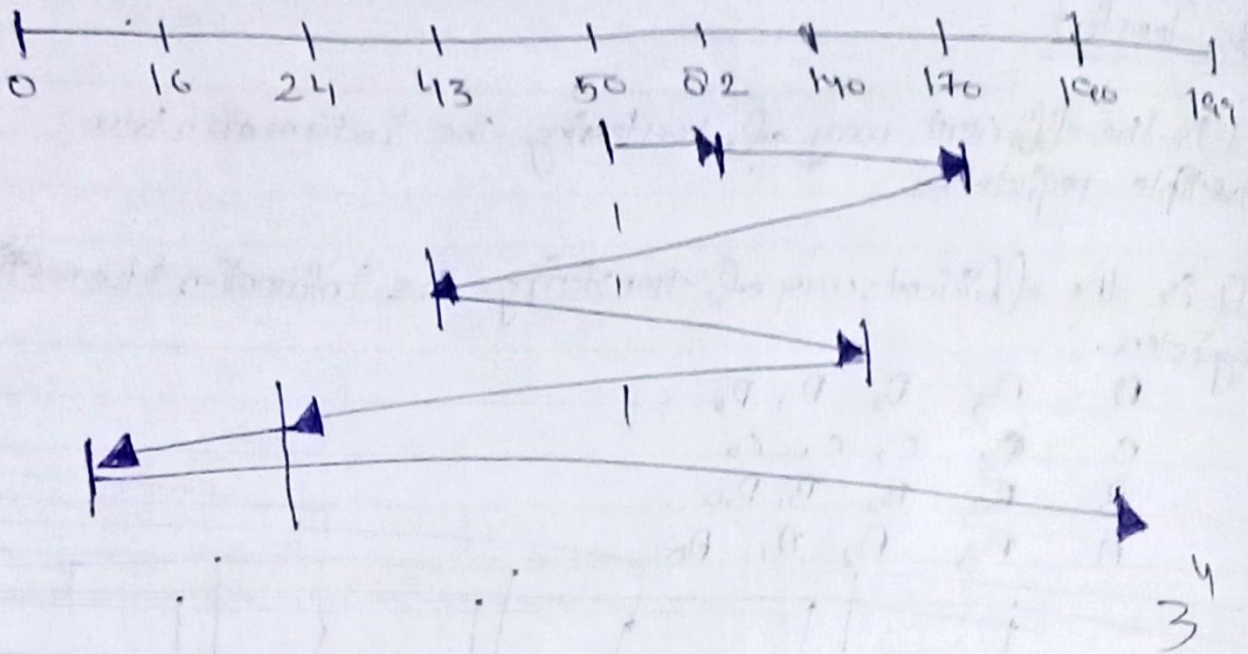
General Register Organization :-

Disk Scheduling

1) FCFS :- A disk Queue with I/O request for the blocks on cylinders

82, 170, 43, 140, 24, 16, 190. There are Total 200 tracks & the current head position is 50. Calculate total head movement by arm. If, one track takes 5 ms then total head movement time is.



Total head movement = (82-50) + (170-82) + (43-170)
+ (140-43) + (24-140) + (16-24)
+ (190-16)
= 32 + 88 + 127 + 97 + 116 + 8 + 174
= 746

Name of Experiment..... Date.....

Experiment No..... Experiment Result..... Page No.

Total time taken = 642 x 5

= 3210 ms

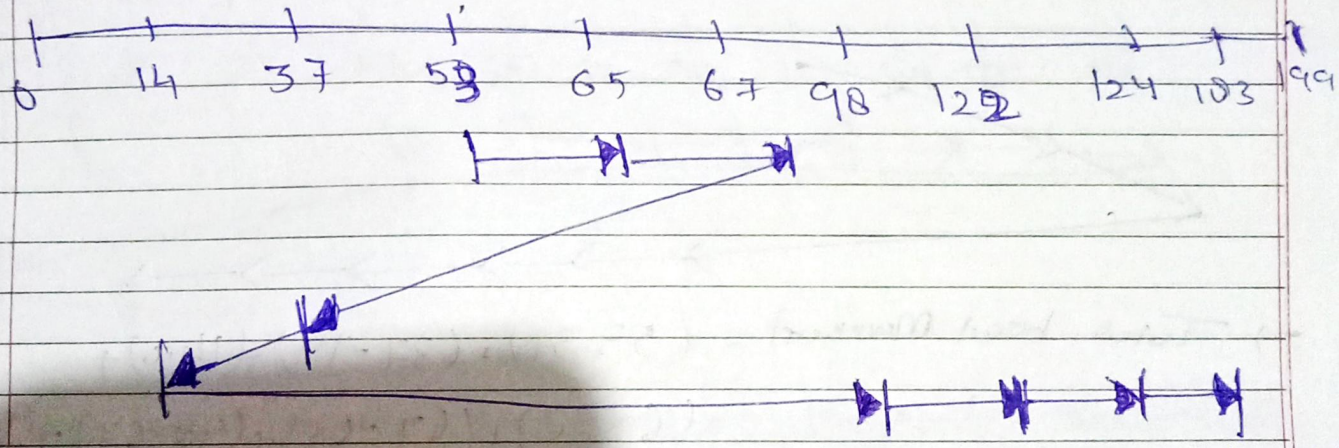
Disadvantage: It doesn't optimise the seek time.

2)- Shortest Seek time first:- (Difference of after & before next time)

-> The I/O request values are.

98, 183, 37, 122, 14, 124, 65, 67 current head positions

=>



=> Total head movement = (65-53) + (67-65) + (98-67) + (37-14) + (183-98) + (122-98) + (124-122) + (183-124)

= 12 + 2 + 30 + 23 + 84 + 24 + 2 + 59

= 236

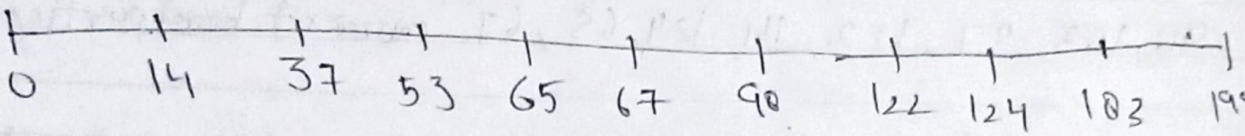
Teacher's Signature:

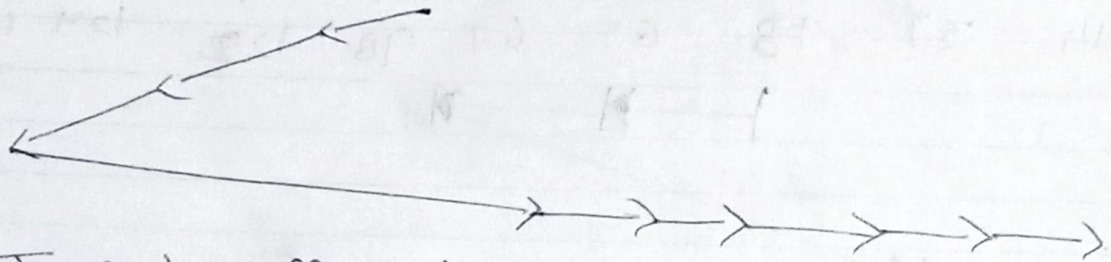
Disadvantage: Before moving on next step we have to calculate seek time in advance!

⇒ Scan Disk Scheduling Algorithm:-

⇒ The disk arm moves in a particular direction and grants the request coming in its path and after reaching the end of the disk it reverse its direction and again grant the request arriving in its path. This also works like an elevator so it is called elevator algorithm.

Q)- 98, 183, 37, 122, 14, 124, 65, 67





⇒ Total head Movement = (53-37) + (37-14) + (14-0) + (65-0) + (67-65) + (98-67) + (122-98) + (124-122) + (183-124) ⇒ 236.

Advantage:- It provide high throughput value and also its average response time is low.

Disadvantage:- Longer waiting time for request for locations close to current head but opting the another direction.

Name of Experiment..... Date.....

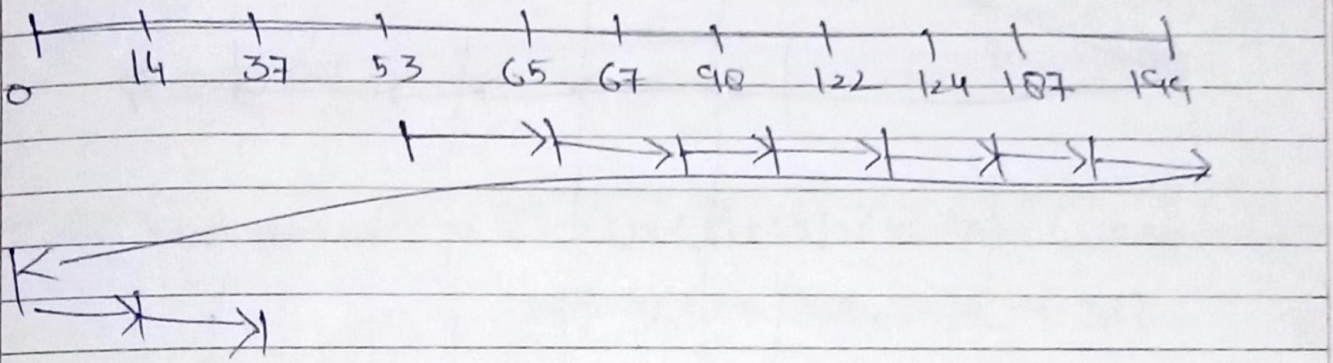
Experiment No..... Experiment Result..... Page No.

4)- Circular Scan Disk Scheduling:

Circular Scan is a variant of scan or elevator algorithm designed to provide a more uniform waiting time. circular scan moves the head from one end of the disk to the other granting request along the way. When the head reaches the other end immediately returns to the beginning of the disk without granting any request on the return path.

Q)- 98, 183, 37, 122, 14, 124, 65, 67, current position of head is 53.

=> forward direction (.)



Total head movement = (65-53) + (67-65) + (98-67) + (122-98) + (124-122) + (183-124) + (199-183) + (199-0) + (14-0) + (37-14)

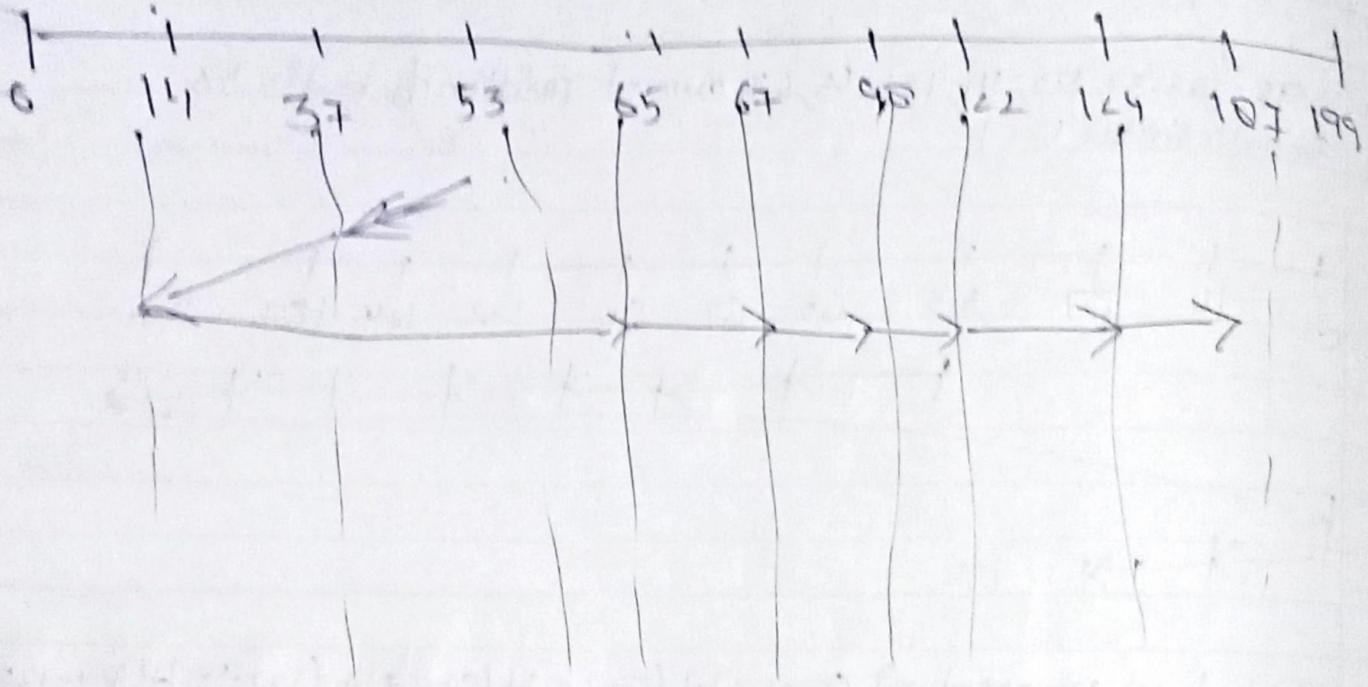
= 382 Ans

Disadvantage:- There is a wastage of path movement because we are moving from beginning to the end without granting any request.

Teacher's Signature : _____

5)- Look Disk Scheduling Algorithm:-

⇒ It is similar to scan or elevator algorithm ~~effect~~ except the direction that the disk arm inspite of going to the end of the disk goes only to the last request to be granted infront of the head and then reverses its direction from that particular track.



Head = 16+23+51+12+31+24+2+59

= 208 Ans.

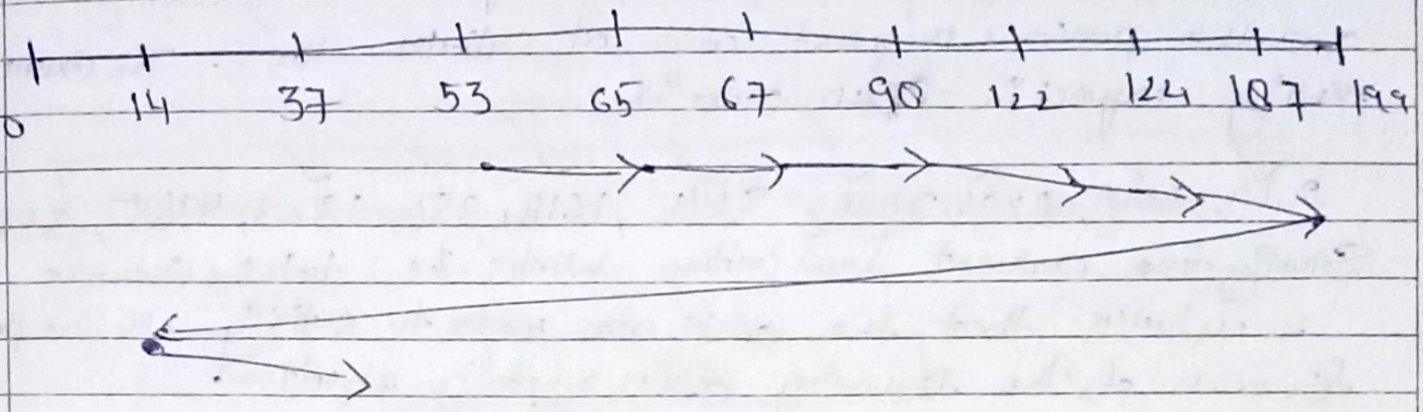
Advantage :- It prevents the ~~extremely~~ extra delay which occured due to unnecessary traversel to the end of the disk.

Name of Experiment..... Date.....

Experiment No..... Experiment Result..... Page No.

6- Circular Look Disk scheduling Algorithm :

→ It is similar to circular scan algorithm, the disk arm inspite of going to the end goes only to the last request to be granted infront of the head and then from their goes to the other end's last request.



Total head movement = (65-53) + (67-65) + (90-67) + (122-90) + (124-122) + (187-124) + (187-14) + (14-37)

= 322 Ans

Advantage :- It also presents the extra delay which occured due to unnecessary traversel.

⇒ 30/11/2021

Operating System

Q) - I/O request are.
160, 175, 51, 133, 8, 140, 73, 77 head → 63
Total cylinders are = 200

FCFS, SSTF, SCAN(left), LOOK(right), C-LOOK (Right) C-SCAN(Right)

Q) - Suppose a disk drive has 5000 cylinders numbered from 0 to 4999. The drive is currently giving a request at cylinder 2150 and the previous request was at cylinder 1805. The queue of pending request in FIFO order is

2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681.

Starting from current head position, what is the total distance in cylinders, that the disk arm moves to satisfy all the pending for each of the following disk scheduling algorithms.

FCFS, SSTF, SCAN(left), C-SCAN(Right), LOOK(Right), C-LOOK(Right)

Memory Management Strategies

⇒ Main memory and the registers build into the processor itself are the only general purpose storage that the CPU can access directly

Limit Register :- It specifies the size of the range.

from the above diagram for ex Base register holds 300040 and limit register is 120900 than the program can legally access all addresses from 300040 through 420939.

2/12/2024

Memory Allocation.

21/12/2021

Memory allocation is of two types

Contiguous	Non-contiguous
1) - It allocates consecutive blocks of memory to a process.	1) - It allocates separate blocks of memory to a process
2) - It does not have the overhead of address translation while execution of a process.	2) - It has overhead of address translation while execution of a process.
3) - A process executes faster in contiguous memory allocation.	3) - A process executes a quite to slower
4) - The memory space must be divided into the fixed size partition and each partition is allocated to a single process only.	4) - Divide the process into several blocks & place them in different parts of the memory according to the availability of memory space.

Fixed Size Partition :- It has fixed size and usually cannot be easily expended or shrunk.

Dynamic Size Partition :- It is applicable when an executing program request that the O.S. give it a block of main memory. Memory may be returned whenever it is no longer needed.

Name of Experiment..... Date.....

Experiment No..... Experiment Result..... Page No.

Initially all memory is available for user processes and is considered one large block of available memory. The concerned of dynamic storage allocation is how to satisfy a request of size n from a list of free spaces. There are various techniques like:

i)- first fit
ii)- Best fit
iii)- worst fit

first fit allocation.

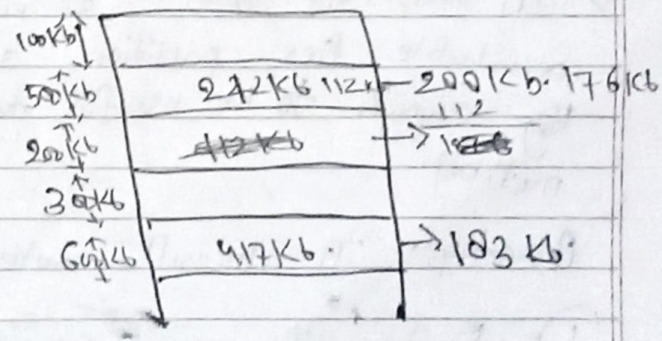
In this approach allocation of any block can be done in first free partition.

Advantage:- fastest algorithm because it searches as little as possible.

Disadvantage:- The remaining unused memory area left after allocation become waste.

Q-1) Given 5 memory partitions of 100Kb, 500Kb, 200Kb, 300Kb, 600Kb in order. How could the first fit algorithm place processes of 212Kb, 417Kb, 112Kb, 426Kb in order.

We could not allocate P4 in this memory allocation block, memory block due to lack of memory.



Teacher's Signature :

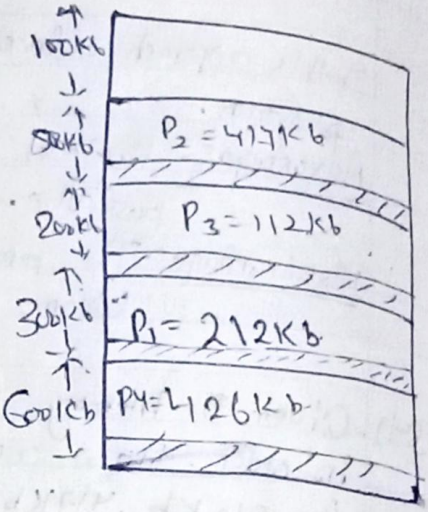
Best fit Allocation

Best fit Allocation deals with allocating the smallest free partition which full fills the requirement of the requesting process.

Advantage:- Memory utilisation is better

Disadvantage:- It is quite slower

Q)- 100Kb, 500Kb, 200Kb, 300Kb, 600Kb
912Kb, 417Kb, 112Kb, 426Kb.



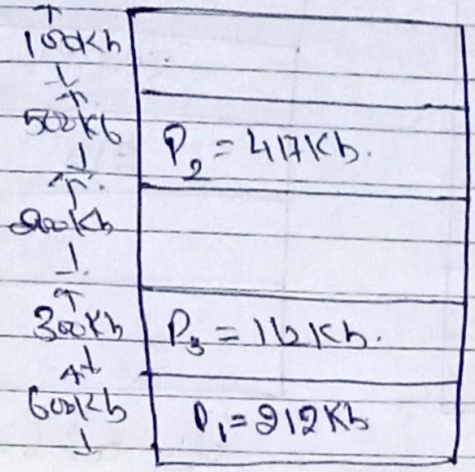
Worst fit Allocation

⇒ In worst fit approach we will allocate any process to largest available free partition so that the portion left will be big enough to be useful again. It is the reverse of best fit method.

Advantage:- It reduces the rate of production of small gaps.

Disadvantage:- If a process requiring larger memory at the later stage then we ~~could not accomodate~~ because it has already occupied by some another request.

100Kb, 500Kb, 200Kb, 300Kb, 600Kb
212Kb, 417Kb, 112Kb, 426Kb.

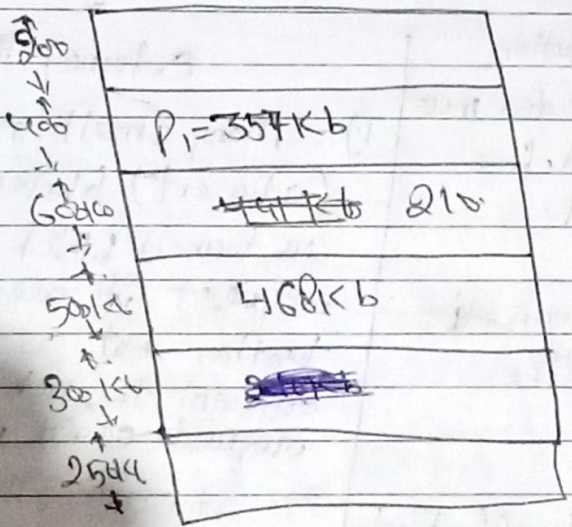


We could not allocate P4 according to worst fit allocation method.

Ans

Q) Given six memory partitions 200Kb, 400Kb, 600Kb, 500Kb, 300K, 250Kb process is 357Kb, 210Kb, 468Kb, 491Kb. Allocate the above request according to best fit worst fit & first fit allocation method.

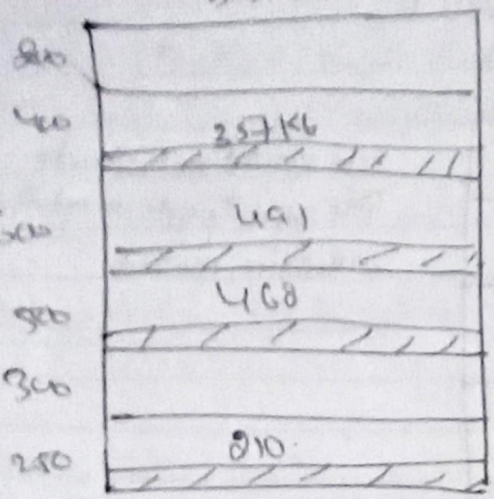
Best fit



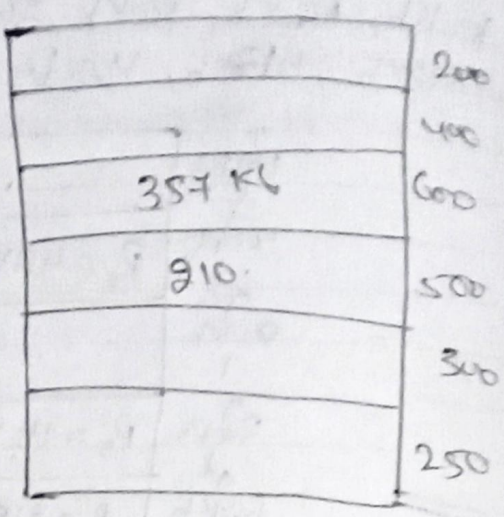
P4 will not allocate.

Teacher's Signature :

Best fit



Worst fit.



only a & b will allocate.

Ans

Internal fragmentation: only one block wastage

fragment Memory = wastage partition (wastage memory).

Ans

External fragmentation = Total wastage of memory block.

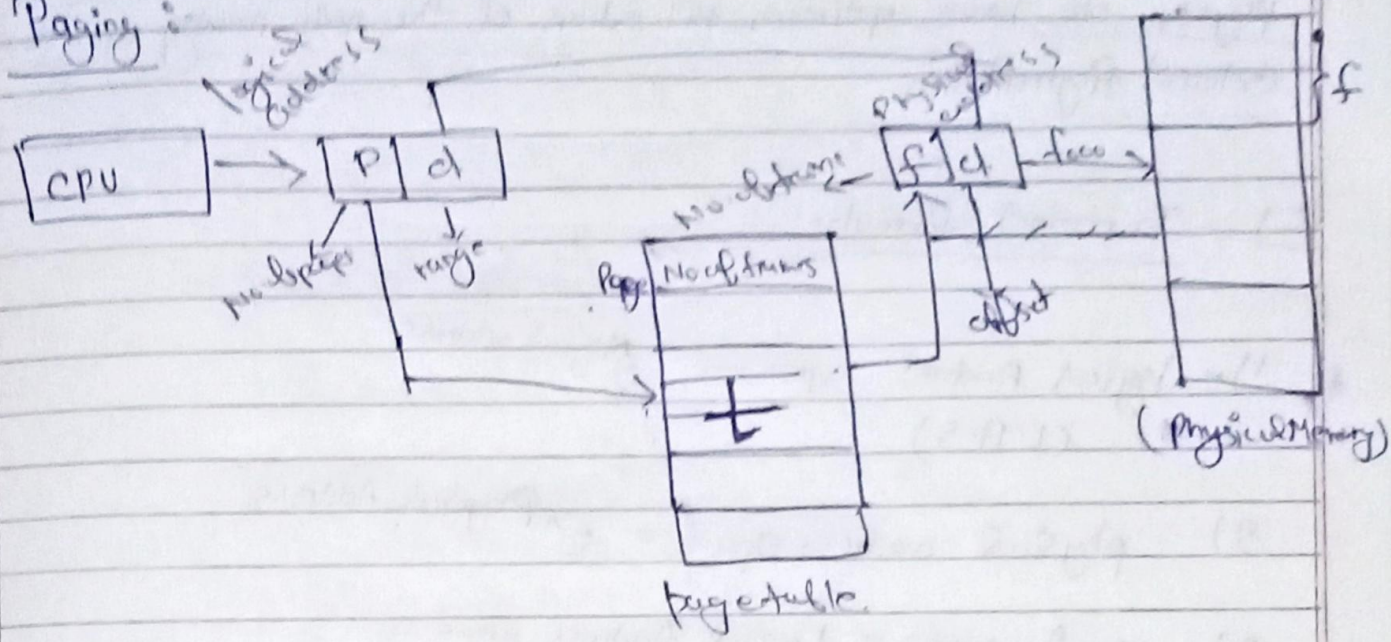
Internal fragmentation	External fragmentation.
1)- When a process is allocated more memory than required. few space is left unused.	1)- Many small non contiguous (adjacent) blocks of unused spaces are formed which can serve a new request if all of them are put together but as they are not adjacent to each other a new request could not be served.
2)- It occurs when memory is ~~divi~~ divided into fixed size partitions.	2)- It occurs when memory is divided into variable size partition.
3)- It can be reduced by effectively assigning the smallest partition but large enough for the process.	3)- It can be avoid by compaction & other techniques.

Name of Experiment..... Date 9/12/2021 Page No.

Experiment No..... Experiment Result.....

Note: Compaction :- External fragmentation can be reduced by compaction or suffle memory contents to place all free memory together in one large block.

=> Paging :



-> Paging is a memory management technique in which address space is broken into blocks of same size called pages.

1)- Page offset :- Page address is called logical address and is represented by page number (P) and page offset (d)

offset = range.

P	d

2)- frame address is called physical address and is represented frame number (f) and the frame offset (d).

f	d

Teacher's Signature :

→ Page Table:- A data structure called page table and is used to keep track of relation b/w a page of a process and frame in physical memory.

→ Main memory is divided into small fix size block of physical memory called frames and the size of the frame is kept similar that of pages to have optimum utilization of the main memory and to avoid external fragmentation.

- Important formulae:-

1)- logical Address space = $2^{\text{logical address}}$
(L.A.S)

2)- physical address space = $2^{\text{Physical Address}}$

3)- No. of pages = $\frac{\text{Logical Address space}}{\text{Page size}}$

4)- No. of frames = $\frac{\text{Physical Address space}}{\text{frame size}}$

Note:- Number of entries in the page table is equal to no. of pages.

Q1) - L A = 31 bits.

LAS = 2^{31} =

2^{10} = 1K.
2^{20} = 1M.
2^{30} = Giga = 1G
2^{40} = 1T

Q) - If Logical address space = 128 megawords.

L·A·S = 2^{128}

log(LAS) = 128 log(2)

$128 = 2^{log \cdot A}$

$\log(2^7) = \log(2)^{log A}$

Logical Address = 7 M words.

= $2^7 2^{20}$

= 2^7.

$\log 2^7$

L·A· = 27

Q) - If logical address = 7bits, physical address = 6bits
Page size = 8 bytes words

Sol).

No. of pages = $\frac{2^7}{2^3} = 2^4 = 16$ = 27

No. of frame = $\frac{2^6}{2^3} = 2^3 = 8$

No. of entries in the page table = 16.
Size of a page table = 8.

Q) -

Size of a page table = 16 × 8.
= 128.

Q) - If logical address space = 4gb = 8 physical address space = 64mb = page size = 4kb
→ calculate No. of pages. No. of frame. No. of entries in the page table.

- No of pages = $\frac{2^{4}}{2^{4}} =$

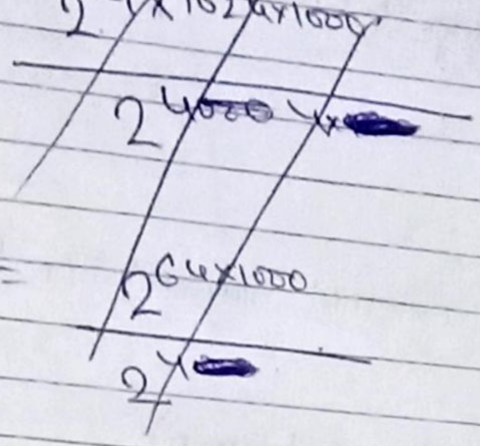
No. of frames = $\frac{2^{64}}{2^{4}}$

1kb = 1000
1000 = 1kb
1mb = 1000kb
1gb = 1024mb.

Experiment No. Date

Experiment Result Page No.

- No of bits = $2^{4} \times 1024 \times 1000$



$= 2^{6} \times 1000$

2^{4}

$4 \times 1000 = 4000$

4096000

$\Rightarrow$ LAS = 4GB

$\Rightarrow$ No of pages $= \frac{LAS}{P.S} = \frac{2^{2} \times 2^{30}}{2^{2} \times 2^{10}} = \frac{2^{32}}{2^{12}} = 2^{20}$

$\Rightarrow$ No of frame $= \frac{PAS}{2^{2}} = \frac{2^{6} \times 2^{20}}{2^{2} \times 2^{20}} = 2^{14}$

$\Rightarrow$ No of entries $= 2^{20}$

$\Rightarrow$ No of Size table $= 2^{34}$ ← Ans

No of bits used to write page = 14 bits required to write 2^{14} page.

Teacher's Signature :

Q) - P.A.S = 2GB and p.size = 512 bytes. calculate no. of pages & page table size where size of page table entry = 4bytes

- Advantages :-

i) - Paging reduces external fragmentation but still suffer from internal fragmentation.

ii) - Paging is simple to implement

iii) - Due to Equal size of the pages & frames swapping becomes very easy

Disadvantages

iv) - Page table requires extra memory space so may not be good for a system having small Ram.

09/12/2021

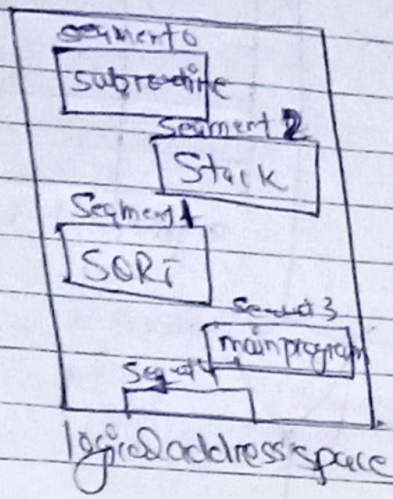
Segmentation.

It is a memory management technique in which memory is divided into variable size chunks / blocks. Which can be allocated to processes. Each chunk is called segment.

Name of Experiment ..... Date .....

Experiment No. 1 Experiment Result ..... Page No.

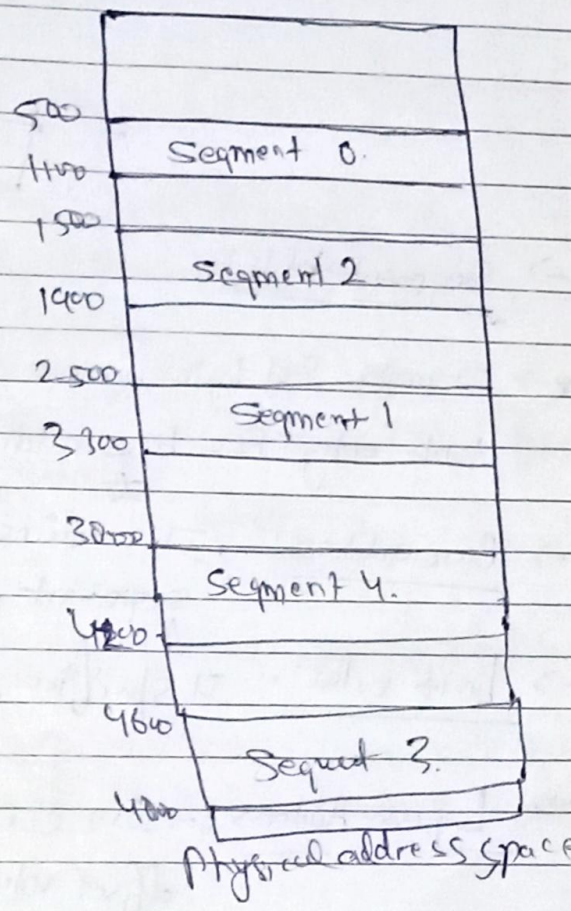
for ex:-



Logical address space

Segment NO	Base address	limit
0	500	600
1	2500	800
2	1500	400
3	4600	200
4	3300	400

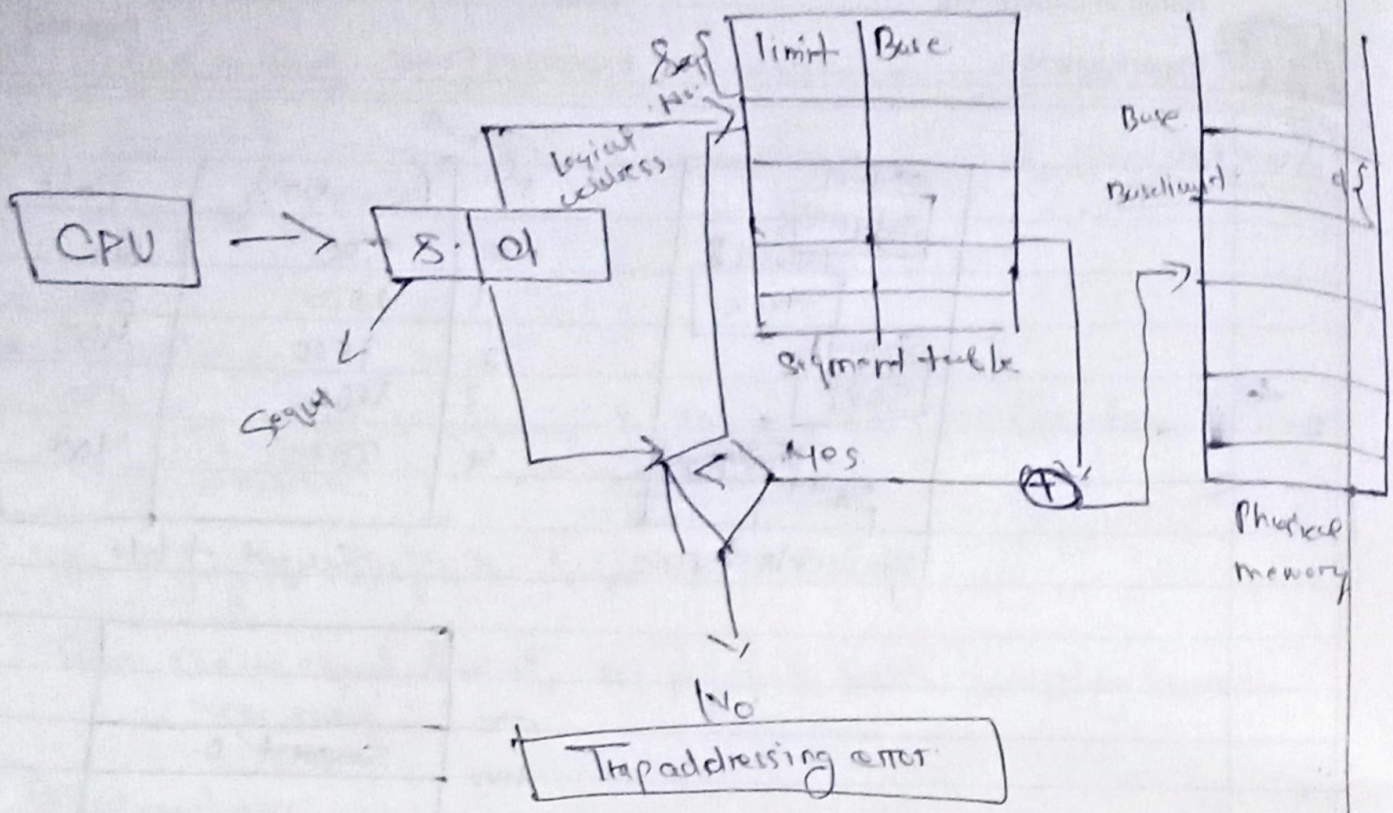
Segment table



Physical address space

Teacher's Signature : .....

Hardware Structure



-> Segment table :-

-> It maps 2-d logical address into one dimensional physical address. Each table entry has base address and limit value

-> Base address :- It contains the starting physical address where the segment reside in memory.

-> Limit value :- It specifies the length of the segment.

-> Logical Address :- It consists of two parts segment number(s) & offset value(d).

The segment number is used as an index to the segment table.

The offset value (d) of the logical address must be ~~between zero~~ $0 < d < limit$ ~~and the segment limit~~

$0 < d < limit$.

Name of Experiment ..... Date .....

Experiment No ..... Experiment Result ..... Page No

→ Advantage :-

i). There is no internal fragmentation

Disadvantage :-

ii)- Might be it suffers from external fragmentation.

Diff Difference b/w Paging & Segmentation

Paging	Segmentation
i)- A page is of fixed block size.	i)- A segment is of variable size.
ii)- Paging may lead to internal fragmentation.	ii)- Segmentation may lead to external fragmentation.
iii)- The user specified address is divided by CPU into a page No(p) and offset(d). p/d	iii)- The user specifies each address by two quantities segment number (S) & offset (segment limit) S/d
iv)- Hardware defines the page size.	iv)- The segment size is specified by the user.
v)- Paging involves a page table that contains B.A. of each page. corresponding frame numbers are also there.	v)- Segmentation involves Segment table that contains B.A, limit value & segment number.

Teacher's Signature : _____

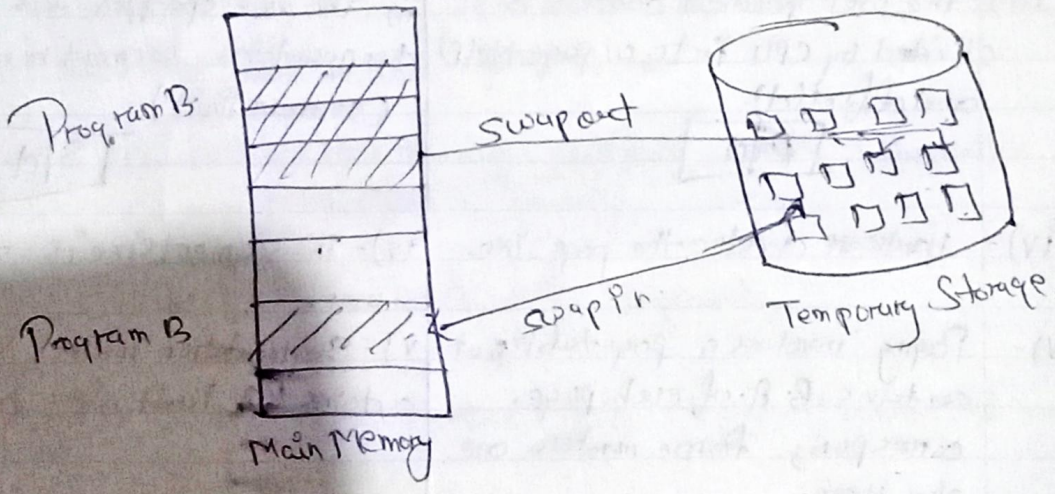
Virtual Memory Management

→ Virtual Memory makes the task of programming much easier because the programmer no longer needs to worry about the amount of physical memory available.

⇒

⇒ **1) Demand Paging :-**

The process of loading the page into memory on demand (whenever page fault occurs) is known as demand paging.



Advantages :-

1) - More efficient use of memory.
2) - There is no limit on degree of multiprogramming.

Name of Experiment

Experiment No. Date

Experiment Result Page No.

-> Disadvantage :-

-> The amount of processor overhead for handling page interrupts are greater than in case of simple page management technique.

-> Page Replacement :-

In an operating system that use paging for memory management, page replacement algorithm are needed to decide which page needed to replace when new page comes in.

Page fault :-

A page fault is a type of ~~interrupted~~ interrupt raised by the h/w. when a running program access a memory page i.e. mapped into the virtual address space but not loaded in physical memory.

Page Replacement Algorithms :-

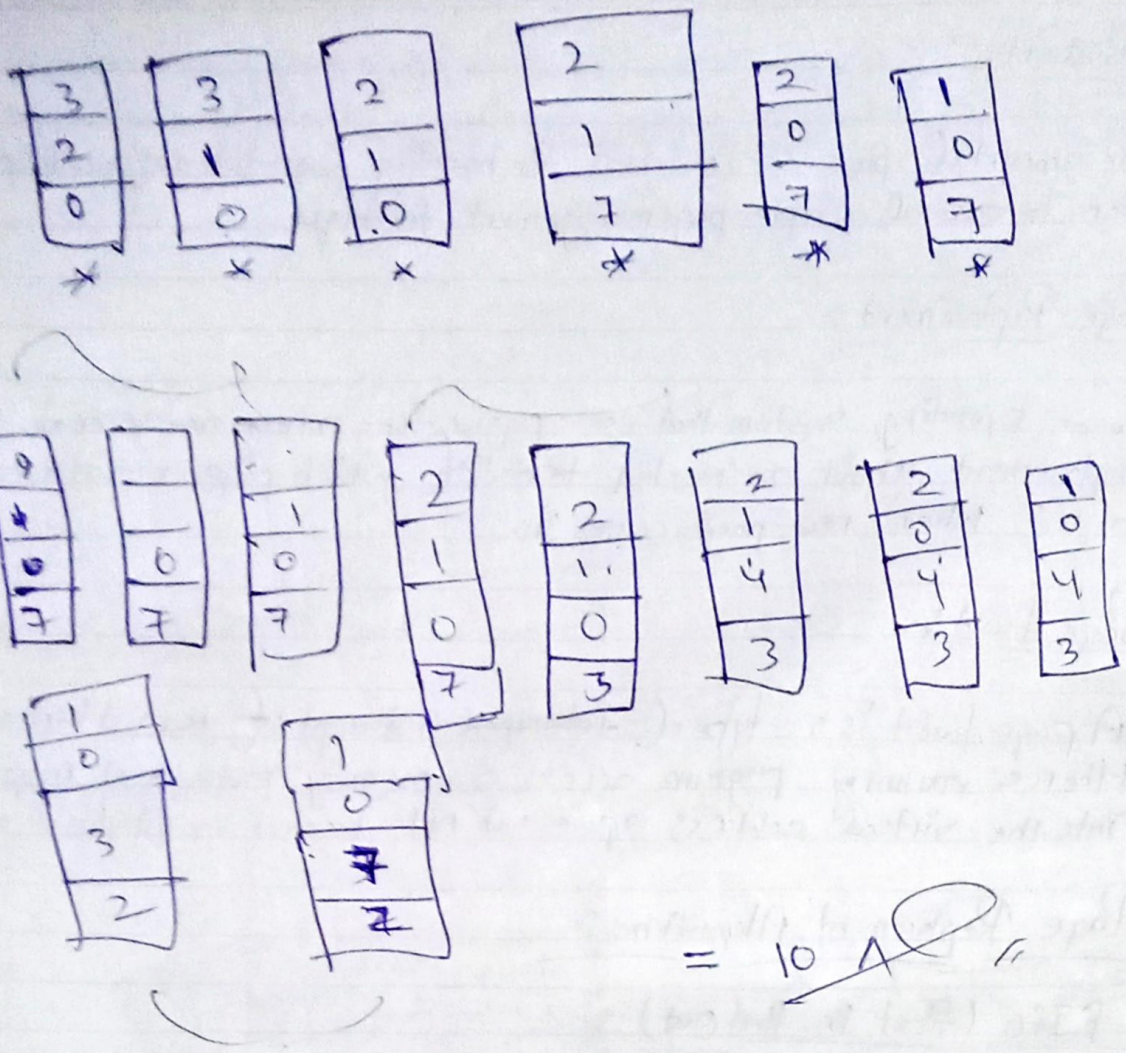
1) FIFO (first in first out) :-

(reference)

Q) Given string (positions) ~~7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1~~ 7,0,1. frame size = 3. calculate no. of page fault.

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

Teacher's Signature :



= 10

Q) - String :- 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5.

frame size = 3 = 9 page fault = 9.

frame siz = 4 = 10 page fault = 10

Note :- from Question No. 2 if frame size = 3 then Page fault = 9.
if frame size = 4 then page fault = 10 then this kind of problem is known as Belady's anomaly means that in general situation if we will increase frame size then the page fault must be decreases.

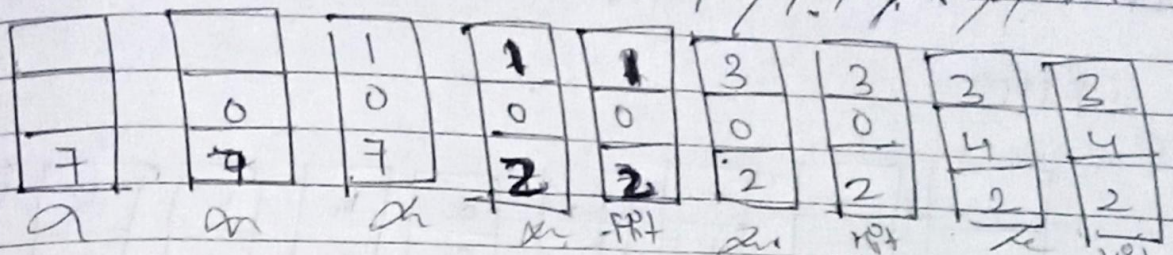
Name of Experiment..... Date.....

Experiment No..... Experiment Result..... Page No.

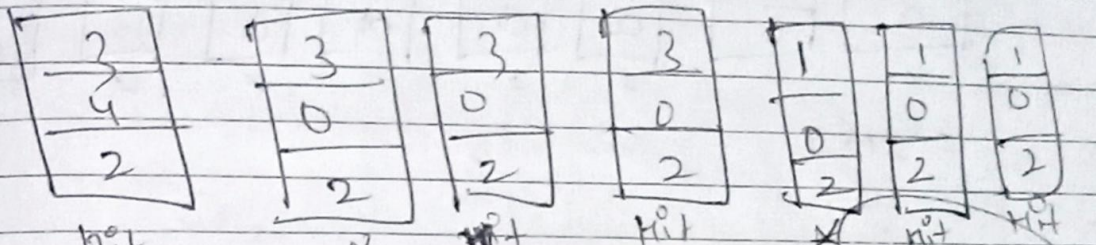
=> optimal page replacement (most recently used) → forward

~~7 0 1~~

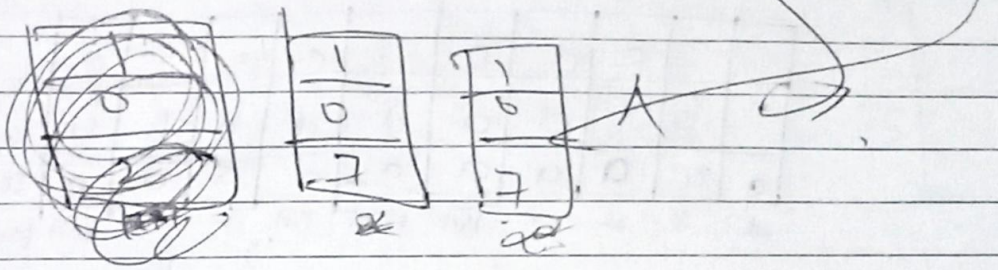
frame size = 3 ~~7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1~~



a a a a fit a rpt a hit



hit a hit hit x hit hit



a a

Advantage of optimal:-

Replace the page that will not be used for longest period of time.

Teacher's Signature :

Question:

reference string

a, b, c, d, e, a, d, b, e, b, a, b, c, d

Frame size = 3

FIFO

optimal

LRU

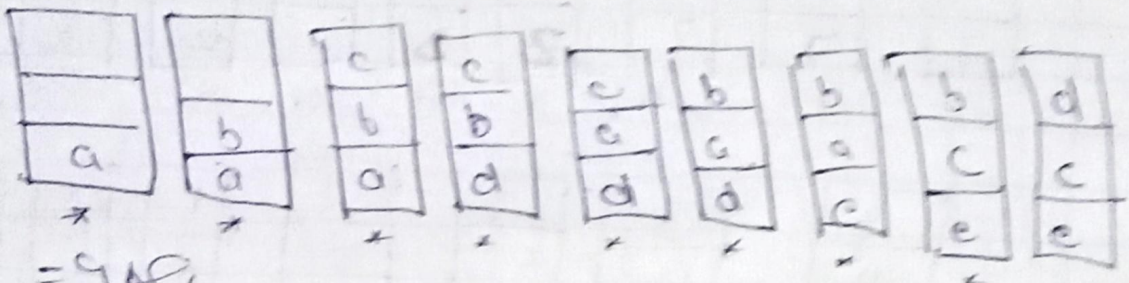
Frame size = 4

FIFO

optimal

LRU

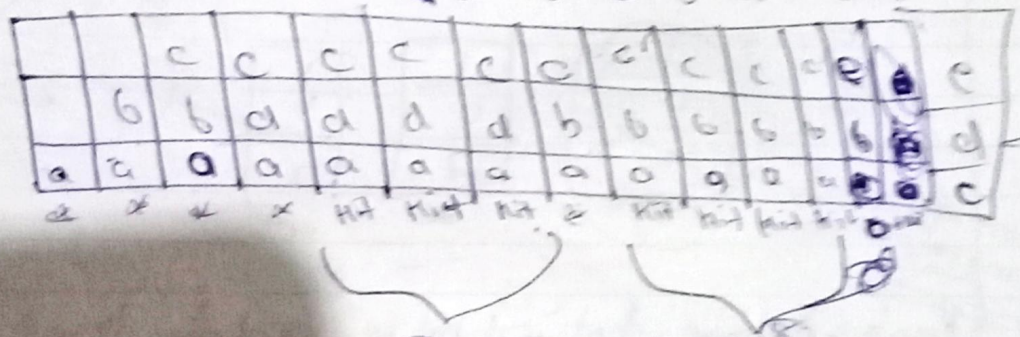
Sol) - FIFO



=>

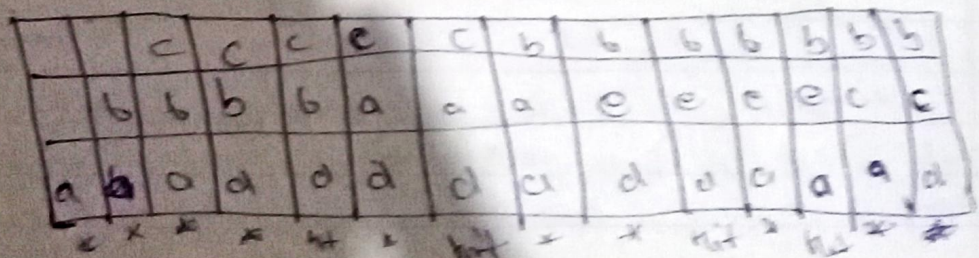
optimal

a b c d e a d b e b a b c d



=> LRU

a b c d c a d b e b a b c d



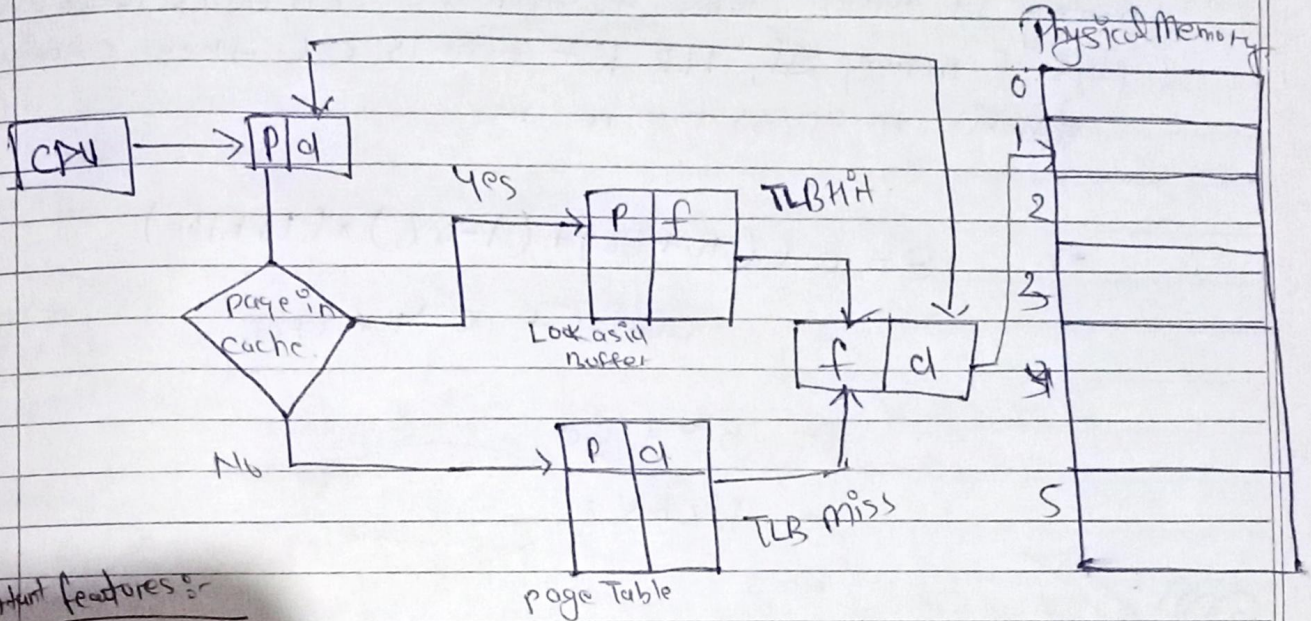
Name of Experiment..... Date.....

Experiment No..... Experiment Result..... Page No.

LRU :-

Page which has not been used for the longest period of time is selected for replacement.

Translation Look aside Buffer



Important features :-

→ **TLB Hit :-** CPU generates virtual address. It is checked in TLB. (if present) corresponding frame number retrieved, which now tells where in the main memory the page lies.

→ **TLB Miss :-** CPU generates virtual address, It is checked in TLB. (if not present). Now the page number is matched to the page table, if not available in the main memory a page fault is issued than the TLB is updated to include the new page entry.

Teacher's Signature : _____

note

Effective Memory access time (EAT) = $h \times (c+m) + (1-h) * (c+2m)$

h → hit ratio of TLB.

1-h → miss ratio of TLB.

m → memory access time.

c → TLB search time. (TLB access time)

Q)- Consider a paging hardware with a Translation Look aheed buffer. Assume that entire page table & all the pages are in the physical memory. It takes 10 ms to search the TLB, 80 ms to access the physical memory If TLB hit ratio is 0.6. then calculate effective m. access time in ms.

=) $E = 0.6(10+80) + (1-0.6) * (10+160)$

$= 60 \times 0.6 + 0.4 * 170$

$= 54 + 68$

$= 122$ ms

Q)

TLB look up takes = 5 ms = c mem access time = 100 ms

hit ratio = 0.8. m = 100

$E =$

$= E = 0.8(5+100 ms) + (0.2(5+200)$

$= 84 + 41 = 125$ ms

Name of Experiment..... Date.....

Experiment No..... Allocation Experiment Result Page No.

Important

- Allocation of frames $(q) = \frac{S_i}{S} * m$.

q_i → Allocation of frames
S_i → No of pages for a particular process
S → Total No. of pages
m → Total no. of frames

Q)- for proportional allocation we consider split 62 frames b/w two processes P1 uses 10 pages & P_2 uses 127 pages by allocation How many frames will be allocated for P_1 & P_2

for P_1

$$q_i = \frac{10}{137} * 62 = \frac{620}{137} = 4.5254$$

for P_2

$$q_i = \frac{127 \times 62}{137} = 57.47 = 57.50 \text{ ans}$$

Q) - for process P_1 = 25 pages for P_2 = 49 pages if total no. of frame = 50 allocated frames across P_1 & P_2 respectively.

$$P_1 = \frac{25 \times 50}{25+49} = \frac{1250}{74} = 16.89 = 17$$

$$P_2 = \frac{49 \times 50}{25+49} = \frac{2450}{74} = 33.10 = 33$$

Teacher's Signature