4. Memory Management

8.1 Explain different types of memory fragmentation.

⇒ i) As processes are loaded and removed from memory, the free memory space is broken into little pieces.

ii) It happens after sometimes that processes cannot be allocated to memory considering their small size and memory blocks remains unused.

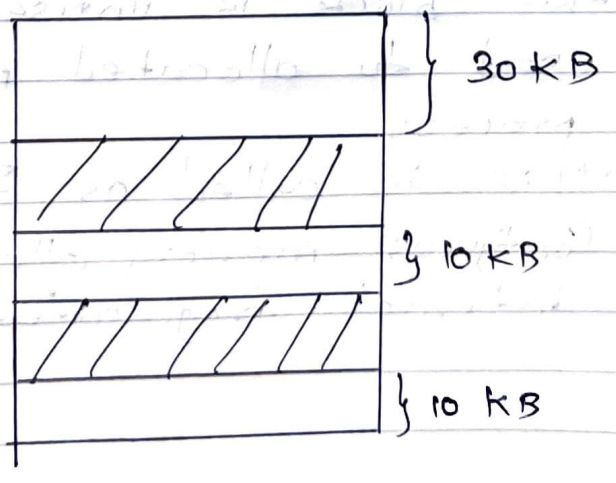
iii) This problem is known as Fragmentation

iv) Fragmentation is of two types:

a) External Fragmentation

⇒ External Fragmentation happens when a dynamic memory allocation method allocates some memory but leaves a small amount of memory unusable.

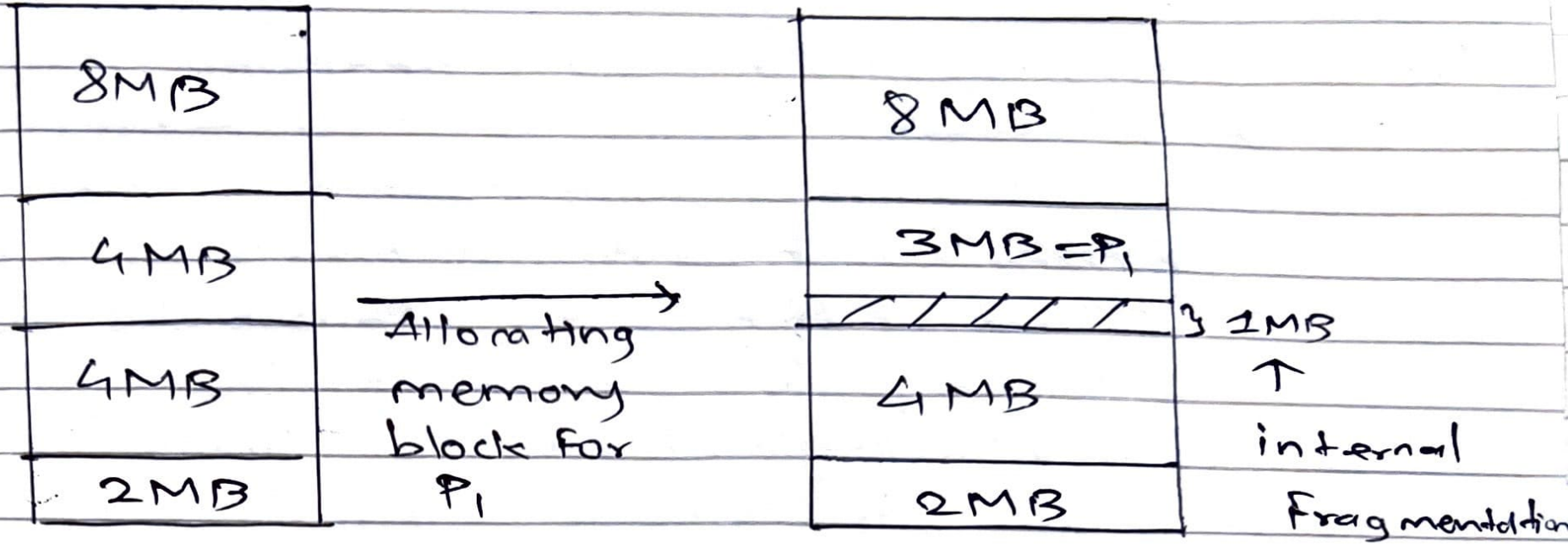
- The quantity of available memory is substantially reduced if there is too much external fragmentation.
- There is enough memory to complete a request, but it is not contiguous.
- This is called as External Fragmentation
- For e.g.)



- In above example, we can see there is a sufficient memory space available to run process of 45KB, but the available memory (50KB) is not contiguous.
- So, the solution to this external fragmentation problem is paging, compaction and segmentation.

b) Internal Fragmentation

⇒ Internal Fragmentation takes place in fixed size partitioning technique.
- If the program size is smaller than partition size, some space of partition will be unused within that partition called as internal fragmentation.
- For e.g.,
- Assume that memory allocation in RAM is done using fixed partitioning.
- 2MB, 4MB, 4MB, 8MB are available sizes
- Suppose process P_1 with size of 3MB arrives and is given a memory block of 4MB.
- As a result, 1MB of free space in this block is unused and cannot be used to allocated memory to another process.
- This is called as Integral Fragmentation.
- Contiguous memory allocation reduces the internal fragmentation.


8MB
4MB
4MB
2MB
Allocating memory block for P_1
8MB
3MB = P_1
1MB
4MB
2MB
internal Fragmentation

Q.3 Explain concept of Paging

⇒ i) Paging is a memory ~~memo~~ management technique used by operating system to implement the concept of virtual memory.

ii) It coorks by dividing the physical memory of computer into fixed-size chunks called as pages.

iii) These pages are then used to store memory pages from running application.

iv) When an application requests memory that is not available in physical memory, the OS performs a process called page fault.

v) During the page fault, the OS checks if the requested memory is present in physical memory.

vi) If it is not present, in OS selects a page to be evicted from physical memory and swapped out to the hard wire.

vii) This frees up space in physical memory for requested to be loaded from the hard drive into physical memory.

Q.4 Explain effect of page size on performance.

⇒ i) Page size is the important factor to decide the performance.

ii) If a page size is small then any program will be divided in many no. of pages leading to have a large page table.

iii) The no. of frames is equal to the size of memory divided by the page-size.

iv) So and increase in page size means a decrease in the no. of available frames.

v) Having a fewer frame will increase the no. of pages faults because of the lower freedom in replacement choice.

vi) Larger pages would also waste space by internal fragmentation.

vii) On the other hand, a larger page-size could draw in more memory per fault; so the no. of fault may decreases if there is limited contention.

viii) Larger pages also reduce the no. of TLB misses.

Q.5 Explain the concept of segmentation.

⇒ i) Paging is not usually visible to the programmer.

ii) Segmentation is visible to the programmer. It is a memory management technique that supports user's view of memory.

iii) Each segment has a name and a length.

iv) A logical address in segmentation is specified -
   a) A segment name
   b) An offset within segment.

v) The user therefore specifies each address by 2 quantities : segment no. and the offset.

vi) The conversion of logical address to physical address :
   a) This mapping is done through segment table.
   b) Every process has its own segment table.
   c) Every entry in the segment table has segment limit and segment base.
   d) Segment limit specifies the length of segment.
   e) Segment base specifies the starting physical address where segment resides in memory.

- For e.g.,

Segment 0
size = 800

segment 1
size = 1200

segment 2
size = 900

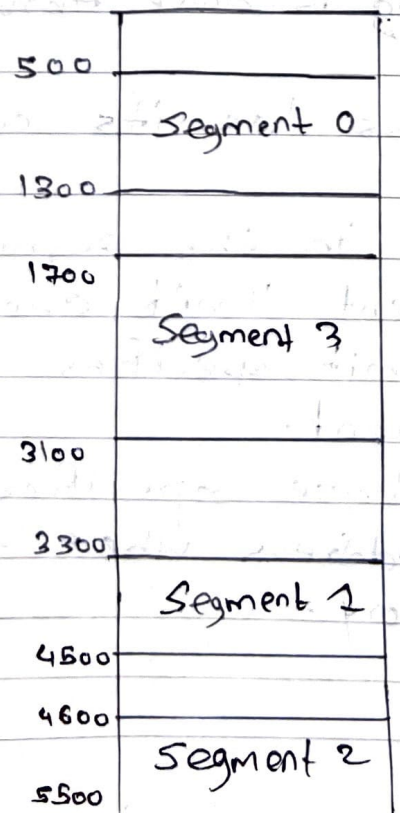
Segment 3
size = 1400

Logical address space

	Limit	Base
0	800	500
1	1200	3300
2	900	4600
3	1400	1700

← segment table

- It's physical memory is -



Q.14 What is TLB?

⇒ i) TLB is translation lookaside buffer which contains most recently used Page table entries.

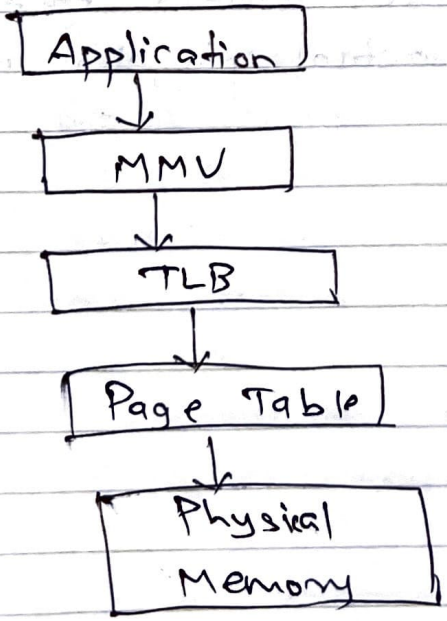
ii) Logical address is in the form -

Page No	OFFSET

iii) During the translation process MMU consults the TLB to see if the matching page table is present.

iv) IF match found, physical address is generated by combining the frame no. present at that entry in the TLB with the offset.

v) IF match not found, entry is accessed from the page table in the memory.



Q.15 Explain demand paging.

=> i) Demand paging is a memory management technique that allows OS to manage system memory more efficiently by loading only those pages of a program into memory that are actually needed at given moment.

ii) This approach conserves memory by not loading unnessary pages into physical memory and improves perfoms by reducing the amount of time it takes to load a program into memory.

iii) Demand paging allows programs to be larger than physical memory since only a portion of the program needs to be loaded into memory at any given time.

iv) When program accesses a page that is not in physical memory, a page fault exception is raised and the required page is loaded from disk to physical memory.

v) The page table is key component of demand paging, as it keeps track of the mapping b/w virtual and physical addresses.

vi) The TLB is also used in demand paging to cache recently accessed virtual-to-physical address translation.

Q.17 Explain different page replacement algorithms

⇒ • Page Replacement Algorithms -

i) First-In, First-Out (FIFO)

⇝ In this algorithm, the OS replaces the page that was first brough into memory. It maintains a queue of all the pages in memory, and the oldest page is replaced when a new page needs to be brought in.

For eg.,

Let the string be

2	3	2	1	5	2	4	5	3	2	5	2
2	2	2	2	5	5	5	5	3	3	3	3
	3	3	3	3	2	2	2	2	5	5	
			1	1	1	4	4	4	4	4	2
		H					H		H		

∴ Hit ratio = 3/12

ii) Least-Recently Used (LRU)

⇒ In this algorithm, the OS replaces the page that has not been accessed for the longest time.

It maintains a list of all the pages in memory, and the least recently used page is replaced when a new page needs to be brought in.

For e.g.)

Let the string be -

2	3	2	1	5	2	4	5	3	2	5	2
2	2	2	2	2	2	2	2	3	3	3	3
	3	3	3	5	5	5	5	5	5	5	5
			1	1	1	4	4	4	2	2	2
		H			H		H			H	H

Hit ratio = 5/12

No. of page faults = 7

iii) Optimal Page replacement

⇒ In this algorithm, the OS replaces the page that will not be needed for the longest time in the future.
However, this algorithm is not pratical because it requires knowledge of the future, which is impossible to implement.
For e.g.,
Let the string be.

2	3	2	1	5	2	4	5	3	2	5	2
2	2	2	2	2	2	4	4	4	2	2	2
	3	3	3	3	3	3	3	3	3	3	3
			1	5	5	5	5	5	5	5	5
		H			H		H	H		H	H

∴ Hit ratio = 6/12
= 1/2

Q.19 Explain thrashing.

=> i) Thrashing occurs when a system does not have enough physical memory to support the demands of processes running on it.

ii) As a result, the OS begins to swap pages between physical memory and disk to free up space for new pages.

iii) When system is thrashing, it spends more time swapping pages than executing processes, which can significantly slow down the system's performance.

iv) Thrashing can occur due to variety of factors, such as running too many processes simultaneously, using a page size that is too small, or having insufficient physical memory.

v) To prevent thrashing, the OS can use techniques such as working sets or page replacement algorithms to determine which pages to keep in physical memory.

vi) If a system is already thrashing, reducing the no. of processes running, increasing physical memory, or adjusting the page size may help alleviate the problem.