Date - 7 Dec

Non-Contiguous memory Allocation :-

Page size = Frame Size

Deviding process in pages and putting them in frames of main menory is called paging.

CPU will execute the process but is doesn't about paging is used.

Whenever the CPU wants to execute the process, it will call the process

but byte no. 3 is locatatt in main memory at 9 so to excess that we need mapping because CPU doesn't generate ab address.

We need mapping which convert 3 to 9 and break us the requested data.

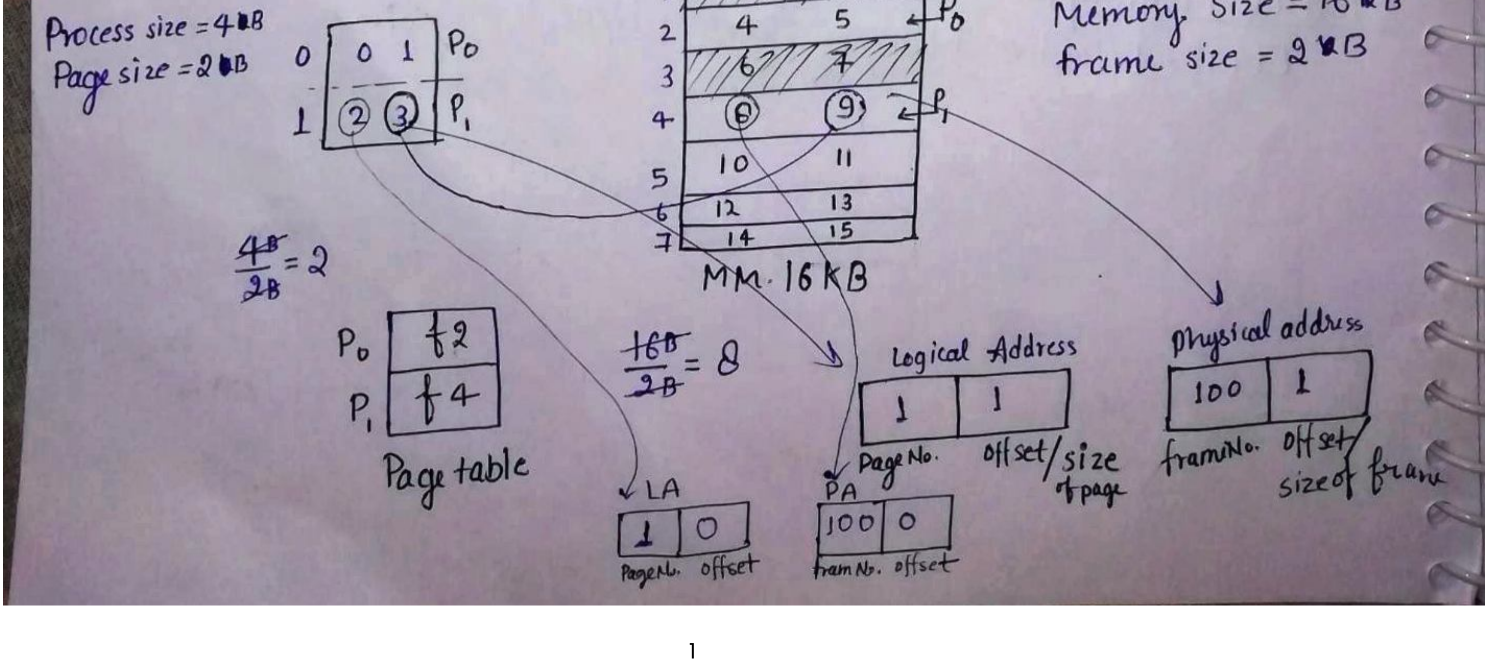
To do mapping we need ~~to~~ MMU (Memory management Unit) convert the CPU generated addresses into ab address.

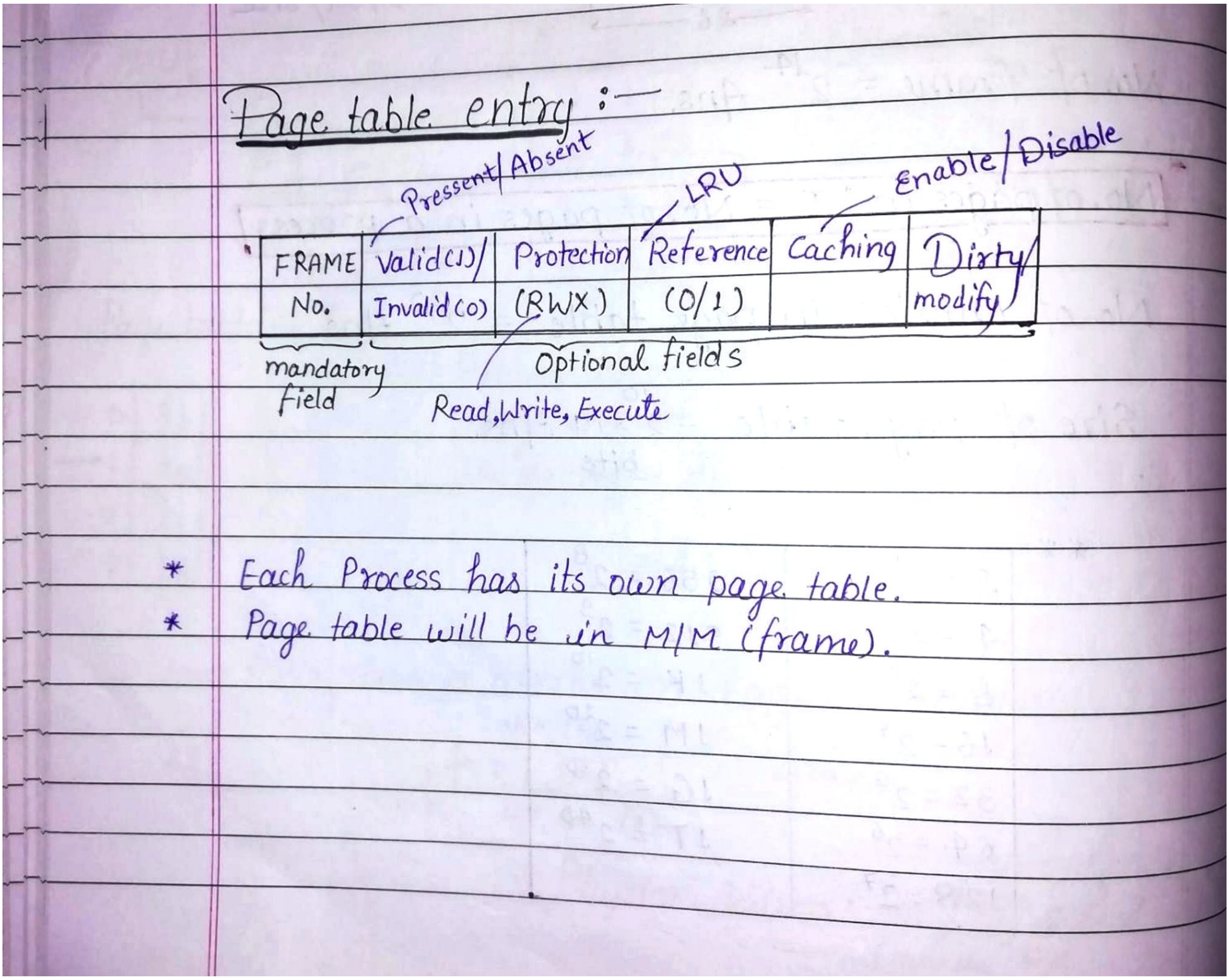
For converting MMU using page table.

Page table contains frame numbers that where page is present in MM

Every process

- CPU always works on logical address that is page number and page offset.





Page table entry :—

Pressent/Absent

LRU

Enable/Disable

FRAME No.	Valid(1)/ Invalid(0)	Protection (RWX)	Reference (0/1)	Caching	Dirty/ modify

mandatory field

Optional fields

Read, Write, Execute

* Each Process has its own page table.

* Page table will be in M/M (frame).

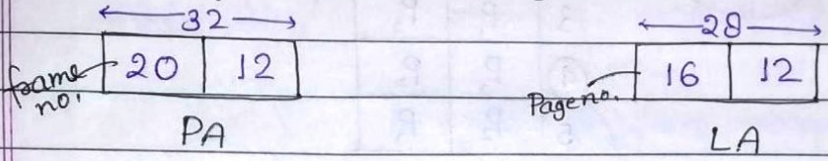
Two-level Paging :-

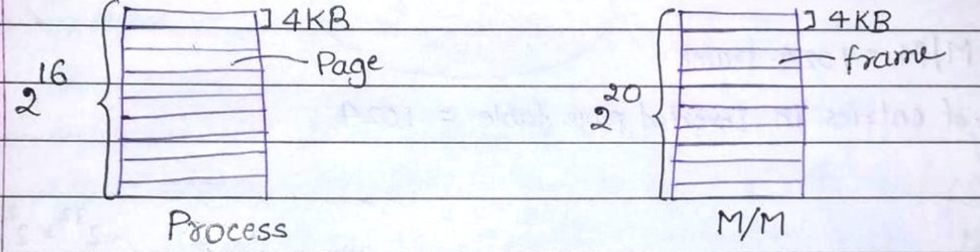
Example :- Physical address space = 256 MB = 2^{28} B

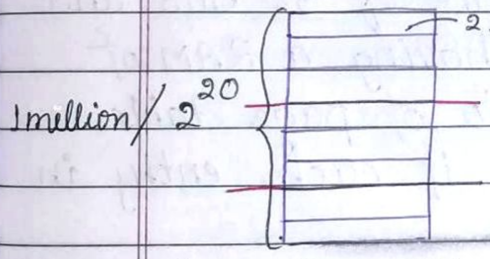
Logical address space = 4GB

Frame size = 4KB = 2^{12} B

Page table Entry = 2B



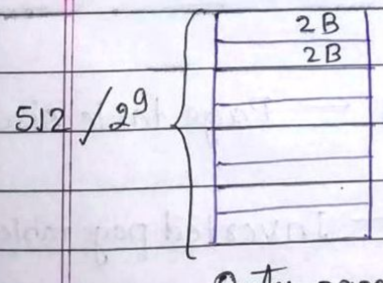




inner Page Table

PT size = $2^{20} \times 2$ B = 2MB

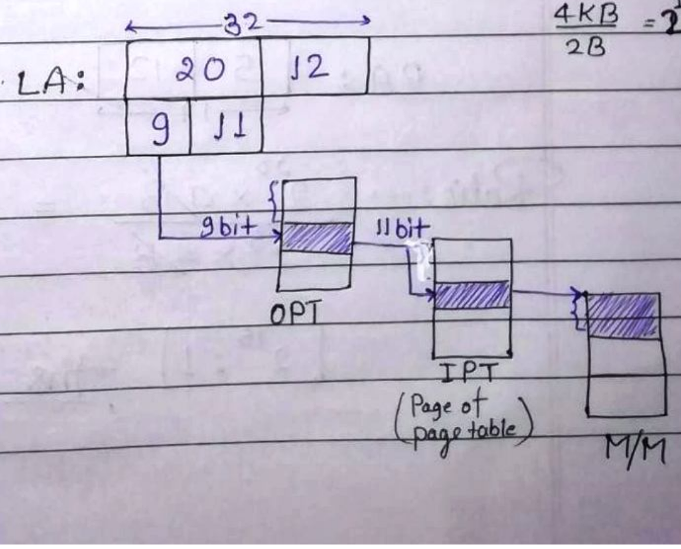
Total No. of pages = $\frac{2MB}{4KB} = 2^{9}$ = 512

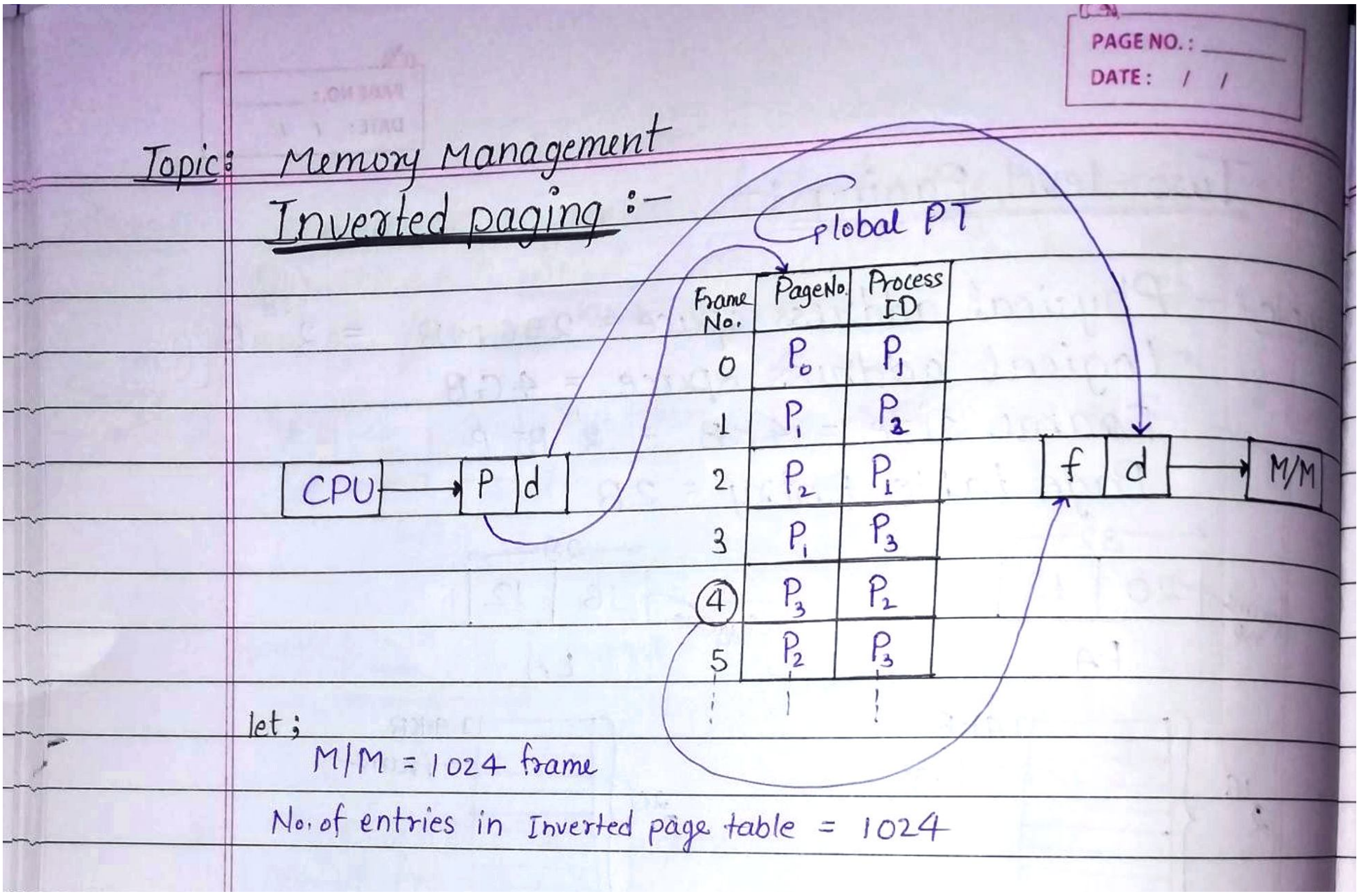


Outer page Table

OPT size = $2^{9} \times 2$ = 1KB

$\frac{4KB}{2B} = 2^{11}$





PAGE NO.:

DATE: / /

Topic: Memory Management

Inverted paging :-

Global PT
CPU
p
d
f
d
M/M

Frame No.	PageNo.	Process ID
0	P_0	P_1
1	P_1	P_2
2	P_2	P_1
3	P_1	P_3
4	P_3	P_2
5	P_2	P_3
⋮	⋮	⋮

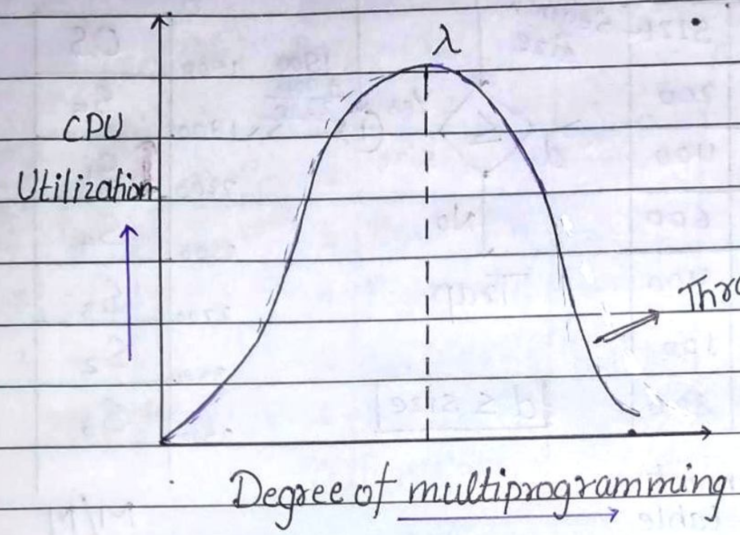
let ;

M/M = 1024 frame

No. of entries in Inverted page table = 1024

PAGE NO. :
DATE : / /

Thrashing :-

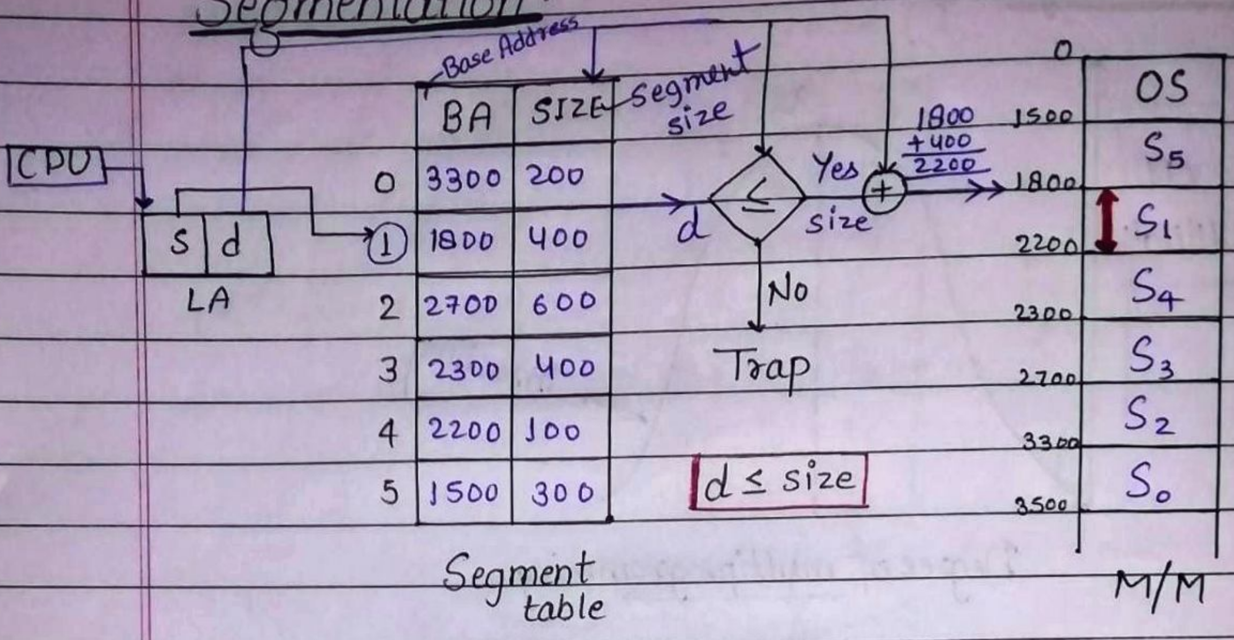


When we store maximum no. of process in RAM ~~is~~ called multiprogramming. & when CPU not hit the process (page) in the ram means page fault then occurs thrashing. Because system take lots of time to service the page fault from hard disk to RAM.

Thrashing remove by :-

1) M/M size increased
2) Long term schedular

Segmentation :-



	BA	SIZE
0	3300	200
1	1800	400
2	2700	600
3	2300	400
4	2200	100
5	1500	300

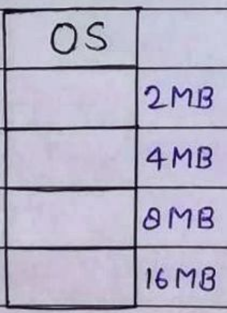
$\boxed{d \le size}$

* Segmentation done by various size.

Let:-

segment no. | 0001 | 10000 | segment offset

Overlay :- * Overlay concepts use in embedded System.



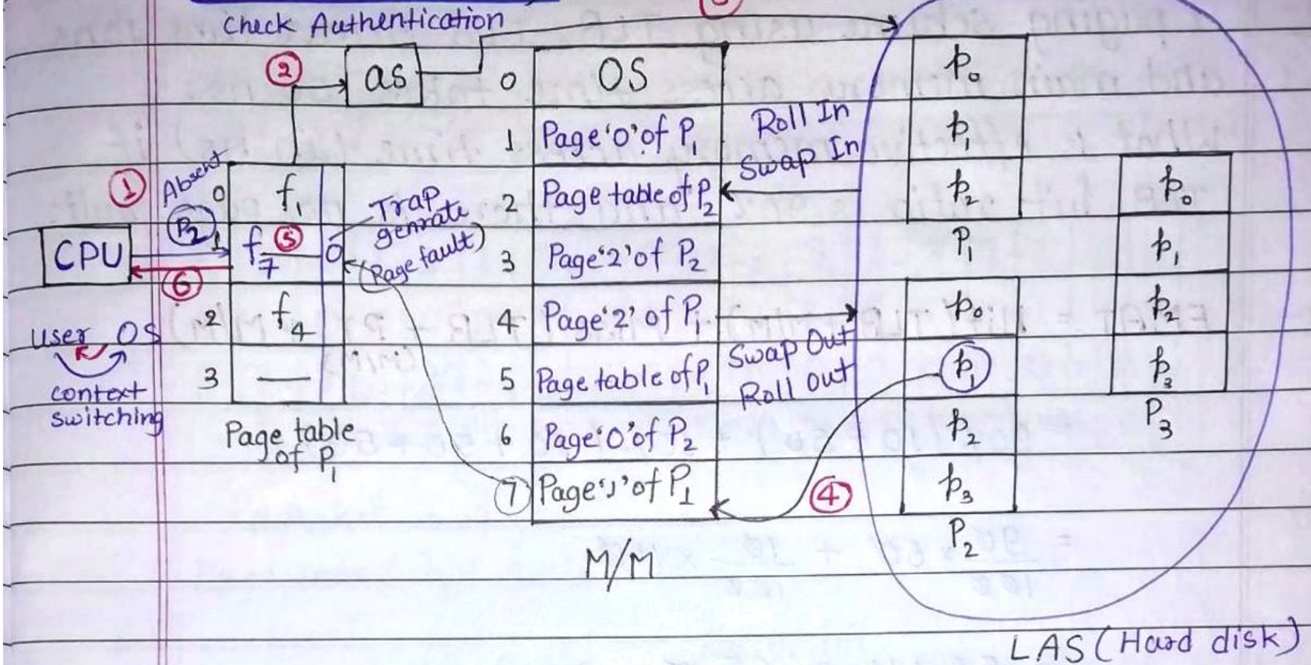
* Overlay use where main memory is limited.

Ques:- Consider a two Pass assembler pass1: 80KB, pass2: 90KB Symbol table: 30KB, common routine: 20KB At a time only one pass is in use what is min partition size required if overlay driver is 10KB size.

⟹

	Pass1	Pass2
80KB	80KB	90KB
90KB	30KB	30KB
30KB	20KB	20KB
20KB	10KB	10KB
10KB	140KB	150KB — Ans.
X 230KB		

Virtual Memory :—

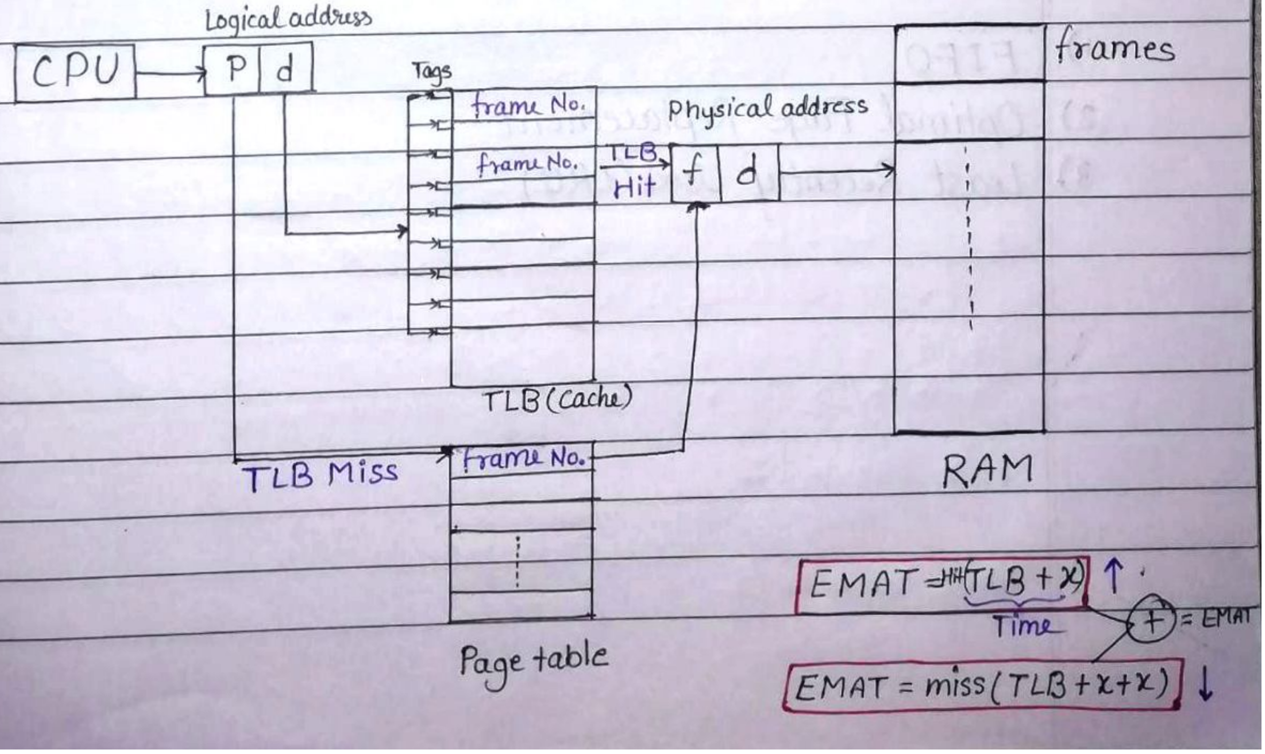


Effective Memory Acess time :

$$EMAT = \underbrace{p}_{\text{Page fault occur}}(\text{page fault service time}^{\text{msec}}) + (1-p)(\text{main memory access time})^{\text{nsec}}$$

** Translation Lookaside Buffer (TLB) :—

(cache memory)



$\boxed{EMAT = \text{Hit}\underbrace{(TLB + x)}_{\text{Time}}}$ ↑

$\boxed{EMAT = \text{miss}(TLB + x + x)}$ ↓

Ques:- A paging scheme using TLB. TLB Access time 10ns and main memory access time takes 50 ns. What is effective memory access time (in ns) if TLB hit ratio is 90% and there is no page fault.

Solve:- $EMAT = Hit(TLB + M/M) + Miss(TLB + \underset{(M/M)}{PT} + M/M)$

$= 90\%(10 + 50) + 10\%(10 + 50 + 50)$

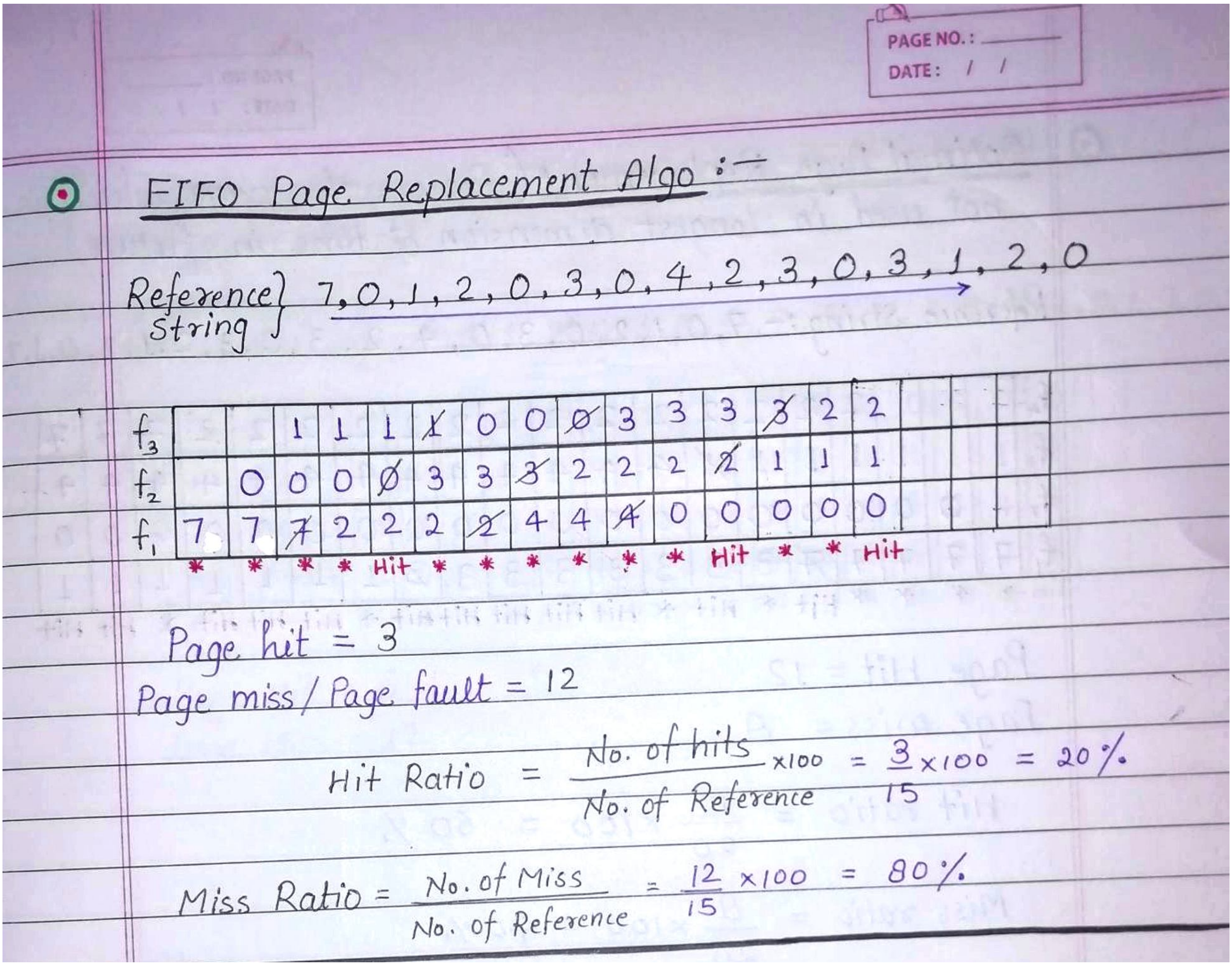
$= \frac{9\cancel{0}}{1\cancel{0}\cancel{0}} \times 6\cancel{0} + \frac{1\cancel{0}}{1\cancel{0}\cancel{0}} \times 11\cancel{0}$

$= 54 + 11 = 65$ ns. Ans.

* If page fault occur then also add page fault service time in EMAT.

Page Replacement algo :—

1) FIFO
2) Optimal Page Replacement
3) Least Recently Used (LRU)



PAGE NO. :

DATE : / /

FIFO Page Replacement Algo :—

Reference String } 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 1, 2, 0

f_3			1	1	1	~~1~~	0	0	~~0~~	3	3	3	~~3~~	2	2			
f_2		0	0	0	~~0~~	3	3	~~3~~	2	2	2	~~2~~	1	1	1			
f_1	7	7	~~7~~	2	2	2	~~2~~	4	4	~~4~~	0	0	0	0	0			
	*	*	*	*	Hit	*	*	*	*	*	*	Hit	*	*	Hit			

Page hit = 3

Page miss / Page fault = 12

$$\text{Hit Ratio} = \frac{\text{No. of hits}}{\text{No. of Reference}} \times 100 = \frac{3}{15} \times 100 = 20\%$$

$$\text{Miss Ratio} = \frac{\text{No. of Miss}}{\text{No. of Reference}} = \frac{12}{15} \times 100 = 80\%$$

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

f_4				2	2	2	2	2	2	2	2	~~2~~	1	1	1
f_3			1	1	1	1	1	1	1	~~1~~	0	0	0	0	0
f_2		0	0	0	0	0	~~0~~	4	4	4	4	4	4	4	4
f_1	7	7	7	7	~~7~~	3	3	3	3	3	3	3	~~3~~	2	2
	*	*	*	*	Hit	*	Hit	*	Hit	Hit	*	Hit	*	*	Hit

Page Hit = 6

Page Miss = 9

Ques 2: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

f_3			3	3	~~3~~	2	2	2	2	~~2~~	4	4					
f_2		2	2	~~2~~	1	1	1	1	~~1~~	3	3	3					
f_1	1	1	~~1~~	4	4	~~4~~	5	5	5	5	5	5					
	*	*	*	*	*	*	*	Hit	Hit	*	*	Hit					

f_4				4	4	4	4	4	4	3	3	3							
f_3			3	3	3	3	3	3	2	2	2	2							
f_2		2	2	2	2	2	2	1	1	1	1	5							
f_1	1	1	1	1	1	1	5	5	5	5	4	4							
	*	*	*	*	Hit	Hit	*	*	*	*	*	*							

Page Hit = 2

Page miss = 10

Optimal Page Replacement (Replace the page which is not used in longest dimension of time in future)

Reference String:- 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,

f_4				2	2	2	2	2	2	2	2	2	2	2	2	2	~~2~~	7	7	7
f_3			1	1	1	1	~~1~~	4	4	4	4	4	4	4	4	4	4	4	4	4
f_2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_1	7	7	7	7	~~7~~	3	3	3	3	3	3	3	~~3~~	1	1	1	1	1	1	1
	*	*	*	*	Hit	*	Hit	*	Hit	Hit	Hit	Hit	Hit	*	Hit	Hit	Hit	*	Hit	Hit

Page Hit = 12

Page miss = 8

Hit ratio = $\frac{12}{20} \times 100$ = 60 %.

Miss ratio = $\frac{8}{20} \times 100$ = 40 %.

Least Recently used (Replace the least recently used page in Past)

Reference String:- 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

f_4				2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
f_3			1	1	1	1	~~1~~	4	4	4	4	4	~~4~~	1	1	1	1	1	1	1
f_2		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f_1	7	7	7	7	~~7~~	3	3	3	3	3	3	3	3	3	3	3	~~3~~	7	7	7
	*	*	*	*	Hit	*	Hit	*	Hit	Hit	Hit	Hit	Hit	*	Hit	Hit	Hit	*	Hit	Hit

Page Hit = 12

Page Miss = 8