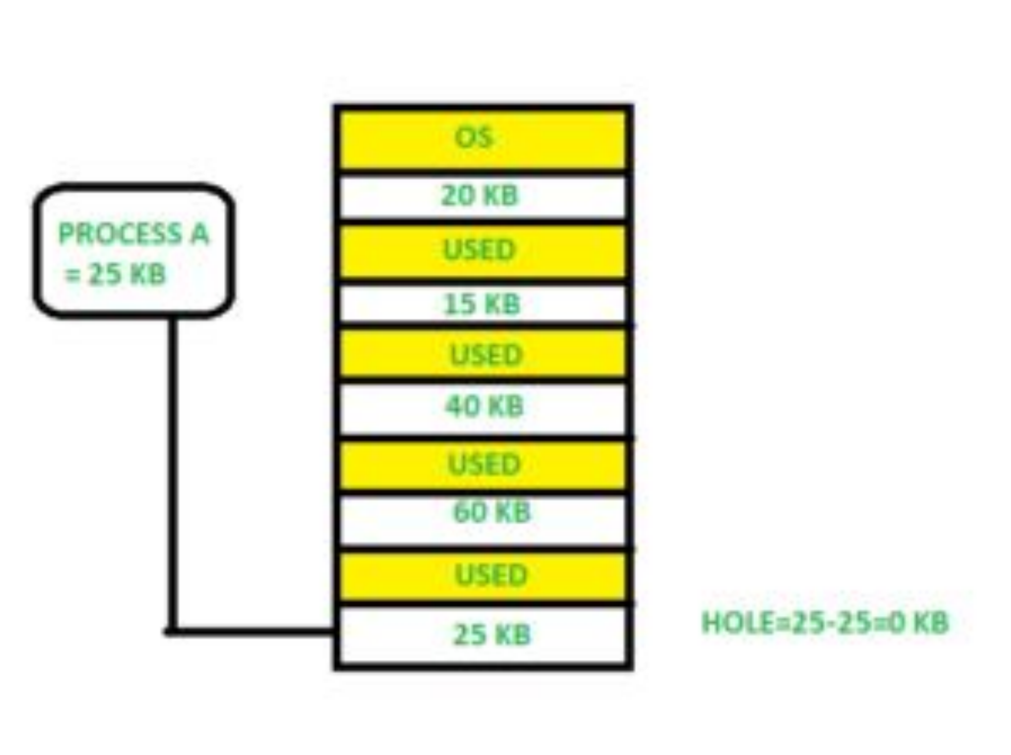
Partition Allocation

First Fit Diagram



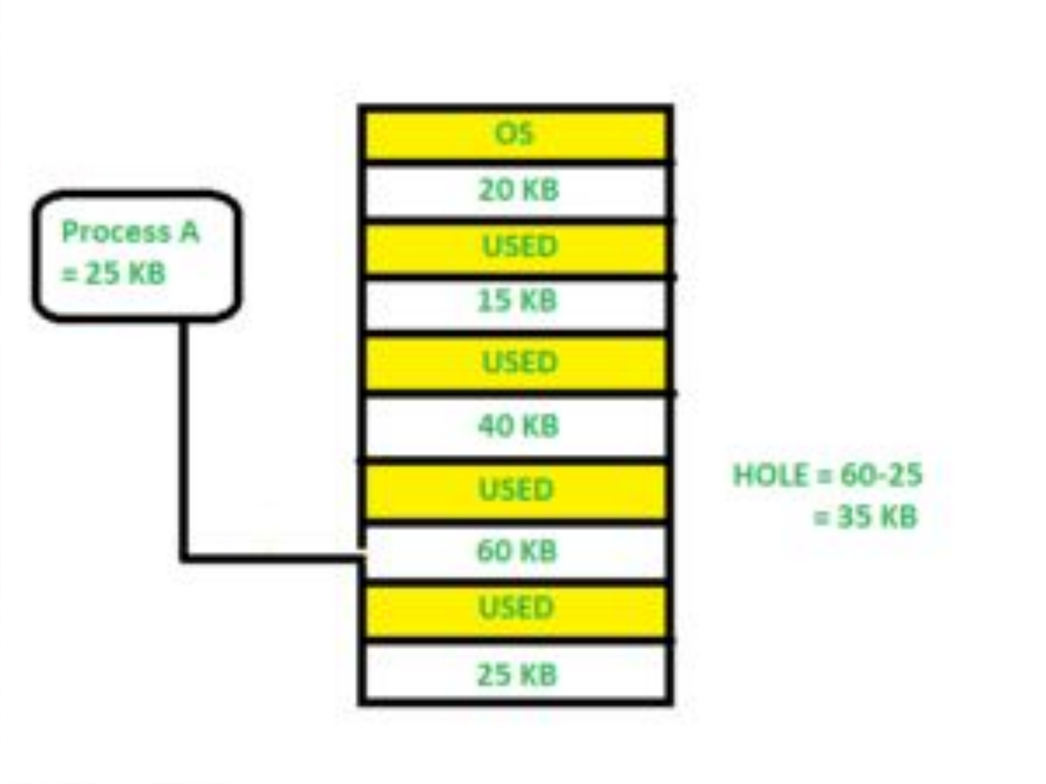
Partition Allocation

Best Fit Diagram



Partition Allocation

Worst Fit Diagram



Paging in Operating System

- Paging is a memory management scheme used by operating systems to manage and allocate physical memory (RAM) efficiently.
- The main idea behind the paging is to divide each process in the form of pages. The main memory will also be divided in the form of frames.
- **key concepts and components of paging in operating systems:**

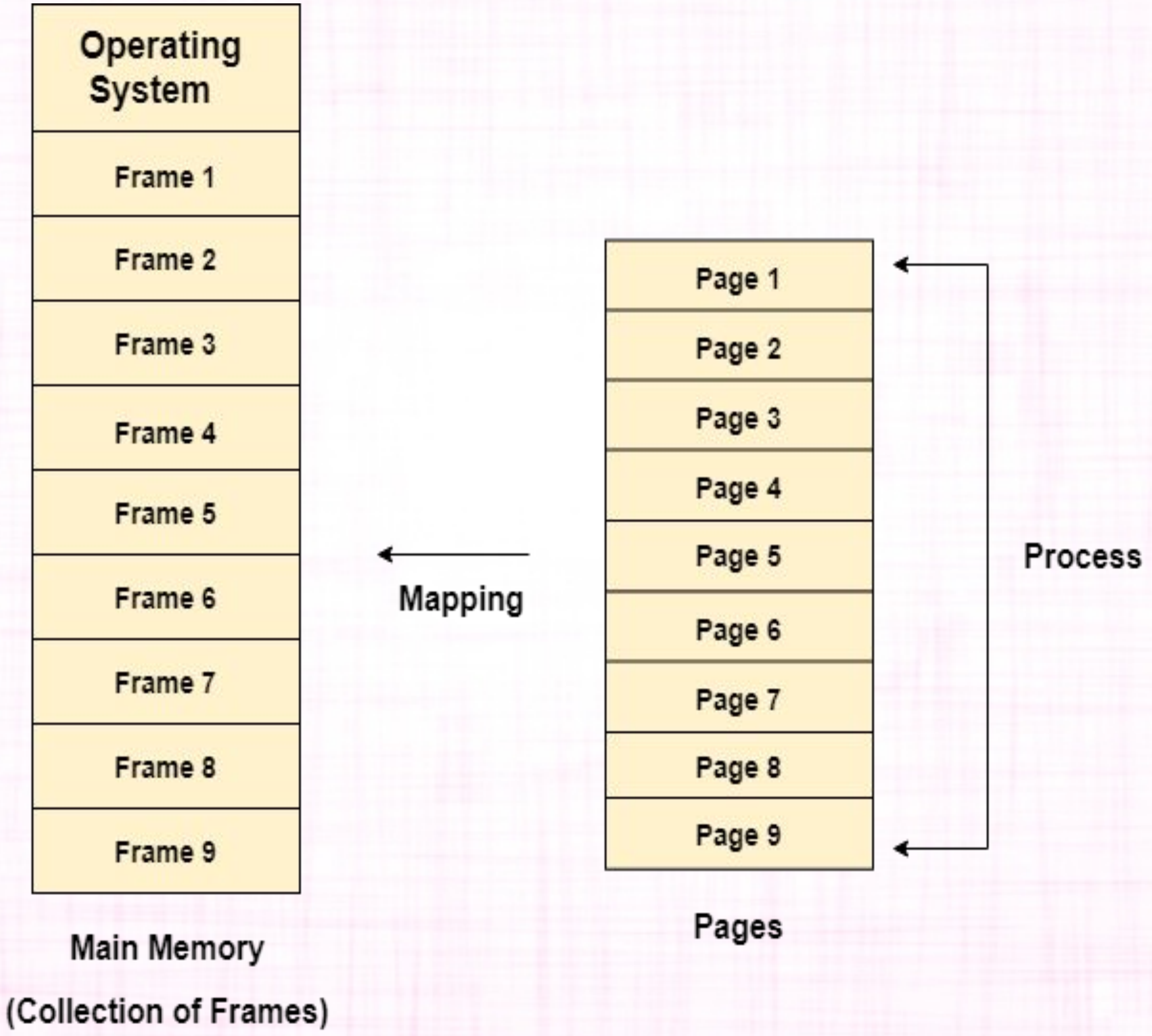
1. **Page:** A page is a fixed-size block of memory, typically 4 KB in size, used as the unit of data . Pages are also the unit of allocation in virtual memory.

2. Page Table: A page table is a data structure used by the operating system to map virtual addresses to physical addresses. Each process has its own page table, which allows the operating system to keep track of the mapping for each process.

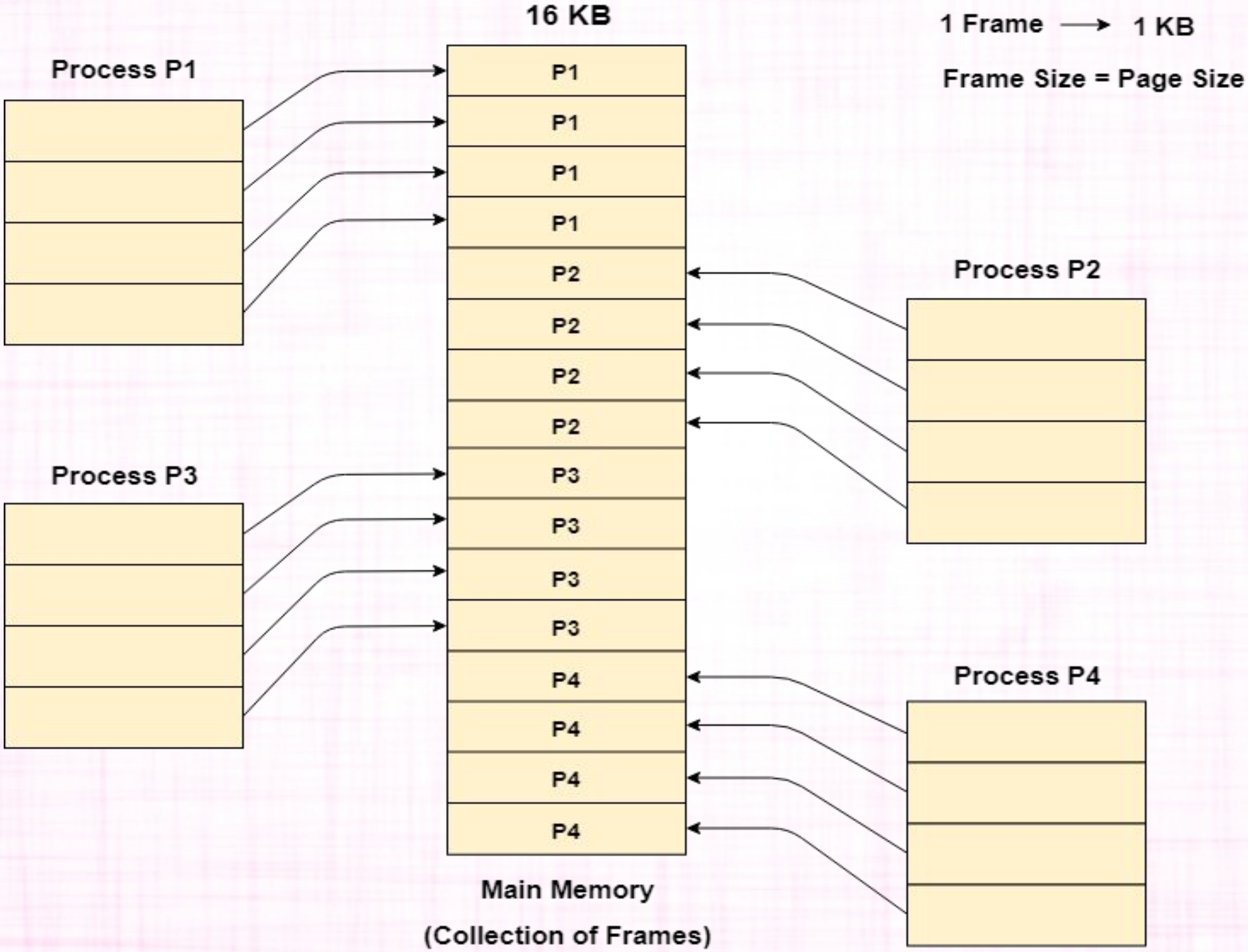
3. Page Frame: A page frame is a fixed-size block of physical memory (RAM) that corresponds to the size of a page. The physical memory is divided into page frames, and each page frame can hold one page of data.

4. Virtual Address: A virtual address is an address generated by a process when it accesses memory. It is the address that the process thinks it is using.

5. Physical Address: A physical address is the actual location in physical memory (RAM) where data is stored.


Operating System
Frame 1
Frame 2
Frame 3
Frame 4
Frame 5
Frame 6
Frame 7
Frame 8
Frame 9
Mapping
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
Page 8
Page 9
Process
Pages
Main Memory
(Collection of Frames)

- One page of the process is to be stored in one of the frames of the memory.
- Considering the fact that the pages are mapped to the frames in Paging, page size needs to be as same as frame size.
- Example:
- 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.
- Frames, pages and the mapping between the two is shown in the diagram below


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)

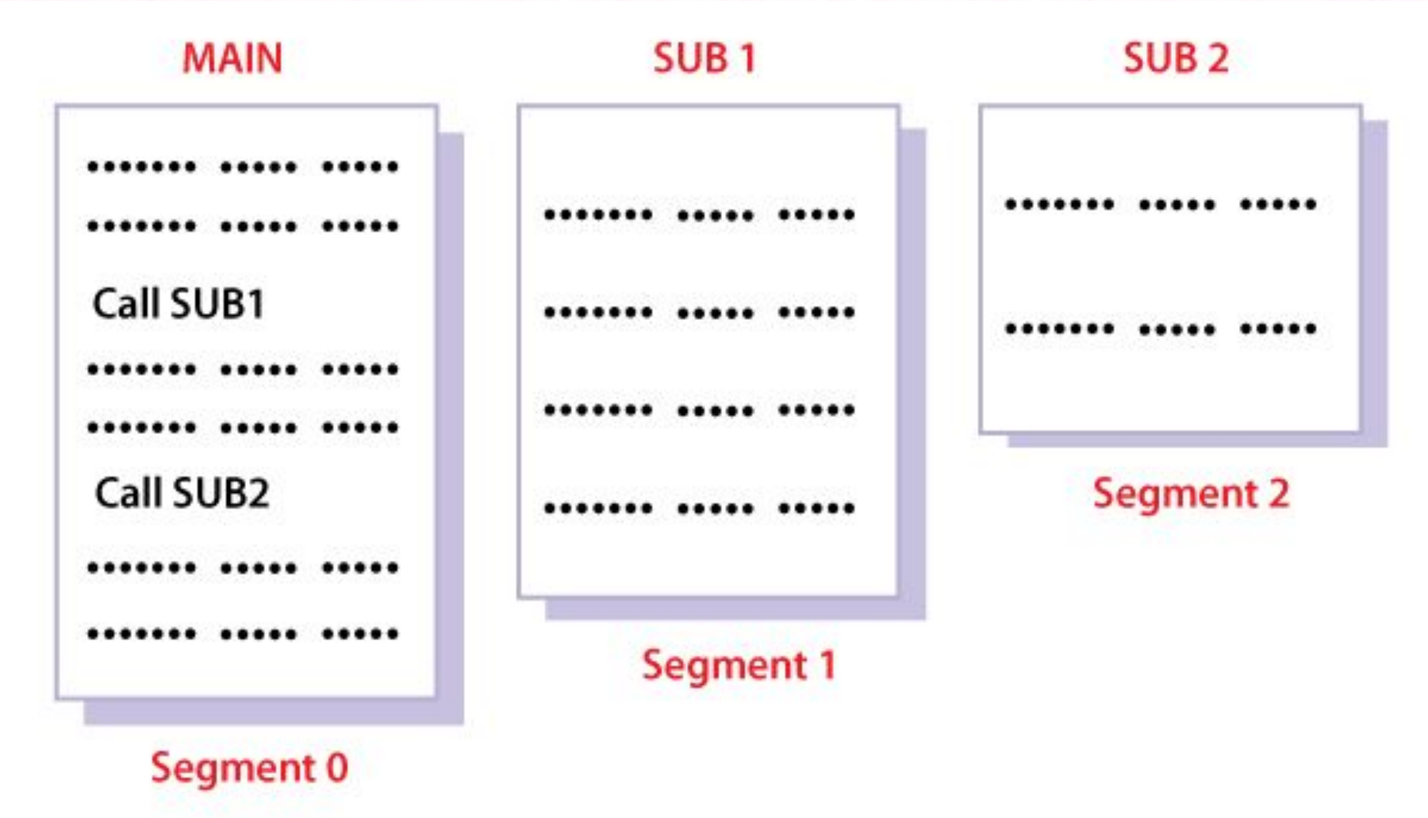
Paging

Segmentation in Operating System

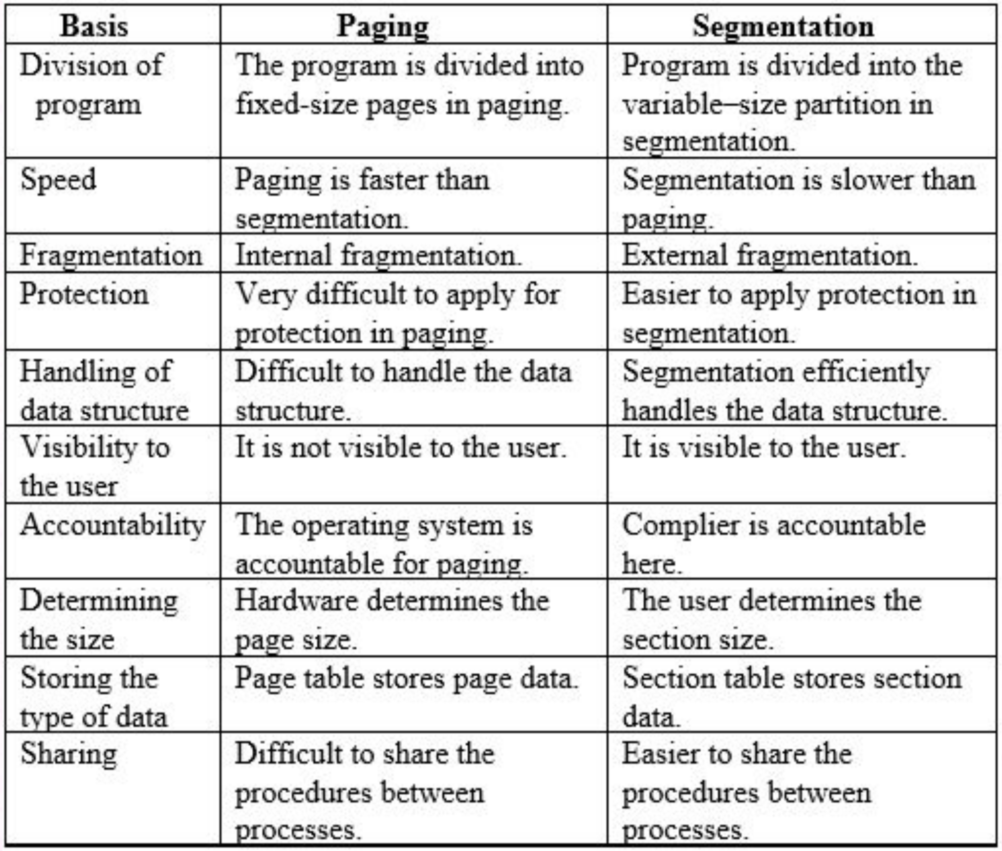
Segmentation divides processes into smaller subparts known as **modules**.

- The divided segments need not be placed in contiguous memory. Since there is no contiguous memory allocation, internal fragmentation does not take place.
- Paging divides all the processes into the form of pages regardless of the fact that a process can have some relative parts of functions which need to be loaded in the same page.
- 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.

Segmentation in Operating System



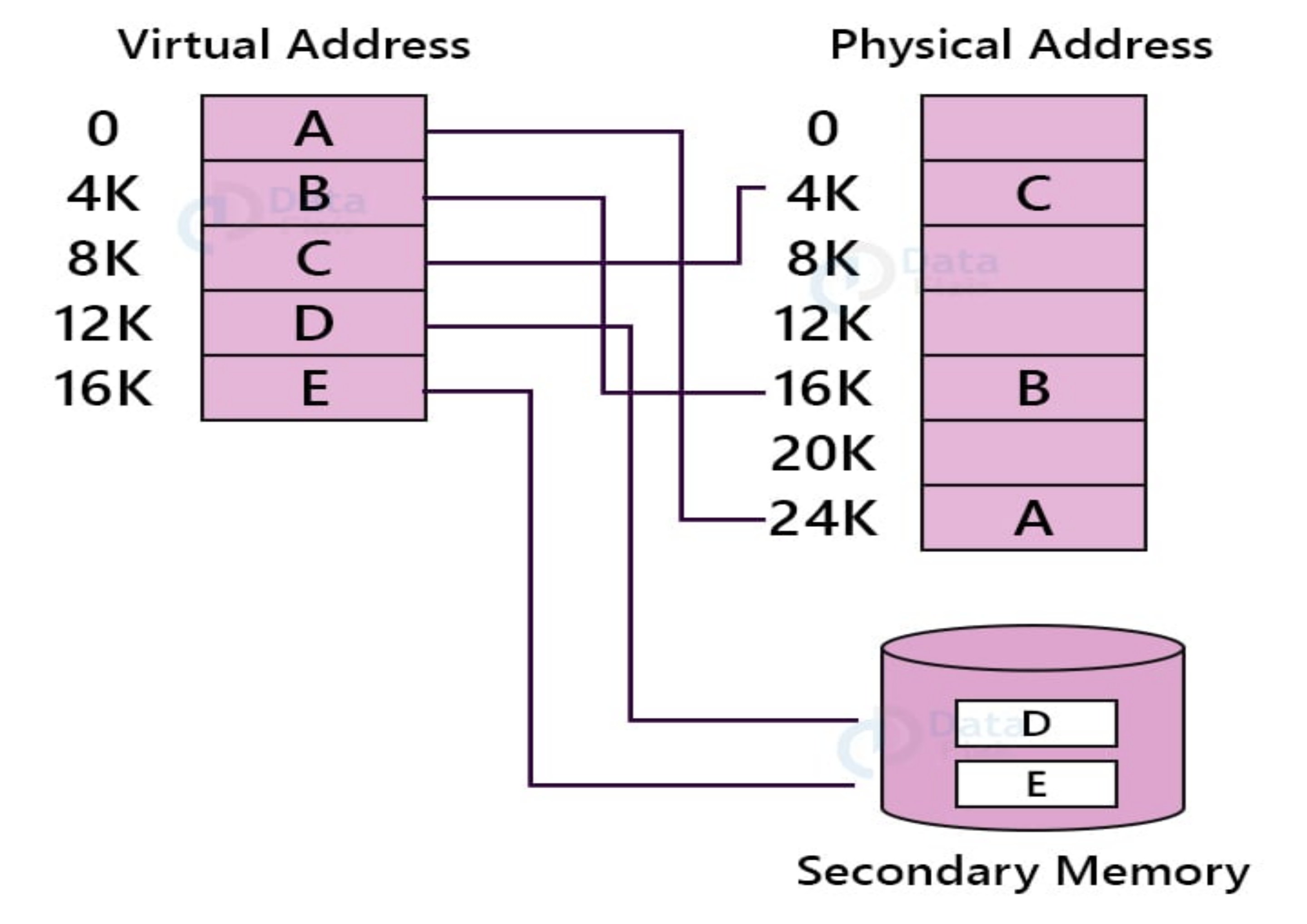
Difference between Paging and Segmentation



Basis	Paging	Segmentation
Division of program	The program is divided into fixed-size pages in paging.	Program is divided into the variable–size partition in segmentation.
Speed	Paging is faster than segmentation.	Segmentation is slower than paging.
Fragmentation	Internal fragmentation.	External fragmentation.
Protection	Very difficult to apply for protection in paging.	Easier to apply protection in segmentation.
Handling of data structure	Difficult to handle the data structure.	Segmentation efficiently handles the data structure.
Visibility to the user	It is not visible to the user.	It is visible to the user.
Accountability	The operating system is accountable for paging.	Complier is accountable here.
Determining the size	Hardware determines the page size.	The user determines the section size.
Storing the type of data	Page table stores page data.	Section table stores section data.
Sharing	Difficult to share the procedures between processes.	Easier to share the procedures between processes.

Virtual Memory

- Virtual Memory is a storage scheme that provides user an illusion of having a very big main memory.
- Here, Instead of loading one big process in the main memory, the Operating System loads the different parts of more than one process in the main memory.
- How Virtual Memory Works?
- Whenever some pages needs to be loaded in the main memory for the execution and the memory is not available for those many pages, then in that case, instead of stopping the pages from entering in the main memory, the OS search for the RAM area that are least used in the recent times or that are not referenced and copy that into the secondary memory to make the space for the new pages in the main memory.


Virtual Address
Physical Address
0
4K
8K
12K
16K
A
B
C
D
E
0
4K
8K
12K
16K
20K
24K
C
B
A
D
E
Secondary Memory

Advantages of Virtual Memory

- It can handle twice as many addresses as main memory.
- It enables more applications to be used at once.
- It has increased security because of memory isolation.
- It enables multiple larger applications to run simultaneously.
- Data can be moved automatically.

Demand Paging in Operating System

- Demand paging is a memory management scheme used in modern operating systems to optimize the use of physical memory (RAM).
- It's a variation of virtual memory management where pages of a process are loaded into memory only when they are needed, on-demand. In other words, it says that do not load any page in the main memory until it is required.
- 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

Advantages of Demand Paging in OS

1. Memory can be put to better use.
2. If we use demand paging, then we can have a large virtual memory.
3. By using demand paging, we can run programs that are larger than physical memory.
4. In demand paging, the sharing of pages is easy.
5. Partition management is simple in demand paging because of the fixed partition size and the discontinuous loading.

Disadvantages of Demand Paging in OS

1. Internal fragmentation is a possibility with demand paging.
2. It takes longer to access memory (page table lookup).
3. Memory requirements
4. Guarded page tables
5. Inverted page tables

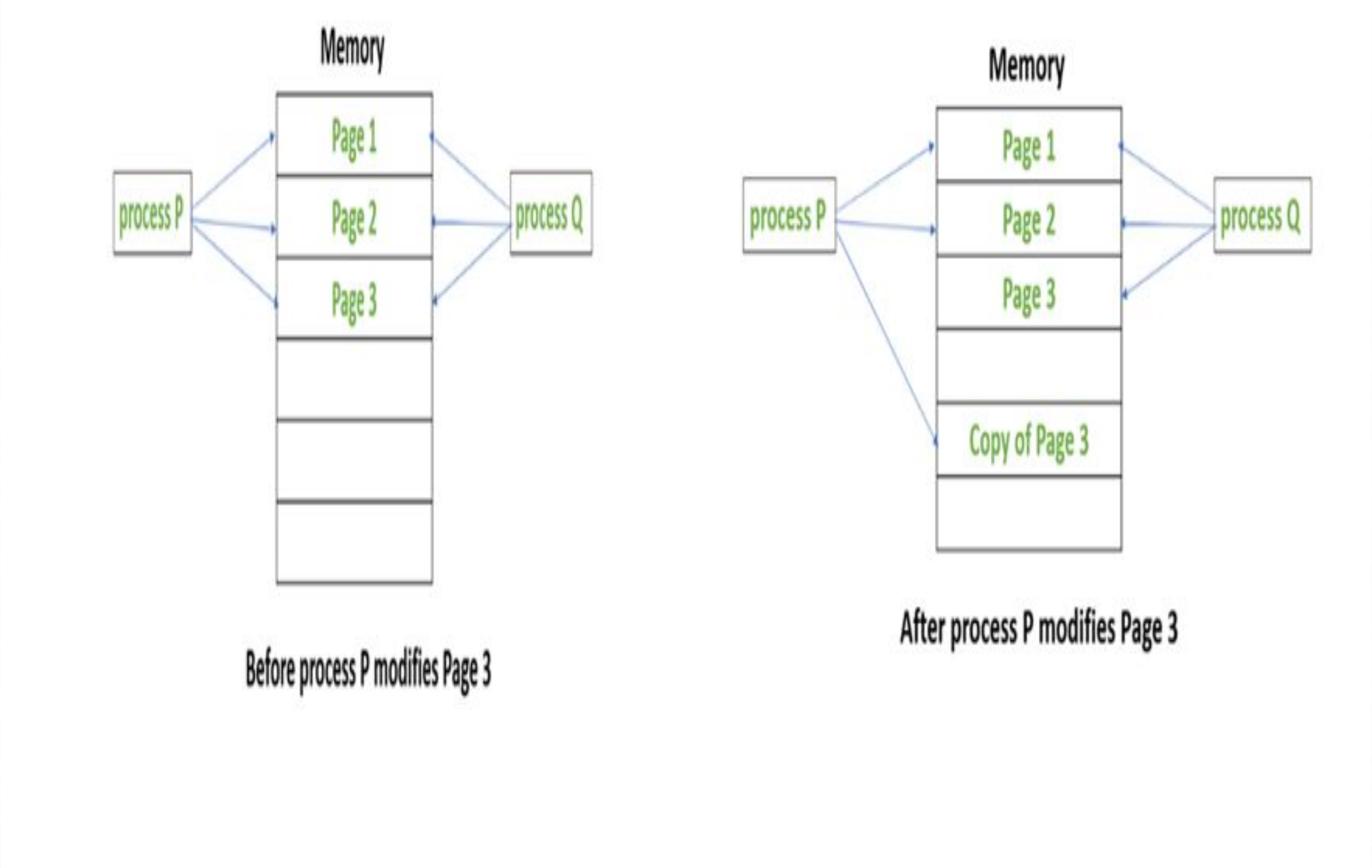
Copy on Write in Operating System

- Copy on Write or simply COW is a resource management technique. One of its main use is in the implementation of the fork system call in which it shares the virtual memory(pages) of the OS.
- In UNIX like OS, fork() system call creates a duplicate process of the parent process which is called as the child process.

Copy on Write in Operating System

- Copy on Write or simply COW is a resource management technique. One of its main use is in the implementation of the fork system call in which it shares the virtual memory(pages) of the OS.
- In UNIX like OS, fork() system call creates a duplicate process of the parent process which is called as the child process.

Copy on Write in Operating System



Page Replacement in Operating System

- **Page replacement algorithms** aim to optimize memory utilization, minimize page faults (when a requested page is not in memory), and maintain a balance between keeping frequently accessed pages in RAM and making room for new pages.
- A page replacement algorithm is needed to decide which page needs to be replaced when a new page comes in.
- **Page Fault:** A page fault happens when a running program accesses a memory page that is mapped into the virtual address space but not loaded in physical memory. Since actual physical memory is much smaller than virtual memory, page faults happen.
- In case of a page fault, Operating System might have to replace one of the existing pages with the newly needed page.

Page Replacement Algorithms

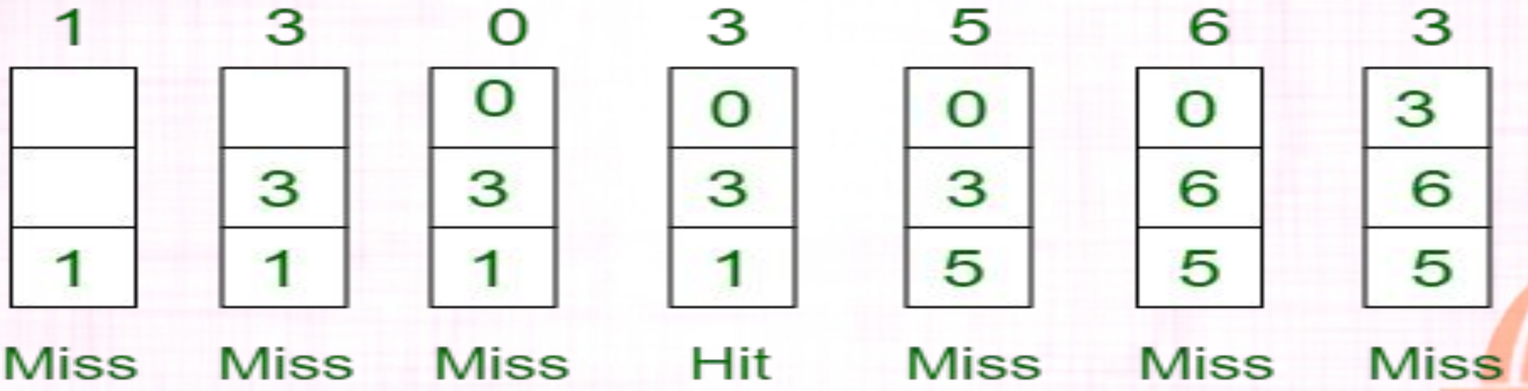
- Page Replacement Algorithms in Operating Systems

1. First In First Out (FIFO)
2. Least Recently Used (LRU)
3. Optimal Page Replacement

1. First In First Out(FIFO)

- In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue.
- When a page needs to be replaced page in the front of the queue is selected for removal.
- **Example 1:** Consider page reference string 1, 3, 0, 3, 5, 6, 3 with 3 page frames. Find the number of page faults.
- Initially, all slots are empty, so when 1, 3, 0 came they are allocated to the empty slots —> 3 Page Faults.
- when 3 comes, it is already in memory so —> 0 Page Faults.
- Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e 1. —>1 Page Fault.
- 6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 —>1 Page Fault. Finally, when 3 come it is not available so it replaces 0 1 page fault.

Page reference 1, 3, 0, 3, 5, 6, 3



1	3	0	3	5	6	3
		0	0	0	0	3
	3	3	3	3	6	6
1	1	1	1	5	5	5
Miss	Miss	Miss	Hit	Miss	Miss	Miss

Total Page Fault = 6

First-In-First-Out (FIFO) Algorithm

Advantages

- Simple and easy to implement.
- Low overhead.

Disadvantages

- Poor performance.
- Doesn’t consider the frequency of use or last used time, simply replaces the oldest page.
- Suffers from Belady’s Anomaly(i.e. more page faults when we increase the number of page frames).

Least Recently Used (LRU)Algorithm

- In this algorithm, page will be replaced which is least recently used.
- The LRU algorithm replaces the page that has not been accessed for the longest time.
- LRU can provide good performance in certain cases but can be challenging to implement efficiently, especially in systems with a large number of pages.
- **Example-3:** Consider the page reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3 with 4 page frames. Find number of page faults.

Page reference 7,0,1,2,0,3,0,4,2,3,0,3,2,3

No. of Page frame - 4

7	0	1	2	0	3	0	4	2	3	0	3	2	3
			2	2	2	2	2	2	2	2	2	2	2
		1	1	1	1	1	4	4	4	4	4	4	4
	0	0	0	0	0	0	0	0	0	0	0	0	0
7	7	7	7	7	3	3	3	3	3	3	3	3	3
Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit	Hit	Hit	Hit

Total Page Fault = 6

Here LRU has same number of page fault as optimal but it may differ according to question.

- Initially, all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> **4 Page faults**.
  0 is already their so —> **0 Page fault.**
- When 3 came it will take the place of 7 because it is least recently used —>**1 Page fault.**
  0 is already in memory so —> **0 Page fault**.
  4 will takes place of 1 —> **1 Page Fault**
  Now for the further page reference string —> **0 Page fault** because they are already available in the memory.

Least Recently Used (LRU)Algorit hm

- Advantages
  - – Efficient.
  - – Doesn't suffer from Belady’s Anomaly.
- Disadvantages
  - – Complex Implementation.
  - – Expensive.
  - – Requires hardware support.

Optimal Page Replacement

- In this algorithm, pages are replaced which would not be used for the longest duration of time in the future.It is also known as OPT, clairvoyant replacement algorithm, or Belady's optimal page replacement policy.

Example-2: Consider the page references 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3 with 4 page frame. Find number of page fault.

Page reference 7,0,1,2,0,3,0,4,2,3,0,3,2,3 No. of Page frame - 4

7	0	1	2	0	3	0	4	2	3	0	3	2	3
			2	2	2	2	2	2	2	2	2	2	2
		1	1	1	1	1	4	4	4	4	4	4	4
	0	0	0	0	0	0	0	0	0	0	0	0	0
7	7	7	7	7	3	3	3	3	3	3	3	3	3
Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit	Hit	Hit	Hit

Total Page Fault = 6

Optimal Page Replacement

Initially, all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> **4 Page faults**
0 is already there so —> **0 Page fault.**
when 3 came it will take the place of 7 because it is not used for the longest duration of time in the future.—>**1 Page fault.**
0 is already there so —> **0 Page fault.**
4 will takes place of 1 —> **1 Page Fault.**
Now for the further page reference string —> **0 Page fault** because they are already available in the memory.
Optimal page replacement is perfect, but not possible in practice as the operating system cannot know future requests

Optimal Page Replacement

- Advantages
  - Easy to Implement.
  - Simple data structures are used.
  - Highly efficient.
- Disadvantages
  - Requires future knowledge of the program.
  - Time-consuming.

Allocation of frames in OS

- The main memory of the system is divided into frames.
- The OS has to allocate a sufficient number of frames for each process and to do so, the OS uses various algorithms.
- The five major ways to allocate frames are as follows:

1. Proportional frame allocation
2. Priority frame allocation
3. Global replacement allocation
4. Local replacement allocation
5. Equal frame allocation

Allocation of frames in OS

- The proportional frame allocation algorithm allocates frames based on the size that is necessary for the execution and the number of total frames the memory has.
- The only disadvantage of this algorithm is it does not allocate frames based on priority. This situation is solved by Priority frame allocation.

1. Proportional frame allocation

- The proportional frame allocation algorithm allocates frames based on the size that is necessary for the execution and the number of total frames the memory has.
- The only disadvantage of this algorithm is it does not allocate frames based on priority. This situation is solved by Priority frame allocation.

2. Priority frame allocation

- Priority frame allocation allocates frames based on the priority of the processes and the number of frame allocations.
- If a process is of high priority and needs more frames then the process will be allocated that many frames. The allocation of lower priority processes occurs after it.

3. Global replacement allocation

- When there is a page fault in the operating system, then the global replacement allocation takes care of it.
- The process with lower priority can give frames to the process with higher priority to avoid page faults.

4. Local replacement allocation

- In local replacement allocation, the frames of pages can be stored on the same page.
- It doesn’t influence the behavior of the process as it did in global replacement allocation.

5. Equal frame allocation

- In equal frame allocation, the processes are allocated equally among the processes in the operating system.
- The only disadvantage in equal frame allocation is that a process requires more frames for allocation for execution and there are only a set number of frames.

What is Fragmentation?

- Processes are stored and removed from memory, which creates free memory space, which are too small to use by other processes.
- After sometimes, that processes not able to allocate to memory blocks because its small size and memory blocks always remain unused is called fragmentation.
- This type of problem happens during a dynamic memory allocation system when free blocks are quite small, so it is not able to fulfill any request.

What is Fragmentation?

- Two types of Fragmentation methods are:

1. External fragmentation

External fragmentation can be reduced by rearranging memory contents to place all free memory together in a single block.

2. Internal fragmentation

The internal fragmentation can be reduced by assigning the smallest partition, which is still good enough to carry the entire process.

Thrashing in OS (Operating System)

- Thrash is a term used in computer science to describe the poor performance of a virtual memory system when the same pages are loaded repeatedly owing to a shortage of main memory to store them in secondary memory.
- Thrashing happens in computer science when a computer's virtual memory resources are overutilized, resulting in a persistent state of paging and page faults, which inhibits most application-level activity.
- It causes the computer's performance to decline or collapse. The scenario can last indefinitely unless the user stops certain running apps or active processes to free up extra virtual memory resources.

Algorithms during Thrashing

1. Global Page Replacement
2. Local Page Replacement

Algorithms during Thrashing

1. Global Page Replacement
   - Since global page replacement can bring any page, it tries to bring more pages whenever thrashing is found.
   - But what actually will happen is that no process gets enough frames, and as a result, the thrashing will increase more and more.
   - Therefore, the global page replacement algorithm is not suitable when thrashing happens.

Algorithms during Thrashing

2. Local Page Replacement

- Unlike the global page replacement algorithm, local page replacement will select pages which only belong to that process.
- So there is a chance to reduce the thrashing. But it is proven that there are many disadvantages if we use local page replacement.
- Therefore, local page replacement is just an alternative to global page replacement in a thrashing scenario.