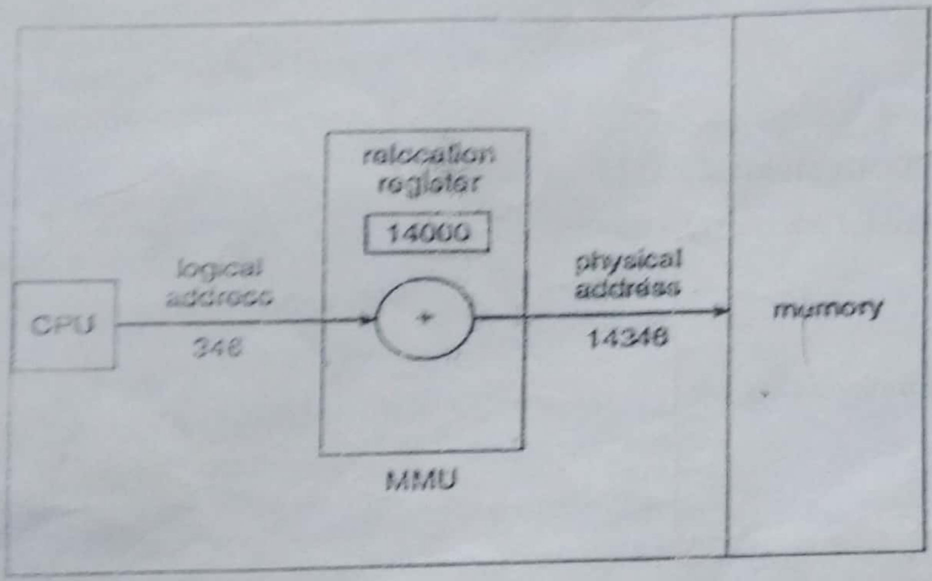
Varshith

e Q. WRITE ABOUT LOGICAL ADDRESS SPACE AND PHYSICAL ADDRESS SPACE?

An logical address is generated by the CPU is commonly referred as logical address and it is found in the memory unit is called as physical address.

The collection of logical address that generates a program is called as logical address space and the collection of physical addresses corresponding to the logical address is called as physical address space.

For example, it can be represented as follows .



Logical vs physical address space

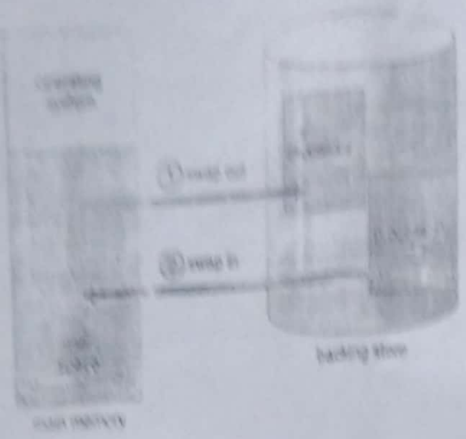
Basis for Comparison	Logical Address	Physical Address
Basic	It is the virtual address generated by CPU	The physical address is a location in a memory unit.
Address Space	Set of all logical addresses generated by CPU in reference to a program is referred as Logical Address Space.	Set of all physical addresses mapped to the corresponding logical addresses is referred as Physical Address.
Visibility	The user can view the logical address of a program.	The user can never view physical address of program
Access	The user uses the logical address to access the physical address.	The user can not directly access physical address.
Generation	The Logical Address is generated by the CPU	Physical Address is Computed by MMU

Q. WRITE ABOUT SWAPPING? 15481

A process swapped temporarily from memory to a backing store and then brought back into main memory for continued execution is called swapping so, the swapping requires a backing store.

The backing store is commonly a fast disk.

For example, a multi programming environment requires with a round robin cpu scheduling al [illegible]
When the time slice expires the memory manager will start to swap out the process and swap in [illegible]
process into main memory it can be represented as follows.



Q. EXPLAIN ABOUT CONTIGUOUS MEMORY ALLOCATION?

In generally the memory can be divided into two types they are

System area

User area

An user area can be divided into partitions. They are

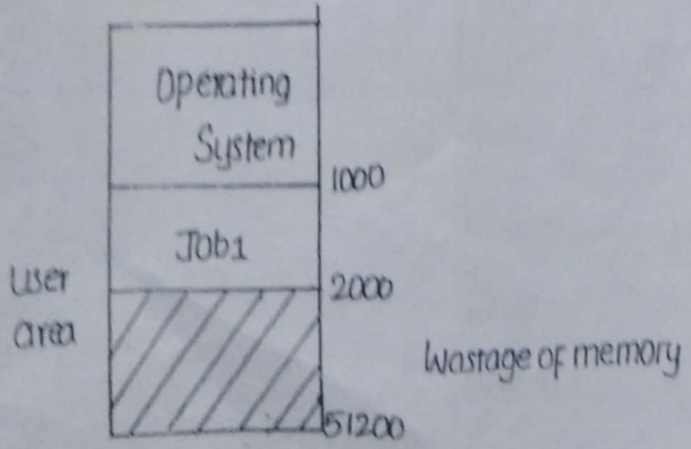
1. Single portioned allocation

2. Multiple portioned allocation

SINGLE PARTITIONED ALLOCATION:-

In this mechanism, the user area is treated as only one partition. So we can execute only one job at a time into main memory. This allocation doesn't support multiprogramming environment. The main disadvantages of this allocation is a lot of memory will be wasted.

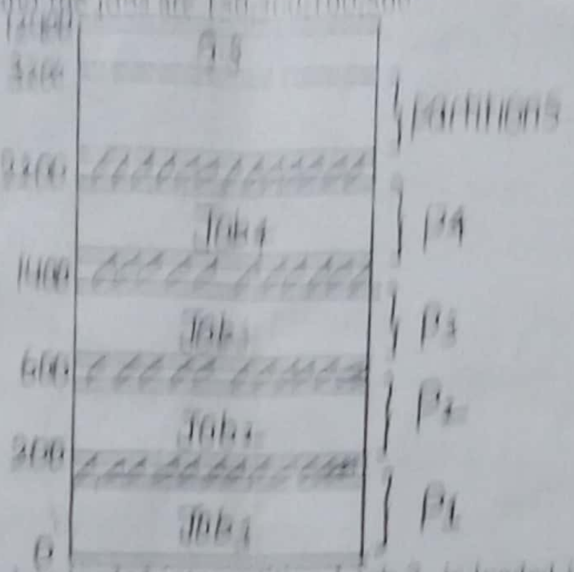
For example it can be represented as follows



MULTIPLE PARTITIONED ALLOCATION:-

In this mechanism user area can be divided into fixed sized partitions. Each partition Can execute exactly one partition at a time. It supports multi programming. It depends on the no. of partitions. For example, an user area can be divided into 5 partitions [illegible] are 200,400,600,800,1000 and the jobs are 150,300,100,500

can be represented as follows



Job	memory(k)
J1	150
J2	300
J3	100
J4	500

From the above example the job 1 is loaded into partition 1 job 2 is loaded into partition 2 and j3 is loaded into partition 3 and so on. The wastage of memory in the partition is entirely wasted is called as external fragmentation

The wastage of memory space in the internal and external fragmentation can be grouped then it is called as "compaction".

ADVANTAGES:

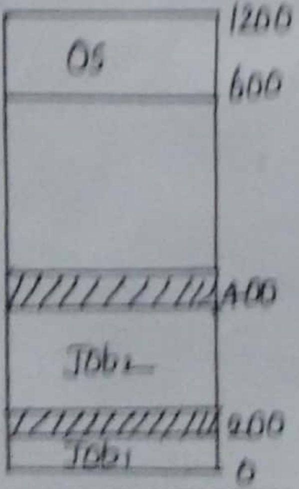
1. It supports multi programming environment.
2. It supports time sharing systems

DISADVANTAGES:

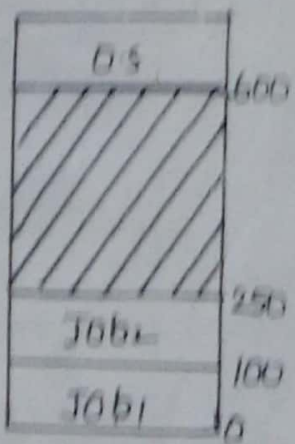
1. Suffering from internal and external fragmentation
2. The main memory is not used in proper way.

Q. DEFINE COMPACTION AND GIVE EXAMPLE?

The collection of internal and external fragmentation is called as compaction. The grouping of wastage of memory space that belongs to internal and external fragmentation



Job	memory
J1	100
J2	150



Q. EXPLAIN ABOUT MEMORY ALLOCATION ALGORITHMS?

Initially, all the memory which is available for the user is known as large hole. Whenever a process arrives, it is placed in that hole. This type of allocation is known as Contiguous Memory Allocation.

Now we divide our memory into several blocks. Whenever a
performs a set in the free holes to accommodate the process. There are 3 typ
algorithms

They are

1.first fit

2.best fit

3.worst fit

a) First-fit: Allocate the first hole that is big enough.

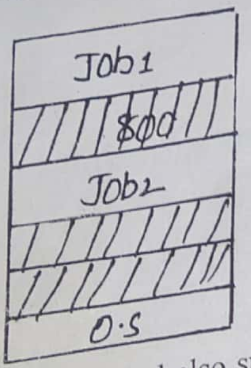
b) Best-fit: Allocate the smallest hole that is big enough; we must search entire list, unless ordered by size and produce the smallest leftover hole.

c) Worst-fit: Allocate the largest hole; we must also search entire list and produce the largest leftover hole.

First-fit and best-fit are better than worst-fit in terms of speed and storage utilization..

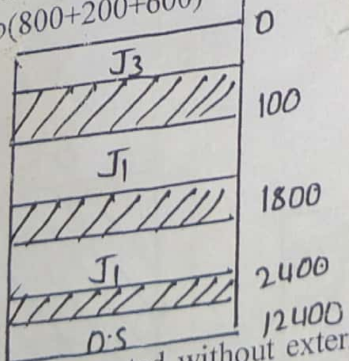
These algorithms are explained with the following example and find out the wastage of memory space.

The partitions are 1000,800,600 and the jobs are 200,600 and 900.

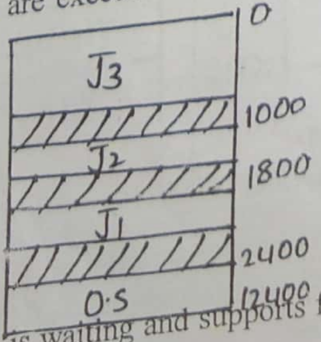


Job	memory
J_1	200
J_2	600
J_3	900

In this example the job is waiting and also supports internal and external fragmentation the wastage of memory space is 1600kb(800+200+600)



In this example all jobs are executed without external fragmentation the wastage of memory space is 100+200+400=700kb.



In this example the job 3 is waiting and supports from ainternal fragementation the wastage of memory space 800+200+600=1600kb

XPLAIN PAGING IN MEMORY MANAGEMENT? 8

the OS, the logical memory is divided into fixed sized blocks is called pages and the physical memory is divided into fixed sized blocks called frames. The page size must be Equal to the frame size. So it removes the internal fragmentation. The size of the page frame is depending on operating system

every address generated by the cpu is divided into 2 parts viz..,

a) Page number (p) – used as an index into a page table which contains base address of each page in physical memory.

b) Page offset (d) – combined with base address to define the physical memory address that is sent to the memory unit.

The physical memory contains two parts

1.frame number

2.page number

The page table is used to mapping between the logical memory and physical memory

Advantages :-

It supports time sharing system

It doesnot effects by the fragementation

It supports virtual memory

The page table is used to dynamically convert logical addresses into physical addresses

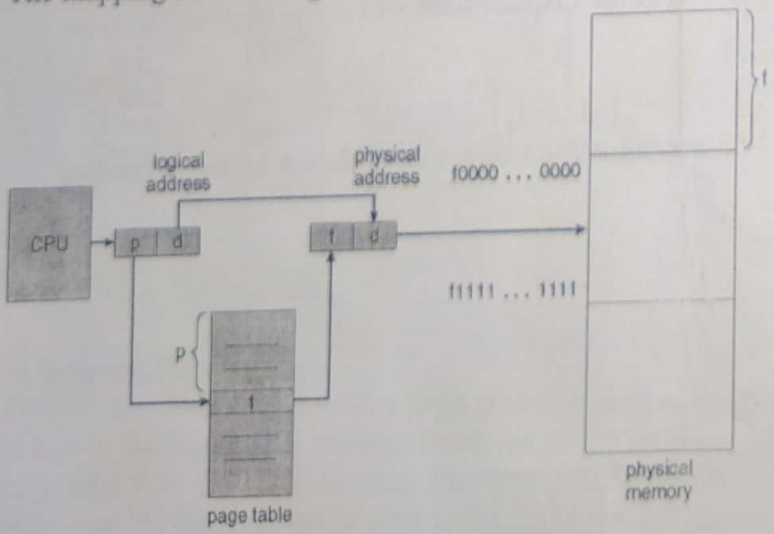
Small frames reduce fragmentation, but increase the size of the page table

Disadvantages:-

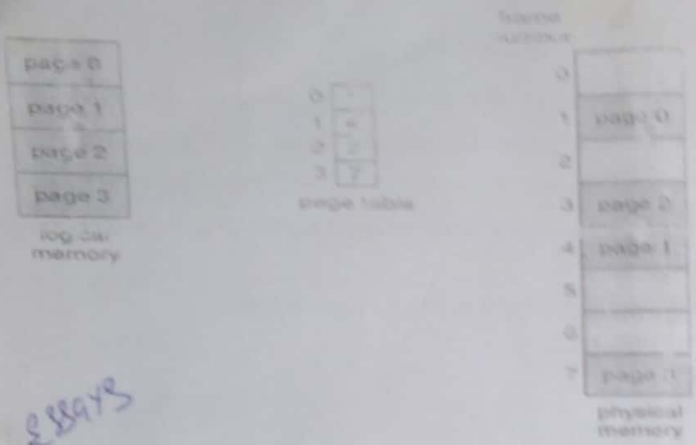
If the no. of pages are high it is difficult to maintain page tables

This suffers from page breaks

The mapping between logical and physical memories are represented as follows



For example the logical memory can be divided into 4 pages like page 1,page2,Page3,page4 these are mapped into page table in physical memory is represented as follows



Essays

Q) WRITE ABOUT STRUCTURE OF PAGE TABLE?

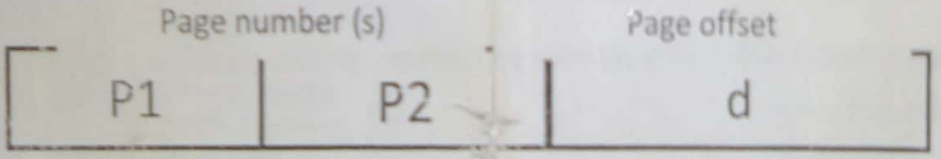
Page table structure : The following are the most common techniques for structuring the page table:

1. Hierarchical paging
2. Hashed page tables
3. Invented page tables

1) Hierarchical Paging (multilevel paging):

- ➔ Suppose a computer has a large amount of auxiliary memory space say 2^{32} words and we assume that page size is 1 KB that equal to 2^{10}, so we have $2^{32}/2^{10} = 222$ pages.
- ➔ So our page table contains 2^{22} entities and we assume that the main memory capacity is 4 KB.
- ➔ So, at any given time only 4 pages out of 2^{22} pages are stored in the main memory i.e., our page table contains$2^{22} - 4$ null entries that is maximum amount of space in this page table is waste and it is not easy to store entire page table in contiguous memory location.
- ➔ This drawback is cleared in multilevel paging.
- ➔ In the two level paging scheme, page table is again paged.

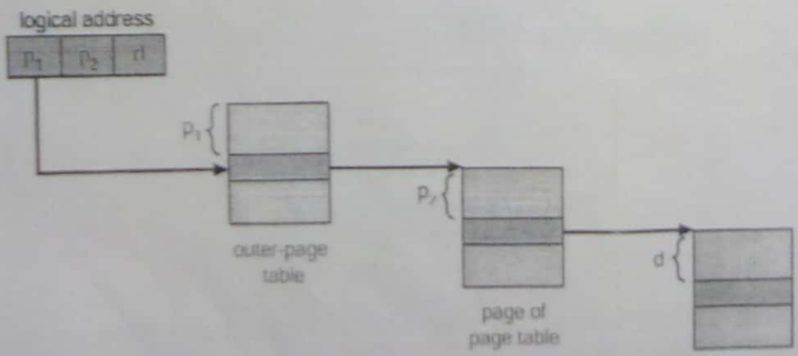
This is shown in the following diagram.

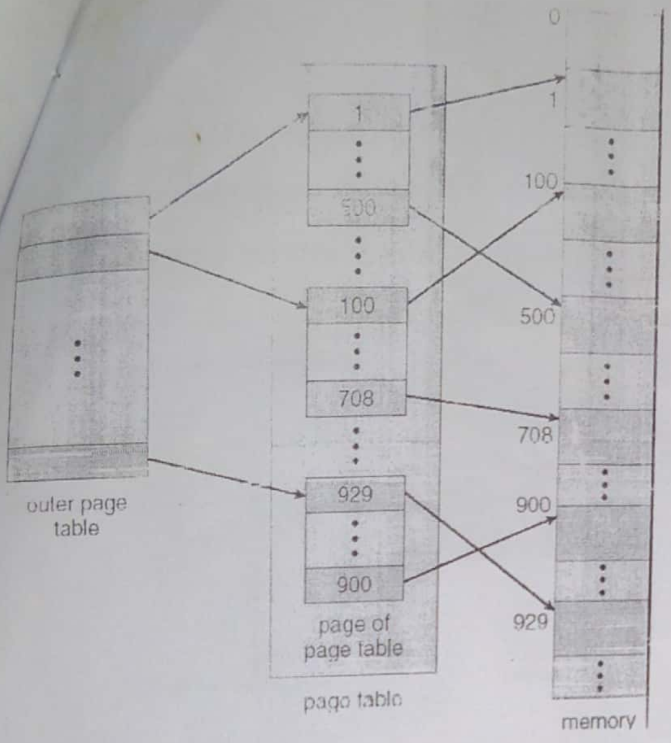


- ➔ P1 is an index into the outer page table, and p2 is the displacement within the page of the outer page table.

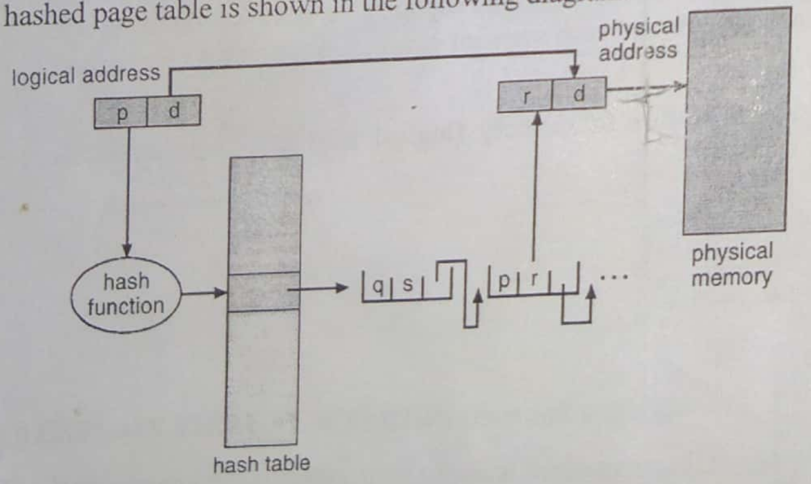
Address-Translation Scheme:

Address-translation scheme for a two-level 32-bit paging architecture and is shown in the following diagram:

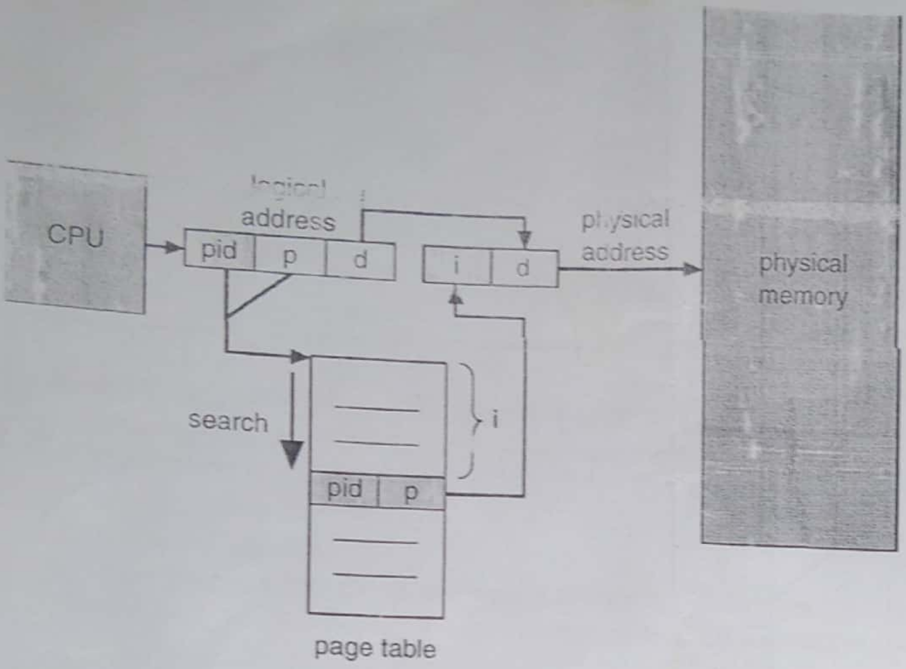




2) Hashed Page Tables:
A common approach for handling address spaces larger than 32 bits is to use a *hashedpage table*, with the hash value being the virtual-page number. Each entry in the hash table contains a linked list of elements that hash to the same location (to handle collisions). Each element consists of 3 fields:
a) the virtual page number
b) the value of the mapped page frame and
c) a pointer to the next element in the linked list.
This method stores page table information in hash table for faster access & less wastage of memory. The hashed page table is shown in the following diagram:



3) Inverted Page Tables:
The modern computer allows a page table for each and every process. The entire page table is to be stored in the contiguous memory locations in the memory. Entry consists of the virtual address of the page stored in that real memory location, with information about the process that owns that page. Decreases memory needed to store each page table, but increases time needed to search the table when a page reference occurs.
It is better to use hash table to limit the search to one — or at most a few — page-table entries. This is shown in the following diagram:

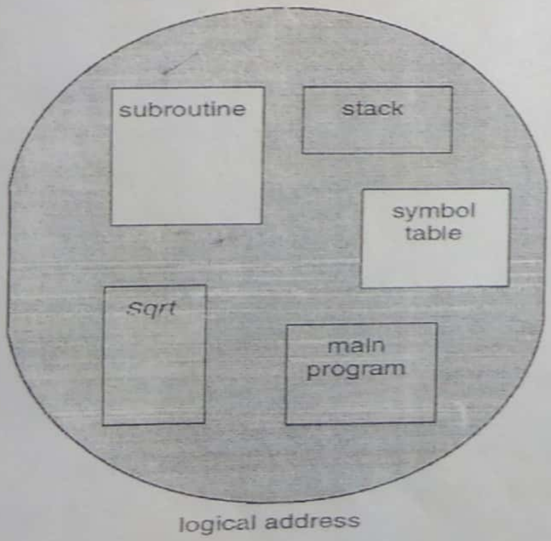


5 Q.WRITE ABOUT SEGMENTATION?

Segmentation is a memory-management scheme that supports user view of memory. A program is a collection of segments. A segment is a logical unit such as main program, procedure, function, method, object, local variables, global variables, common block, stack, symbol table and arrays.

In the segmentation the logical memory can be divided into fixed sized blocks these blocks are called as segments. Segmentation is a memory management functionality that supports the user view of memory. In this, the logical memory is divided into segments and each segment consists of segments number and offset

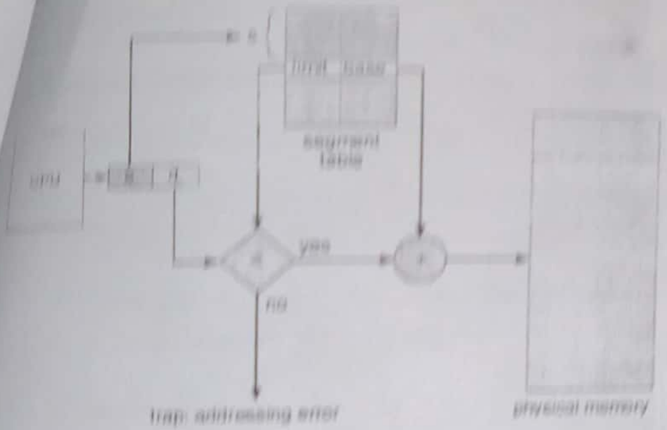
For example the segments in the user view of memory (logical address)is represented as follows.



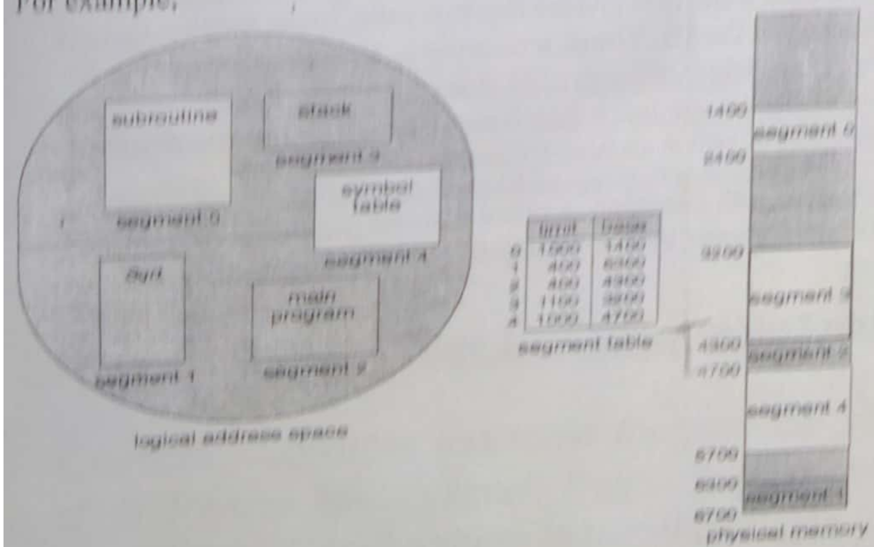
The segment table is used to mapping between the logical memory and physical memory the segment table consists of two columns [illegible] – contains the starting physical address where the segments reside in memory.

[illegible] – specifies the length of the segment

[illegible] mapping is represented as follows



For example,

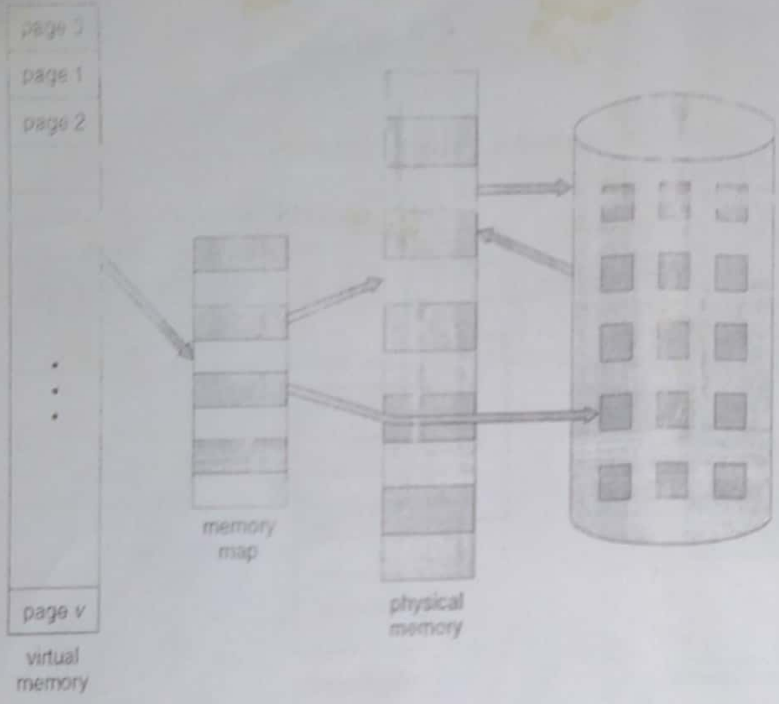


	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

(Q) EXPLAIN VIRTUAL MEMORY (demand paging):-

The logical memory is greater than physical memory is called virtual memory in the logical memory it is divided into pages and the physical memory is divided into frames the mapping between logical memory and physical memory is with the help of page table the page table contains page number and frame number

In the demand pagging we have to use backing store the functionality of backing store is operate swapin and swapout principles i.e;



In the memory management an address generated by the CPU is divided into two parts. page number and off set and an address is found in the main memory is divided into two parts. frame number and off set .The size of the frame and page depends on the OS. Virtual memory is a technique that allows the execution of processes that may not be completely in memory. The size of the logical program can be larger than physical memory. If the no. of pages are more than frames the remaining pages will be loaded into backing store this is the procedure of virtual memory technique.

The virtual memory supports paging system with swapping is called demand paging the idea behind demand paging is when we want to execute a process, then only swap that process into main memory by demand so it is called as demand paging

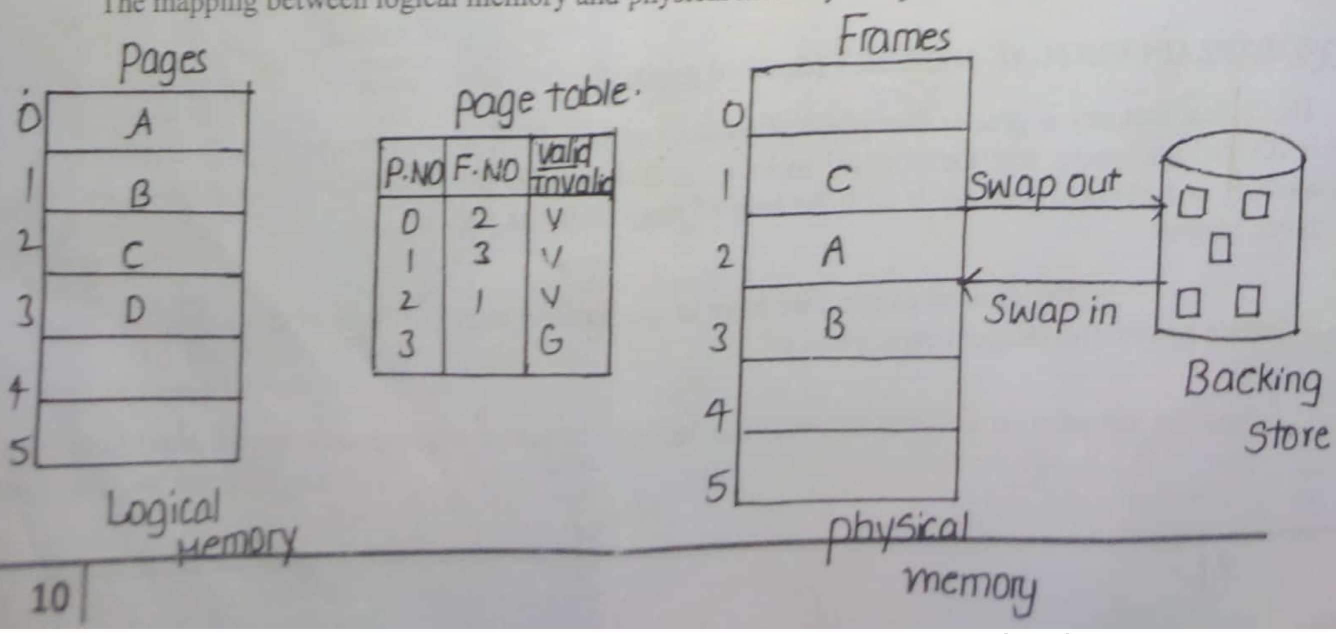
In this demand paging the page contains 3 coloums viz,.

1.page number

2.frame number

3.valid/invalid bit

The mapping between logical memory and physical memory is represented as follows



P.NO	F.NO	valid Invalid
0	2	V
1	3	V
2	1	V
3		G

Sol 3

NE PAGE FAULT? WRITE ABOUT PAGE REPLACEMENT RITHM[BELADYS ALGORITHM]?

In a page table the invalid bit refers to the page fault ie ;

corresponding page is not in the physical memory(frame)

There are various types of page replacement algorithms to support in the o.s they are

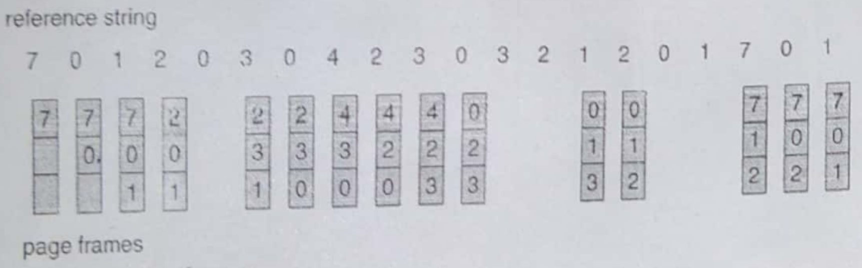
1.FIFO(first in first out)

2.Optimal algorithm

3.LRU(least recently used)algorithm

FIFO:-it is the simplest page replacement algorithm we replace the page at the head of the queue .when a page is brought into the memory we can insert at the tail of the queue

For example, Let us assume that the reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 2, 1, 2, 0, 1, 7, 0, 1 for a memory with 3 frames.

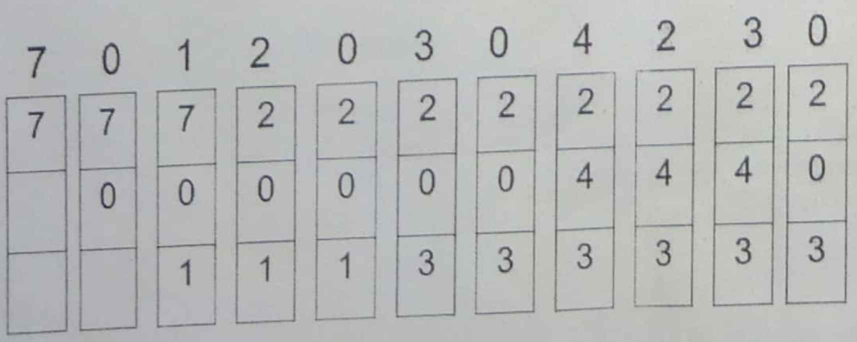


OPTIMAL:- An optimal page replacement algorithm has the lowest page fault rate of all algorithms it follows replace the page that will not be used for the largest period of time.

For example it can be used represented as follows

Optimal Page Replacement

Example:- Consider the main memory consists of 3 frames for Optimal Page Replacement algorithm and input pattern of memory pages.



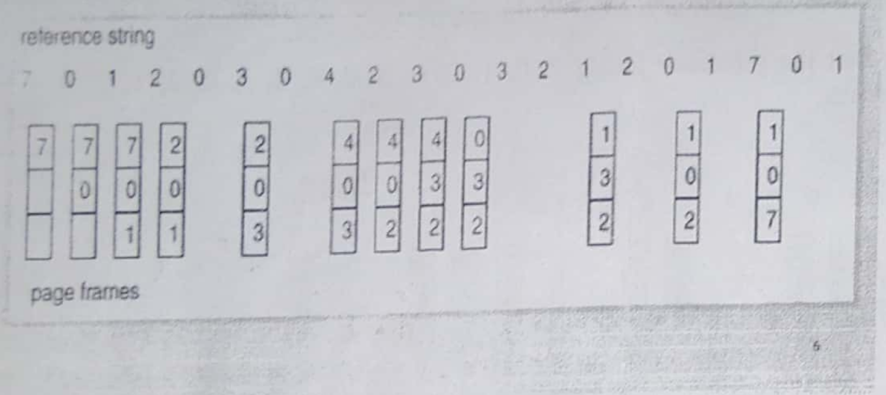
1 5

LRU:-it associates with each page.the time of the pages last used when a page must be replaced lru chooses that the page of the time.the difference between optimal and LRU is working forward in the optimal where as backward in the LRU

For example it can be represented as follows

LRU Page Replacement

- The LRU algorithm produces 12 page faults for the reference string 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1



We observe that an optimal algorithm is best replacement algorithm in the o.s the page fault rate may increased then the no.of allocated frames are also increases.

5Q) **WRITE ABOUT FRAGMENTATION?**

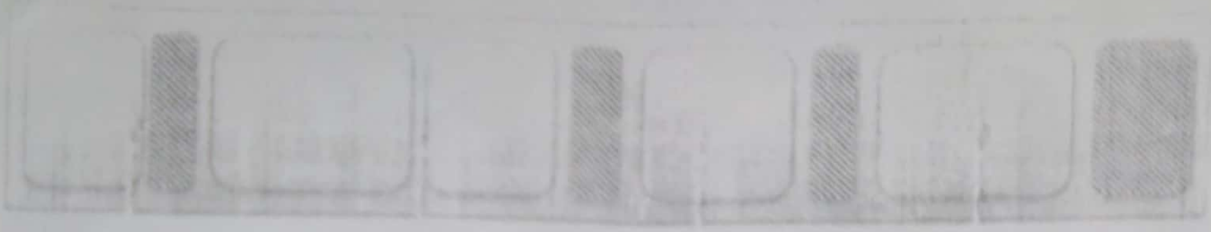
- ➔ As processes are loaded and removed from memory, the free memory space is broken into little pieces.
- ➔ It happens after sometimes that processes cannot be allocated to memory blocks considering their small size and memory blocks remains unused.
- ➔ This problem is known as Fragmentation.

Fragmentation is of two types −

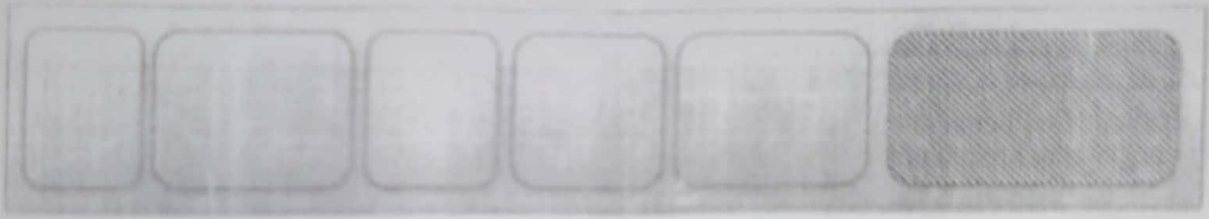
S.N.	Fragmentation & Description
1	External fragmentation Total memory space is enough to satisfy a request or to reside a process in it, but it is not contiguous, so it cannot be used.
2	Internal fragmentation Memory block assigned to process is bigger. Some portion of memory is left unused, as it cannot be used by another process.

→ The following diagram shows how fragmentation can cause waste of memory and a compaction technique can be used to create more free memory out of fragmented memory.

Fragmented memory before compaction



Memory after compaction



→ External fragmentation can be reduced by compaction or shuffle memory contents to place all free memory together in one large block.
→ To make compaction feasible, relocation should be dynamic.
→ The internal fragmentation can be reduced by effectively assigning the smallest partition but large enough for the process.

Q) EXPLAIN ABOUT ALLOCATION OF FRAMES?

Each process needs a ***minimum*** number of pages

Example: IBM 370 – 6 pages to handle SS (storage to storage) MOVE instruction:

- The MVC instruction is 6 bytes long (more than one word), might span 2 pages
- 2 pages to handle characters at the *from* address (source)
- 2 pages to handle characters at the *to* address (target)

Two major allocation schemes

fixed allocation: it is also called as Equal allocation

For example, if there are 100 frames and 5 processes, give each process 20 frames.

s_i = size of process p_i

$S = \sum s_i$

m = total number of frames

a_i = allocation for $p_i = \frac{s_i}{S} \times m$

$m = 64$

$s_1 = 10$

$s_2 = 127$

$a_1 = \frac{10}{137} \times 64 \approx 5$

$a_2 = \frac{127}{137} \times 64 \approx 59$

priority allocation

Use a proportional allocation scheme using priorities rather than size

Proportional allocation – Allocate according to the size of process

If process P_i generates a page fault, [illegible]

select for replacement one of its frames

select for replacement a frame from a process with lower priority number

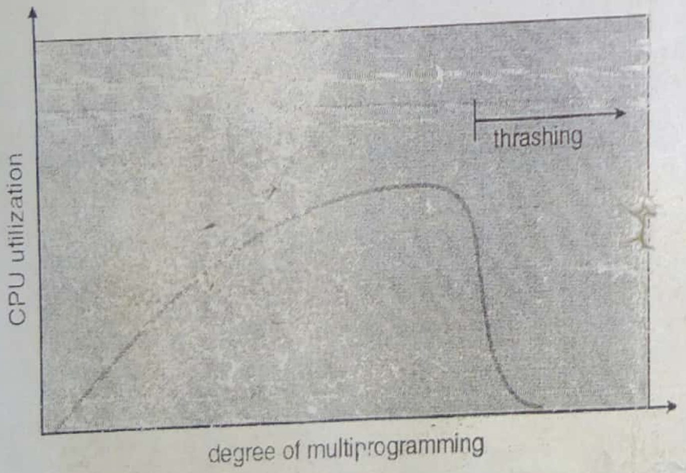
Q. EXPLAIN ABOUT THRASHING?

Thrashing is defined as a situation in which a program cause page faults per every few instructions.

If the degree of multi programming increases CPU utilization also increases. In such a case the CPU utilization drops sharply. It can be represented as follows.

Thrashing ≡ a process is busy swapping pages in and out

Initially when the CPU utilization is low, the process scheduling mechanism, to increase the level of multiprogramming loads multiple processes into the memory at the same time, allocating a limited amount of frames to each process. As the memory fills up, process starts to spend a lot of time for the required pages to be swapped in, again leading to low CPU utilization because most of the proccesses are waiting for pages. Hence the scheduler loads more processes to increase CPU utilization, as this continues at a point of time the complete system comes to a stop.



To prevent thrashing we must provide processes with as many frames as they really need "right now".