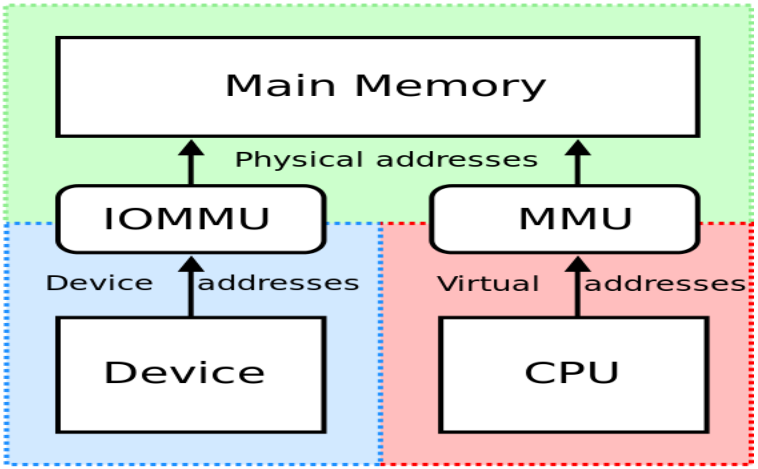
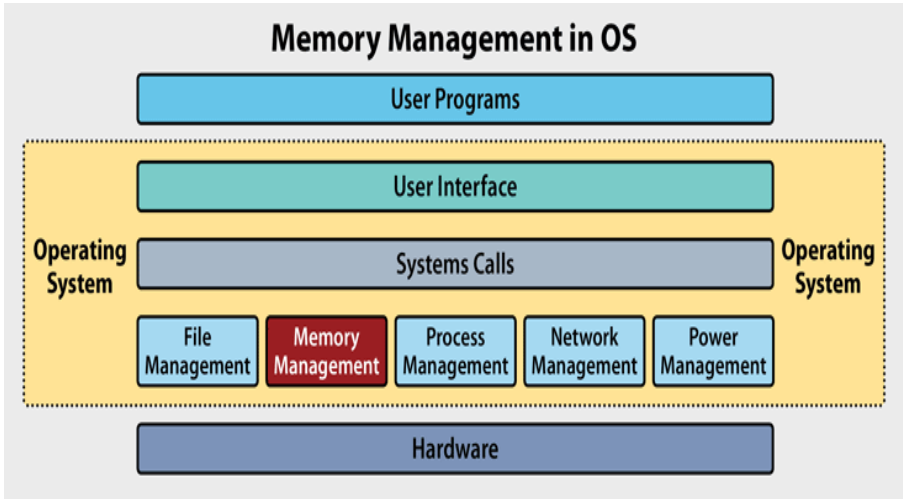
AMIRAJ

COLLEGE OF ENGINEERING & TECHNOLOGY

CHAPTER -6
MEMORY MANAGEMENT

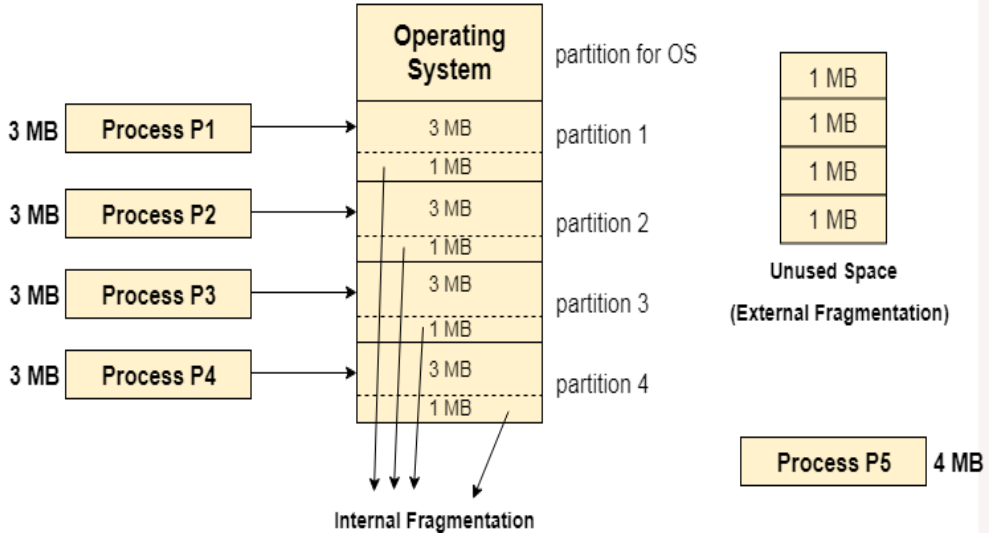




Subject:- OS **Code:-3140702**	**Prepared by:** **Asst.Prof.Foram Patel** **(Computer Department,ACET)**	AMIRAJ COLLEGE OF ENGINEERING & TECHNOLOGY

FIXED PARTITION

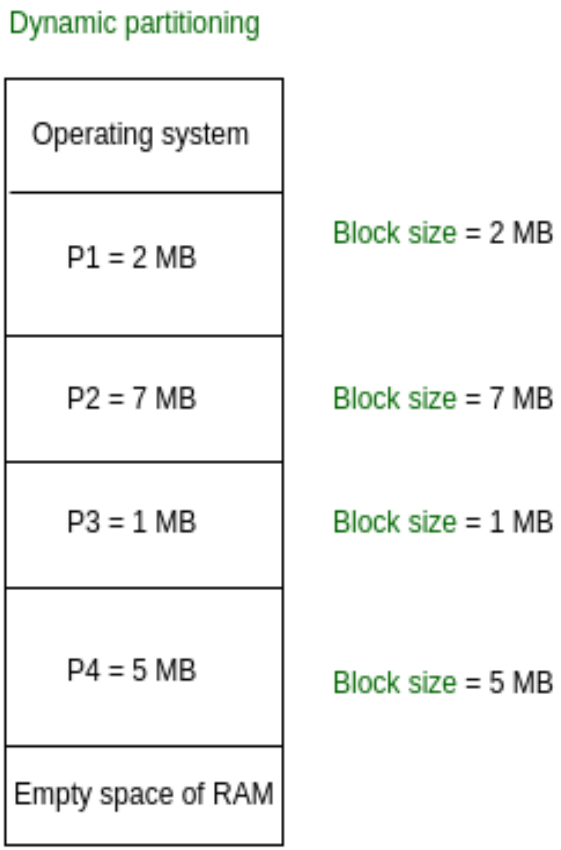
- Oldest and simplest technique
- To load more than one processes into the main memory.
- Main memory is divided into partitions of equal sizes.
- Number of partitions (non-overlapping) in RAM are fixed but size of each partition may or may not be same.



- No spanning is allowed.
- Easy to implement.
- Little OS overhead.

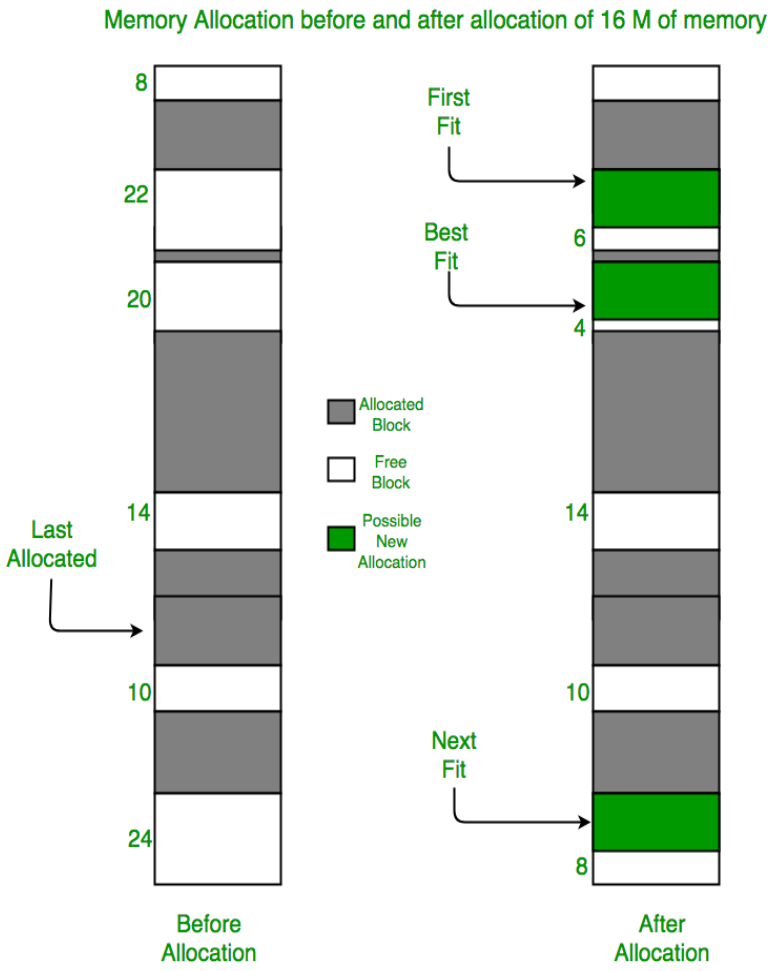
VARIABLE PARTITION

- Partitions are made during the run-time.
- Size of partition will be equal to incoming process.
- Partition size varies according to the need of the process.
- RAM is not fixed and depends on the number of incoming process and Main Memory's size.
- There will be no unused space left in the partition.
- The process size can't be restricted since the partition size is decided according to the process size.



MEMORY ALLOCATION

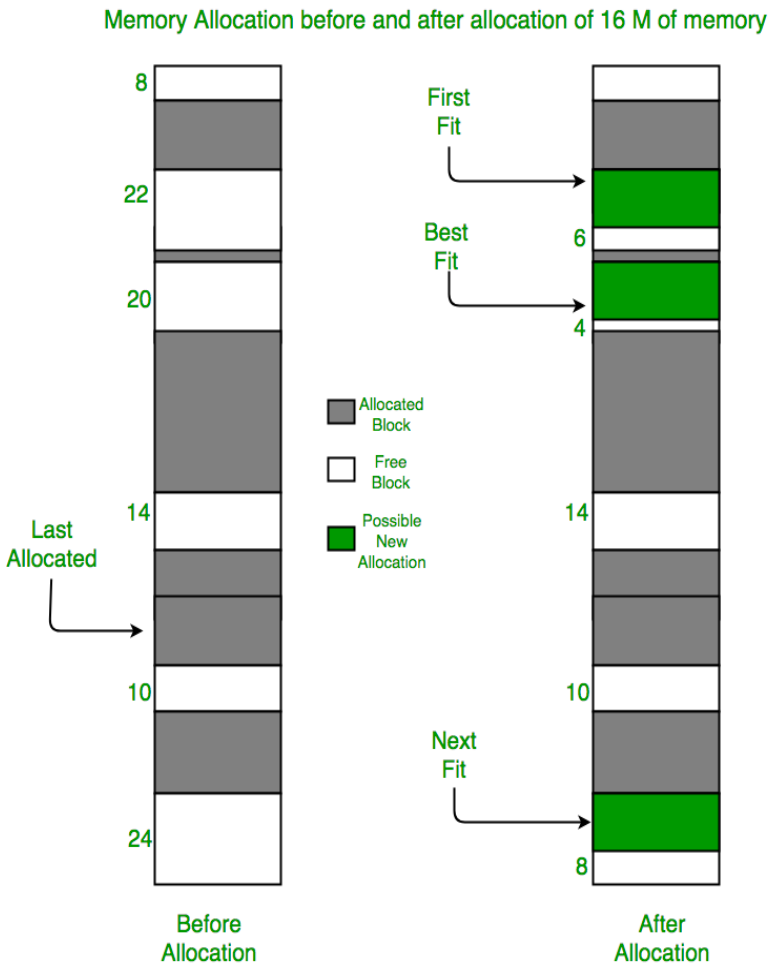
- **First Fit:**
- The free/busy list of jobs organized by memory location, low-ordered to high-ordered memory.
- First available memory with space more than or equal to it's size.
- **Best Fit:**
- The free/busy list in order by size – smallest to largest
- The os searches the whole according to the size of the given job and allocates it to the closest-fitting free partition in the memory



AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

MEMORY ALLOCATION

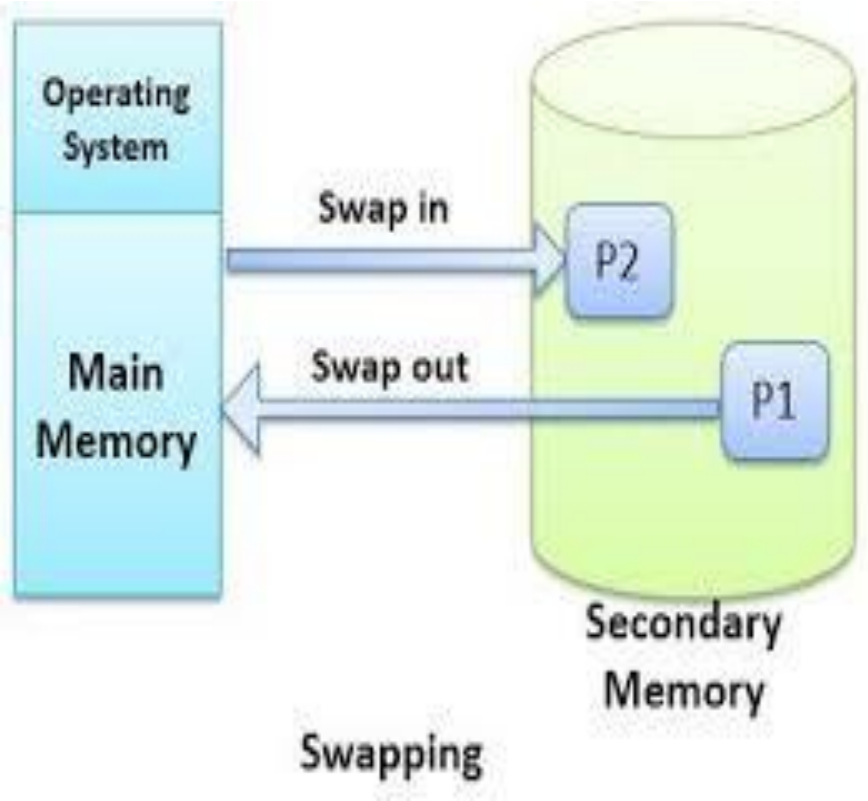
- **Worst Fit:**
- Scans the entire list every time and tries to find out the biggest hole.
- This algorithm produces the larger holes to load the other processes
- It is slower because it searches the entire list every time
- **Next Fit:**
- Next Fit algorithm is similar to First Fit .
- Next fit doesn't scan the whole list, it starts scanning the list from the next node



AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

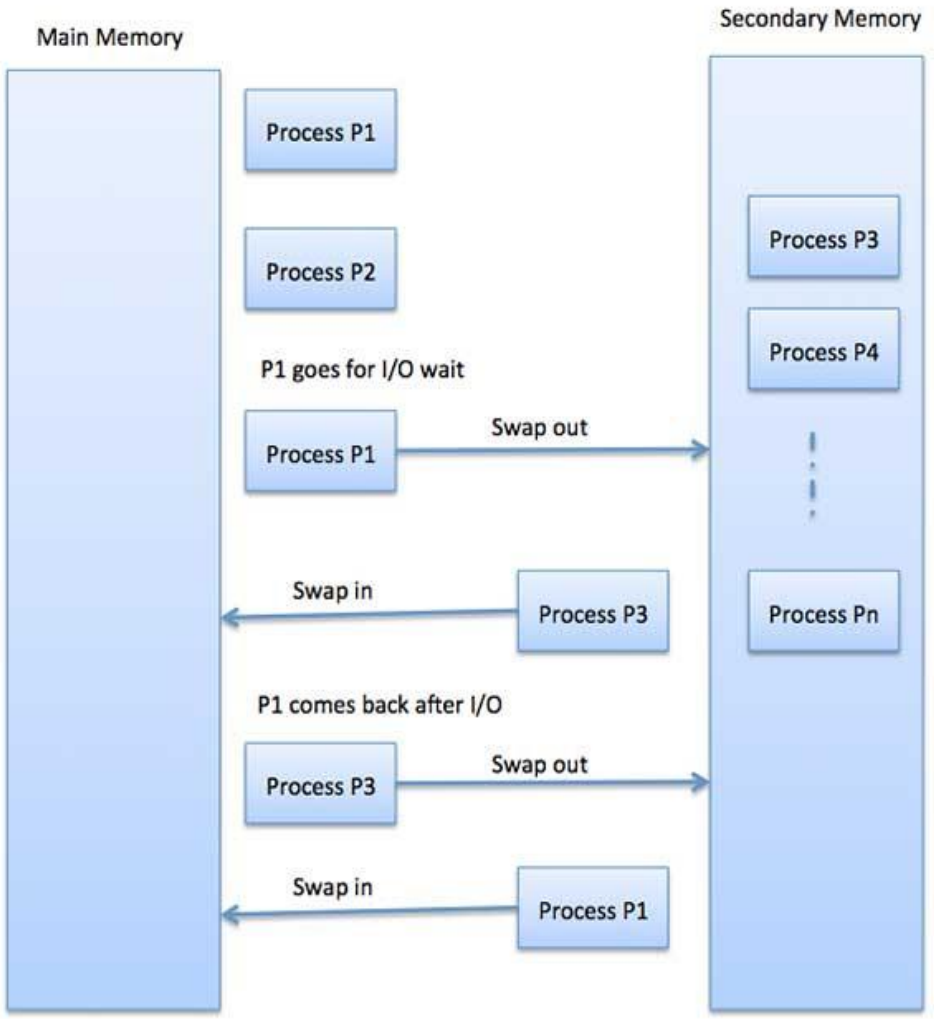
SWAPPING

- **Swapping:**
- A process must be in the main memory before it starts execution.
- A process that is ready for execution is brought in the main memory.
- If a running the process gets blocked.
- The memory manager temporarily swaps out that blocked process on to the disk.
- Makes the space for another process in the main memory.



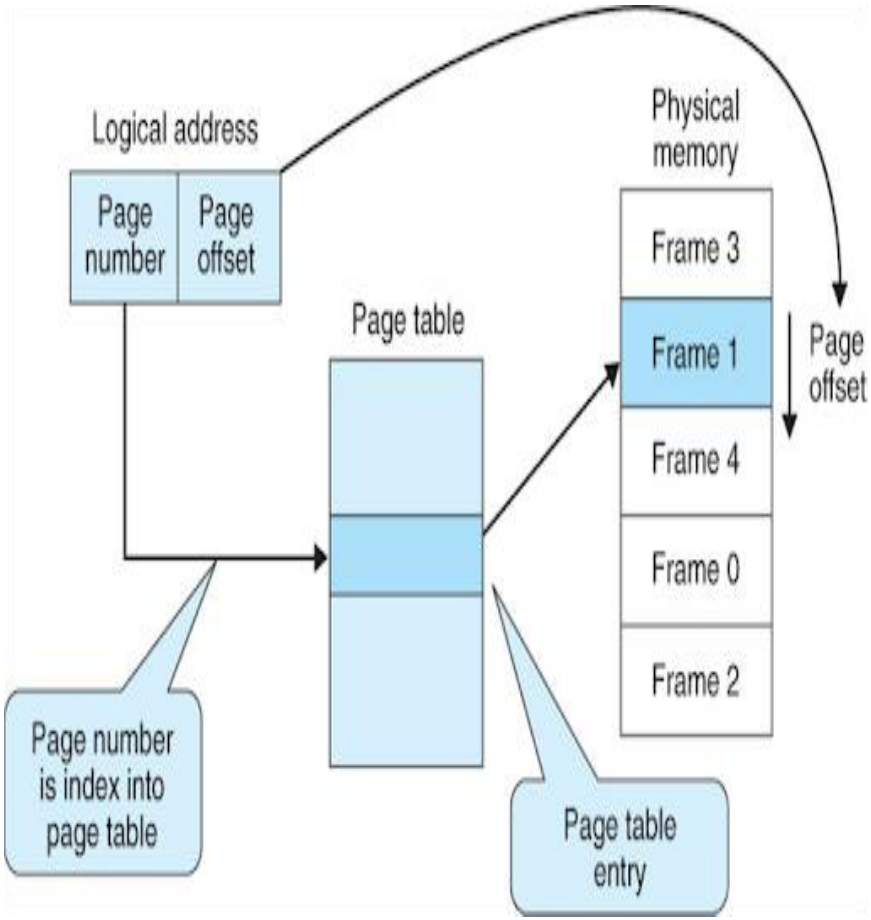
SWAPPING

- The memory manager swaps in the process ready for execution from disk
- Swapped out process is also brought back into the main memory
- Swapping of the processes also depends on the priority-based pre-emptive scheduling
- Allows dynamic relocation.
- It helps to get better utilization of memory.
- Minimum wastage of CPU time



PAGING

- Eliminates the need for contiguous allocation of physical memory
- The Physical Address Space is conceptually divided into a number of fixed-size blocks, called **frames**.
- The Logical address Space is also splitted into fixed-size blocks, called **pages**.
- Logical address is divided into page number and page offset

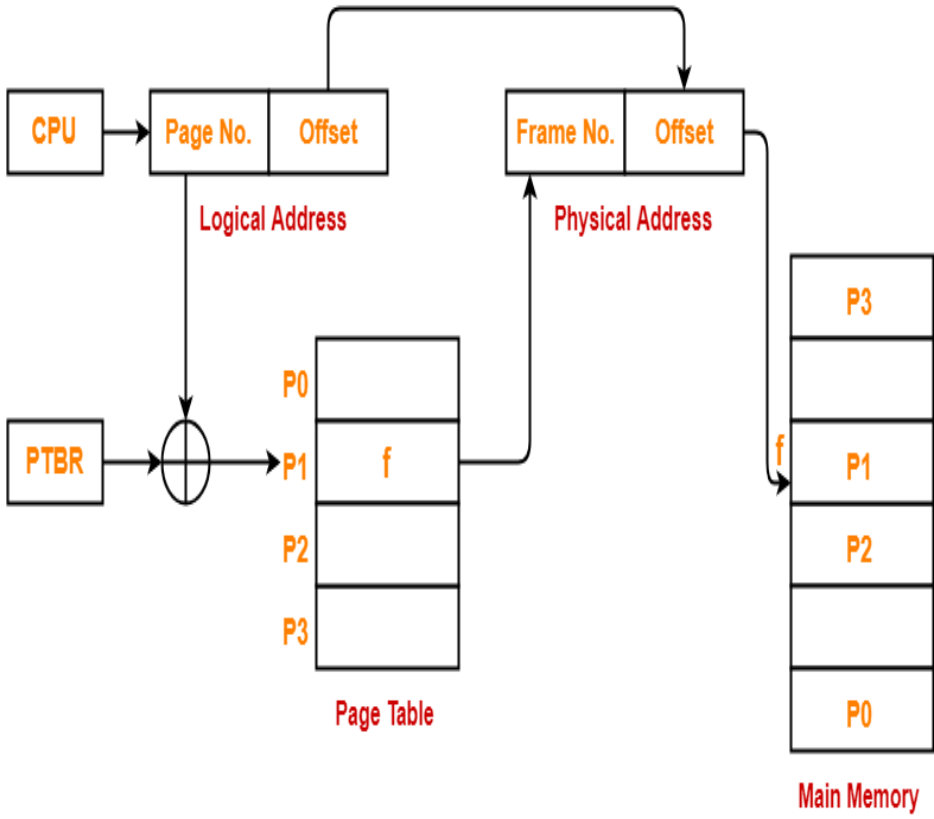


PAGING

- CPU generates a logical address consisting of two parts-

Page Number, Page Offset

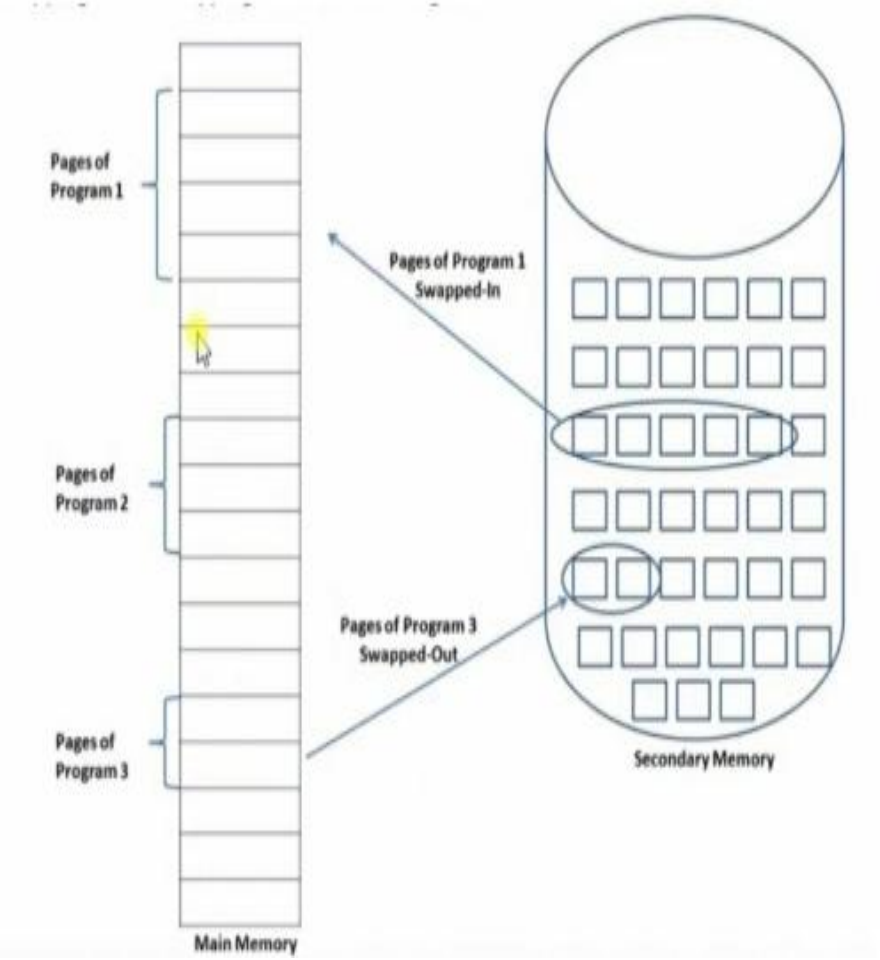
- Page Table provides the corresponding frame number
- The frame number combined with the page offset forms the required physical address.
- Frame number specifies the specific frame where the required page is stored.
- Page Offset specifies the specific word that has to be read from that page.



Translating Logical Address into Physical Address

DEMAND PAGING

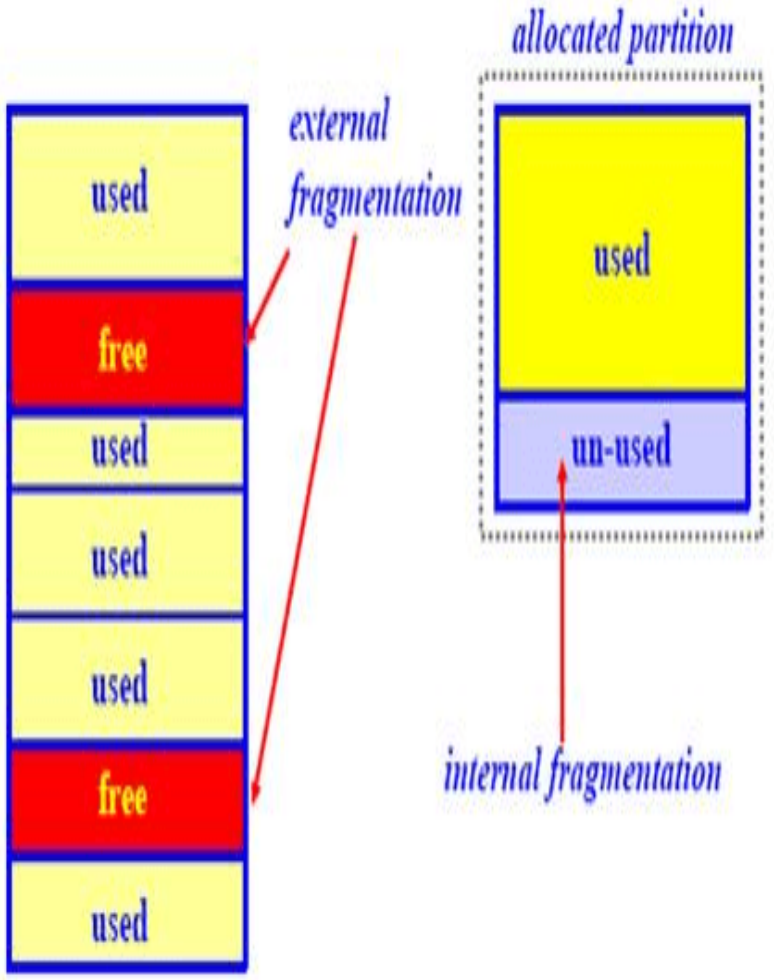
- A demand paging mechanism is very much similar to a paging system with swapping
- processes stored in the secondary memory and pages are loaded only on demand, not in advance
- In demand paging, the pages are of equal size.
- It does not allows sharing of the pages.
- In demand paging, on demand pages are loaded in the memory.
- It provides large virtual memory and have more efficient use of memory.



AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

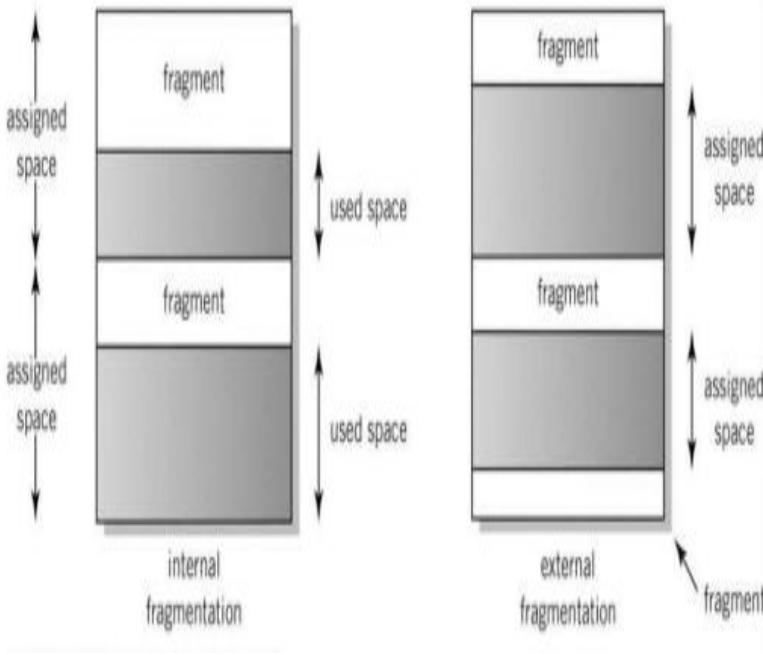
FRAGMENTATION

- As processes are loaded and removed from memory, the free memory space is broken into little pieces.
- processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused.
- At the time of process loading and swapping there are many spaces left which are not capable to load any other process due to their size.



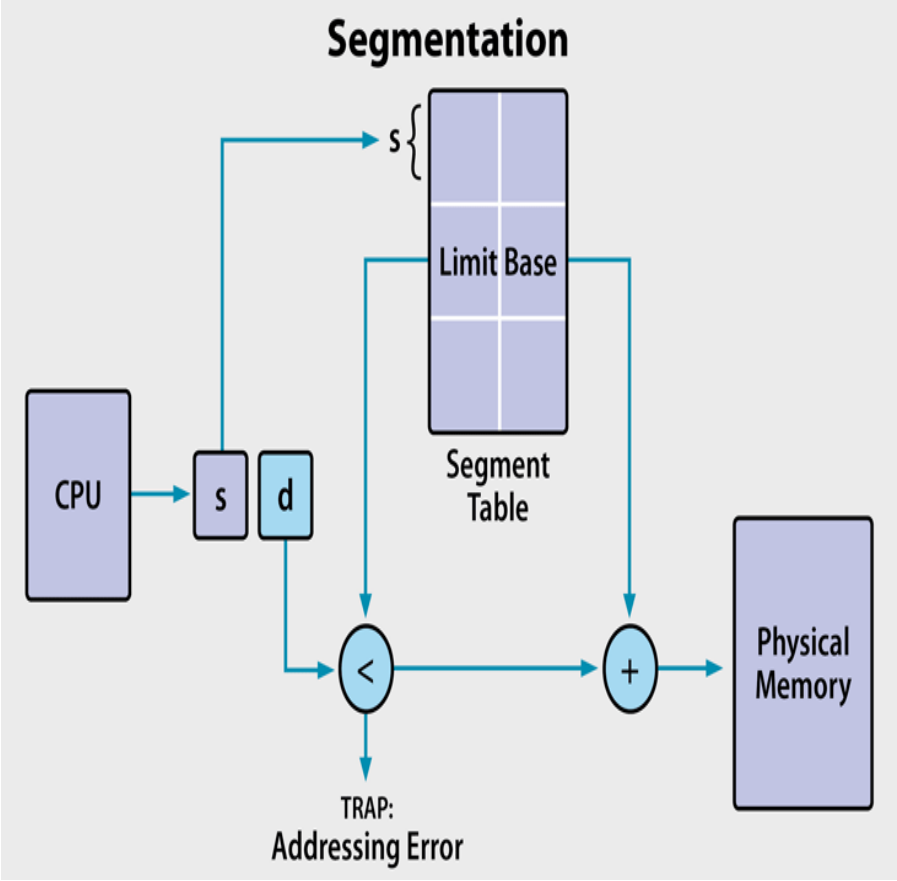
FRAGMENTATION

- **Internal fragmentation:**
- when the memory is split into mounted sized blocks.
- An approach is to allocate very small holes as part of the larger request.
- The allocated memory may be larger than the requested memory.
- The solution of internal fragmentation is best-fit block.
- **External fragmentation:**
- The process's memory request cannot be fulfilled because the memory offered is during a non-contiguous manner.
- Solution of external fragmentation is compaction, paging and segmentation.



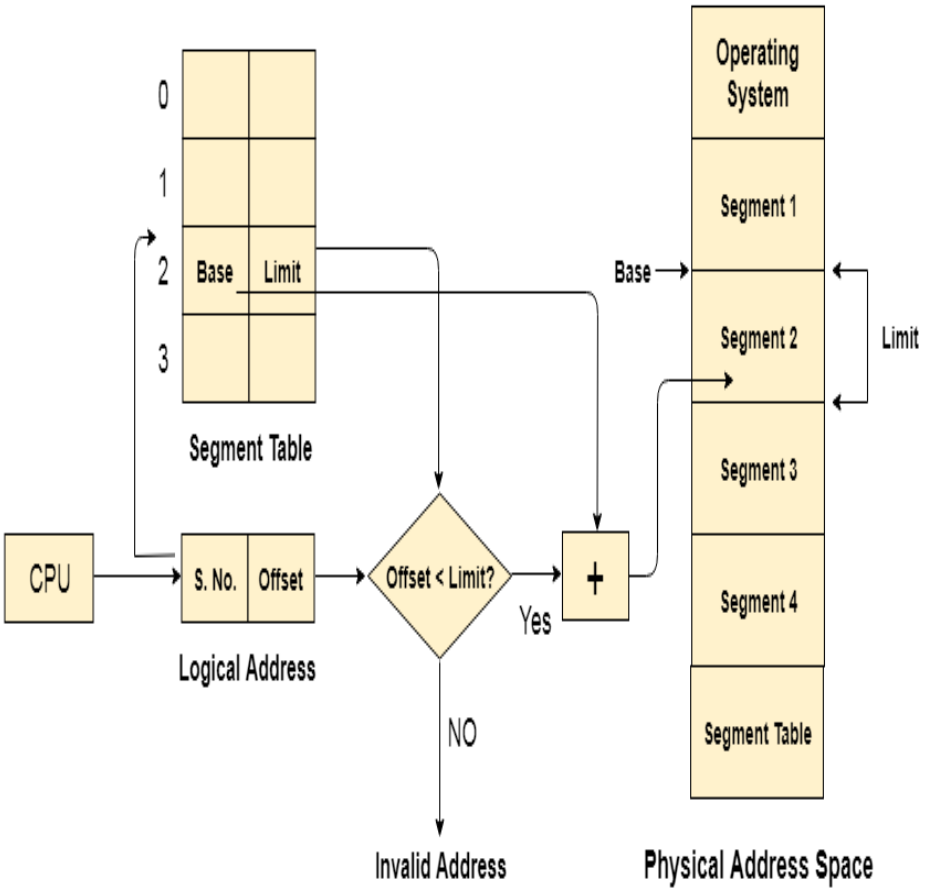
SEGMENTATION

- Segmentation is a memory management technique in which, the memory is divided into the variable size parts
- **Virtual memory segmentation:** Each process is divided into a number of segments, not all of which are resident at any one point in time.
- **Simple segmentation:** Each process is divided into a number of segments, all of which are loaded into memory at run time, though not necessarily contiguously.



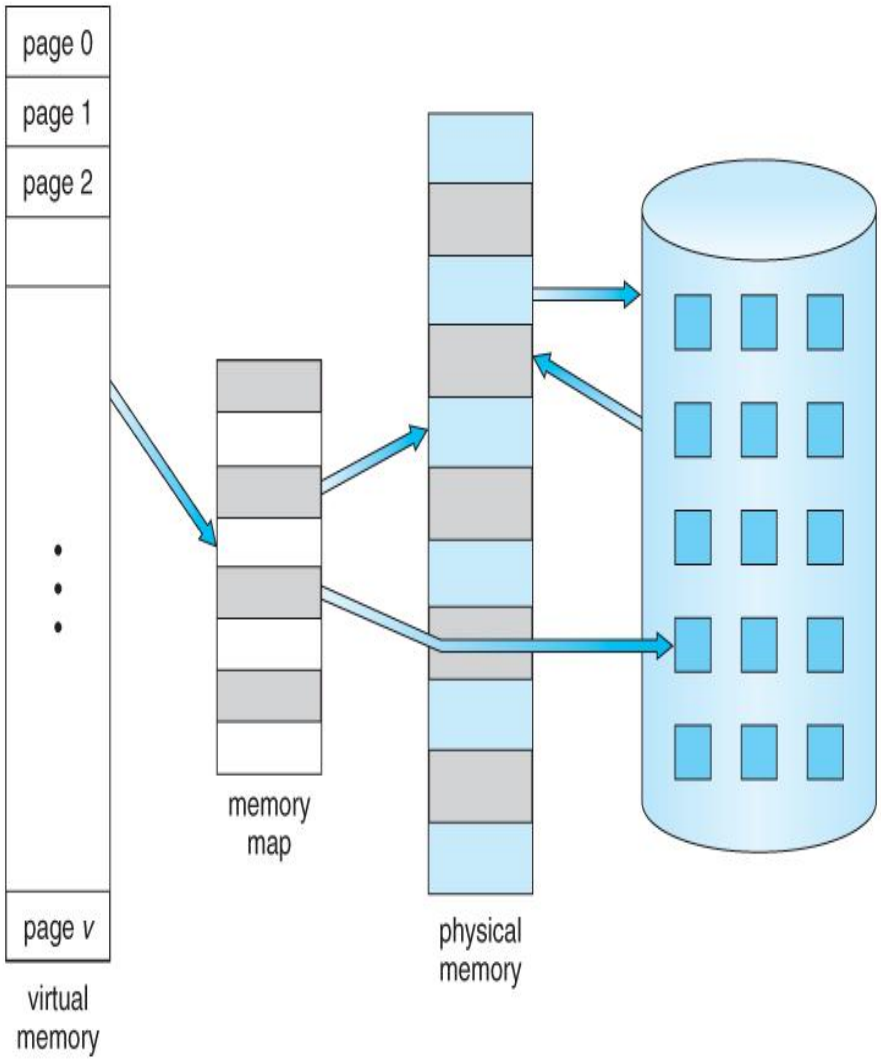
SEGMENTATION

- CPU generates a logical address which contains two parts:
  Segment Number , Offset
- The Segment number is mapped to the segment table.
- The limit of the respective segment is compared with the offset
- If the offset is less than the limit then the address is valid otherwise it throws an error as the address is invalid
- Valid address, the base address of the segment is added to the offset to get the physical address of actual word in the main memory.



VIRTUAL MEMORY

- **Virtual Memory** is a storage mechanism which offers user an illusion of having a very big main memory.
- The user can store processes with a bigger size than the available main memory.
- **Virtual memory serves two purposes:**
- First, it allows us to extend the use of physical memory by using disk.
- Second, it allows us to have memory protection, because each virtual address is translated to a physical address.



PAGE REPLACEMENT ALGORITHM

- Want lowest page fault rate.
- Evaluate algorithm by running it on a particular string of memory references and computing the number of page faults and page replacements on that string.

- FIFO page replacement
- Optimal page replacement
- LRU page replacement

FIFO (FIRST IN FIRST OUT) PAGE REPLACEMENT

- Want lowest page fault rate.
- Simple to implement.
- When the buffer is full, the oldest page is replaced. Hence first in first out :
- A frequently used page is often the oldest, so it will be repeatedly paged out by FIFO.
- Easy to understand.
- Performance is not always good.

FIFO (FIRST IN FIRST OUT) PAGE REPLACEMENT

1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
1	1	1	1	1	2	2	3	5	1	6	6	2	5	5	3	3	1	6	2
	2	2	2	2	3	3	5	1	6	2	2	5	3	3	1	1	6	2	4
		3	3	3	5	5	1	6	2	5	5	3	1	1	6	6	2	4	3
*	*	*			*		*	*	*	*		*	*		*		*	*	*

FIFO
Total 14 page faults

OPTIMAL PAGE REPLACEMENT ALGORITHM

- An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms.
- It is practically impossible to implement this algorithm.
- This is because the pages that will not be used in future for the longest time can not be predicted.
- In this algorithm, pages are replaced which would not be used for the longest duration of time in the future.

OPTIMAL PAGE REPLACEMENT ALGORITHM

7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7

we have 3 page slots empty

	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7
			1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1
		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0
	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7
	*	*	*	*	hit	*	hit	*	hit	hit	*	hit	hit	*	hit	hit	hit	*

Page hits = 9 page faults = 9

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY

LRU PAGE REPLACEMENT ALGORITHM

- Page which has not been used for the longest time in main memory.
- Easy to implement, keep a list, replace pages by looking back into time.
- This algorithm works on the principle of “Least Recently Used“.
- It replaces the page that has not been referred by the CPU for the longest time.

LRU PAGE REPLACEMENT ALGORITHM

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2		4	4	4	0			1		1		1		
	0	0	0		0		0	0	3	3			3		0		0		
		1	1		3		3	2	2	2			2		2		7		

- **PAGE HIT: 8**
- **PAGE FAULT : 12**

AMIRAJ
COLLEGE OF ENGINEERING & TECHNOLOGY