Date: / / Page no:

UNIT - III

* Concept of Memory Management -

→ The task of subdividing the memory among different processes is called Memory Management.
→ Memory Management is a method in the OS to manage operations betw main memory & disk during process execution.
→ Main aim of memory management is utilizatn of memory.

* Why Memory Management is required?

(1) Proper utilizatn of main memory.
(2) To minimize fragmentation issues.
(3) To keep track of used memory space by processes.
(4) To maintain data integrity while executing of process.
(5) Allocate & deallocate memory before and after process execution.

* Logical & Physical Address Space -

→ **Logical Address Space** - An address generated by CPU is called logical address.
* Also known as Virtual Address.
* It can be defined as the size of process.
* A logical address can be changed.

Date: / / Page no:

→ Physical Address Space : An address seen by the memory unit is known as physical address.

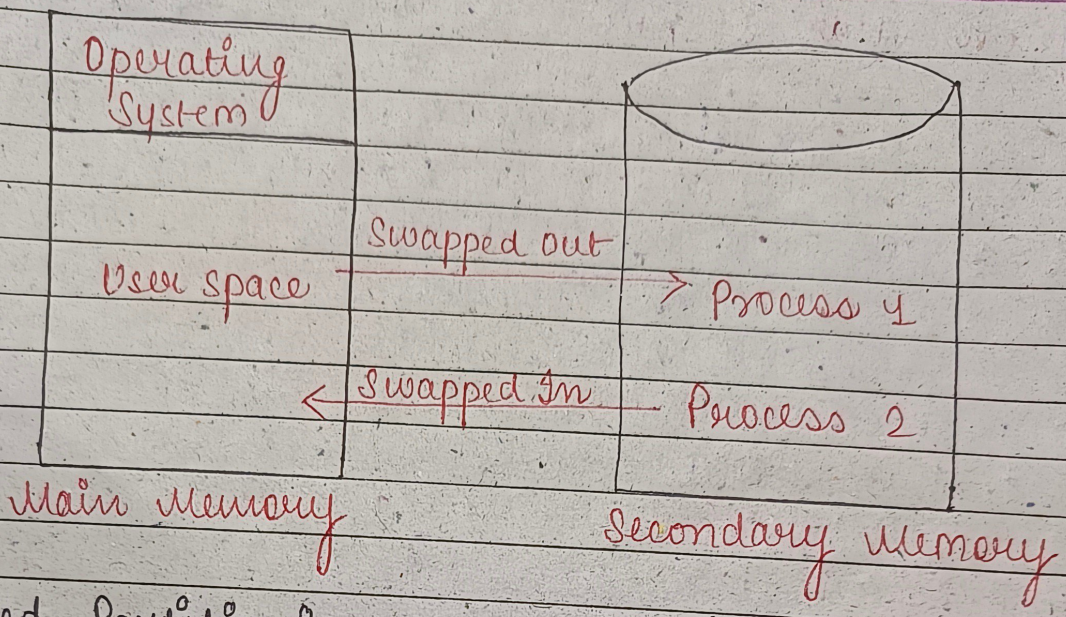
* Also known as Real Address.
* The set of all physical addresses corresponding to these logical addresses is called as Physical Address Space.
* It is computed by Memory Management Unit (MMU)
* The physical address always remain constant.

* Swapping -

Swapping is a process of swapping a process temporarily into a secondary memory from the main memory which is fast as compared to secondary memory.

* Swapping allows more processes to be run & can be fit into memory at a time.
* The transferred time & total time is directly proportional to the amount of memory swapped.
* Swapping is also known as roll-out or roll because if a higher priority process arrives & wants service, the memory manager can swap out the lower priority process & then load & execute the higher priority process.
* After finishing higher priority work, the lower priority work is swapped back in memory & continued to the execution process.

Date: / / Page no:



* Fixed Partitioning -

→ In this, the memory is divided into fixed size partitions or blocks during system initialization.
→ Each partition can accomodate exactly one process.
→ When a process needs memory, it is allocated one of the available partitions based on its size. If there is no partition large enough to accomodate the process, it waits until one becomes available.
→ Fixed Partitioning is relatively simple to implement but may lead to inefficient memory utilization, especially if the processes vary widely in size.
→ The inefficiency in size or memory occurs because small processes may occupy large partitions, leaving unused space within those partitions.
→ It suffers from fragmentation both internal (unused memory within a partition) & external (unused memory scattered between partitions)

Date: / / Page no:

* Dynamic Partitioning -

→ It allows for the allocation of variable sized memory blocks to processes, based on their size requirements.

→ When a process requests memory, the system searches for a suitable block of memory in the available free space. If a block is found that is large enough to accomodate the process, it is allocated to the process.

→ Dynamic Partitioning doesn't define the partitn sizes, allowing for better memory utilizatn, as memory is divided more flexibly.

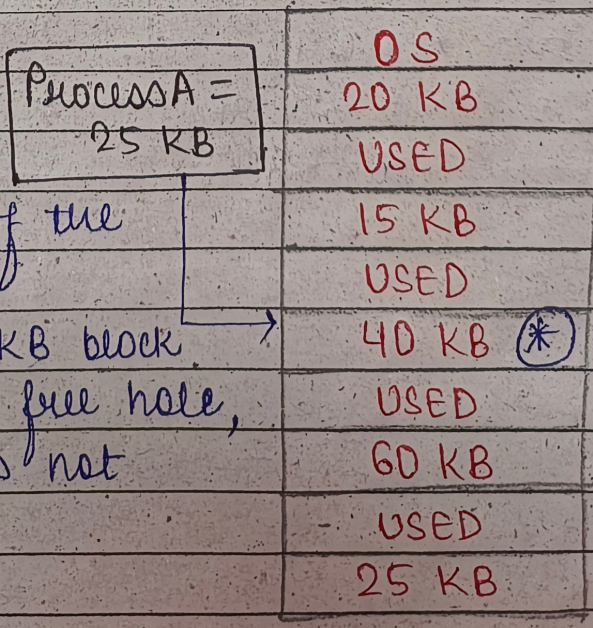
→ It introduces overhead in terms of managing and tracking free memory blocks.

→ It also suffers from fragmentatn, especially external fragmentatn, where free memory blocks become scattered throughout memory, making it challenging to allocate contiguous blocks of memory to processes.

* First Fit Allocation -

In first-fit, the first available free hole fulfil the requirement of the process allocated.

* In this diagram, a 40 KB block is the first available free hole, as the first two was not sufficient.

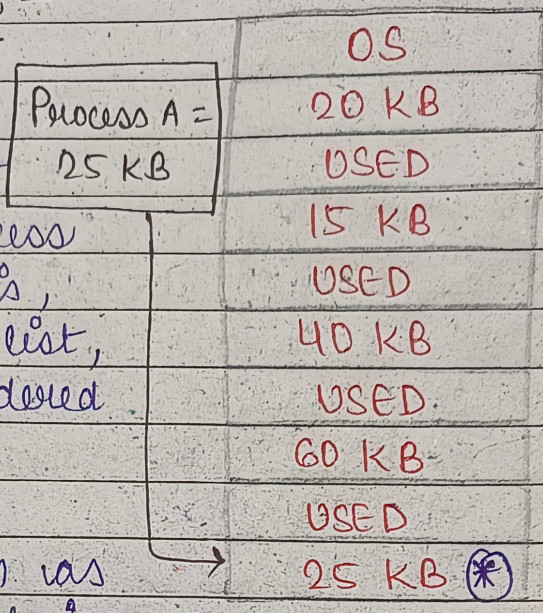


Date: / / Page no:

* Best Fit Allocation -

In best fit allocation, the smallest hole that is big enough to process requirements. For this, we search the entire list, unless the list is ordered by size.

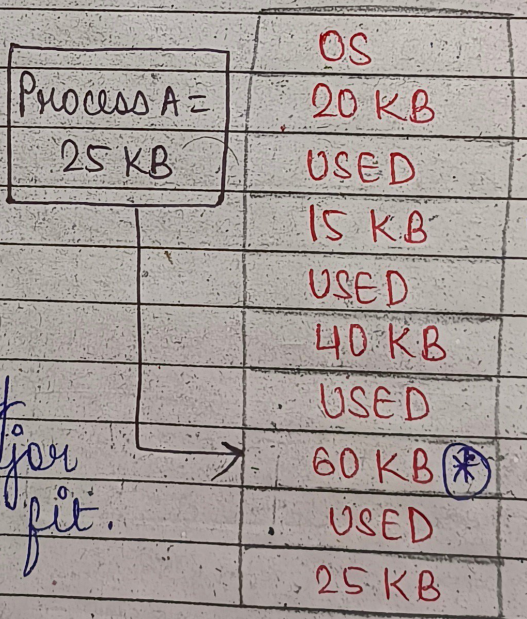
* In this, memory utilizatn is maximum as compared to other techniques.



* Worst Fit Allocation -

In worst fit, it allocates the largest available hole to process. This process produces the largest leftover holes.

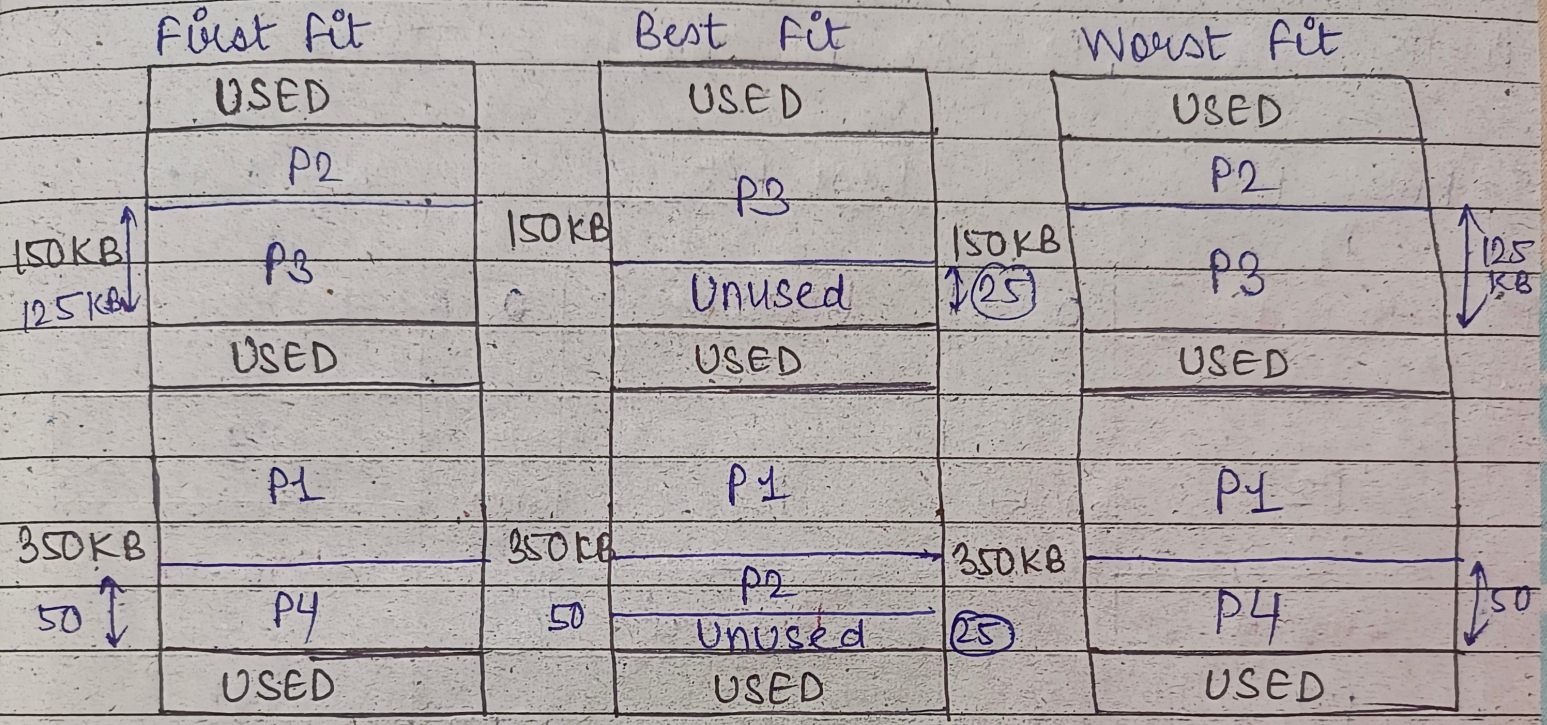
* Inefficient memory utilization is a major issue in the worst fit.



Date: / / Page no:

* Paging

Ques Requests from processes are 300K, 25K, 125K, 50K (P_1, P_2, P_3, P_4) respectively, the above request could satisfy with-



(P4) cannot be allocated.

Date: / / Page no:

Ques. Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, 125 KB in order, how would the first fit, best fit, worst fit algorithms place processes size 115 KB, 500 KB, 358 KB, 200 KB, 375 KB in order in fixed size partitioning.

P1 = 115, P2 = 500, P3 = 358, P4 = 200, P5 = 375 KB

First Fit

P1		
(185) Unused	300 KB	
P2	600 KB	
(100) Unused		
P4	350 KB	
(150) Unused		
Unused	200 KB	
P3	750 KB	
392 P5		
Unused	(17)	(10)
Unused	125 KB	

Best Fit

Unused	300 KB	
P2	600 KB	
Unused	(100)	
Unused	350 KB	
P4	200 KB	
P3	750 KB	
P5	(392)	
Unused	125 KB	
P1	(10)	
Unused		

Worst Fit

Unused	300 KB	
P3	600 KB	
Unused	(242)	
P4	350 KB	
Unused	(150)	
Unused	200 KB	
P1	750 KB	
P2	(635)	
Unused	(135)	
Unused	125 KB	
	(125)	

(P5) cannot be allocated

Date: / / Page no:

* Paging -

→ It is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages.

→ Main idea is to divide each process in the form of pages.
Main memory is divided in form of frames.

→ One page of the process is stored in one of the frames of memory.

→ Pages of process are brought in main memory only when needed, otherwise stored in secondary memory.

→ Each frame must be equal &
Page size = Frame size

* Memory Management Unit -

→ Main Purpose: Convert logical address → Physical Address.
→ Logical address is generated by CPU for every page & physical address is the actual address of the frame where each page will be stored.
→ The logical address has 2 parts:

1. Page Number
2. Offset

MMU converts page number to the frame number.

Date: / / Page no:

* Segmentation –

→ It is a memory management technique in which memory is divided into variable size parts.
→ Each part is known as Segment & can be allocated to a process.
→ details about each segment is stored in Segment Table.
→ Segment Table contains mainly two information about segment :

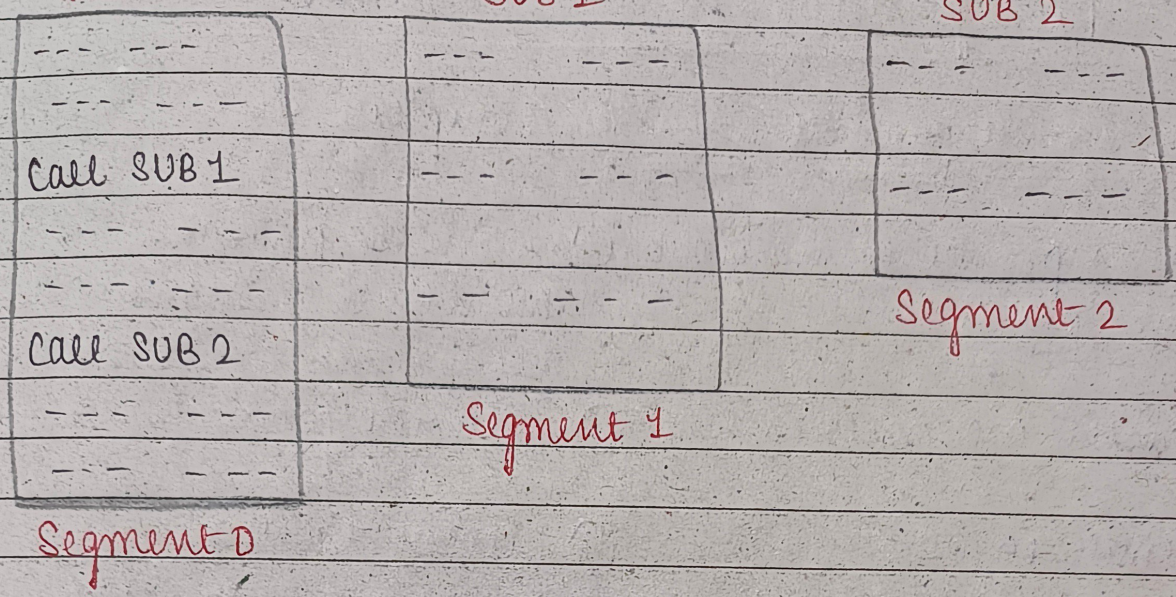
1. Base : base address of segment.
2. Limit : length of the segment.

* Need of Segmentation –

Paging technique is more close to the OS than the User. Paging divides 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 into the same page.
The OS doesn't care about User's view of process. It may divide the same function into diff. pages & those pages may or may not be loaded at the same time into the memory. It decreases the efficiency of system.

∴ Segmentation is required. Each segment contains the same type of functions such as the main function can be included in one segment & the library functions can be included in the other segment.

Date: / / Page no:



* Advantages of Segmentation -

1. No internal fragmentation
2. Less overhead
3. Avg Segment size is larger than Actual Page size.
4. It is easier to relocate segments than entire address space.
5. The segment table is of lesser space as compared to the page table in paging.

* Disadvantages of Segmentation -

1. It can have external fragmentation.
2. Costly Memory Management algorithms.
3. It is difficult to allocate contiguous memory to variable sized partition.

Date: / / Page no:

Paging	Segmentation
① OS is responsible	① Compiler is responsible
② Divides program into fixed size pages.	② Divides program into variable size segments.
③ Faster process.	③ Slower process.
④ closer to OS.	④ closer to User.
⑤ Suffers from internal fragmentation.	⑤ Suffers from external fragmentation.
⑥ Logical address is divided into page no. & page offset.	⑥ Logical address is divided into segment no. & segment offset.
⑦ Page table maintains page information.	⑦ Segment table maintains segment information.
⑧ Page table entry has the frame no. & some flag bits to represent details about pages.	⑧ Segment table entry has the base address of the segment & some protection bits for the segment.

Date: / / Page no:

* Segmented Paging -

→ Not very popular & not being used in many OS.
→ Segmentation can be combined with Paging to get the best features out of both the techniques.
→ In segmented Paging, the main memory is divided into variable size segments which are further divided into fixed size pages.
→ Pages are smaller than segments.
→ Each segment has a page table which means every program has multiple page tables.
→ The logical address is represented as Segment No., Page No. & Page offset.

Segment No. : it points to appropriate segment Number.
Page No. : it points to the exact page within Segment.
Page Offset : used as an offset within the page frame.

* Advantages -

1. It reduces memory usage.
2. No external fragmentation.
3. It simplifies memory allocation.
4. Page table size is limited by segment size.
5. Segment table has only one entry corresponding to one actual segment.

* disadvantages -

1. Internal fragmentation is there.
2. High complexity level as compared to Paging.
3. Page tables need to be contiguously stored in memory.