CS & IT ENGINEERING

Operating System

Memory Management

Lecture -7

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Virtual Memory

Topic Multilevel Paging

Topics to be Covered

Topic Inverted Paging

Topic Hashed Page table

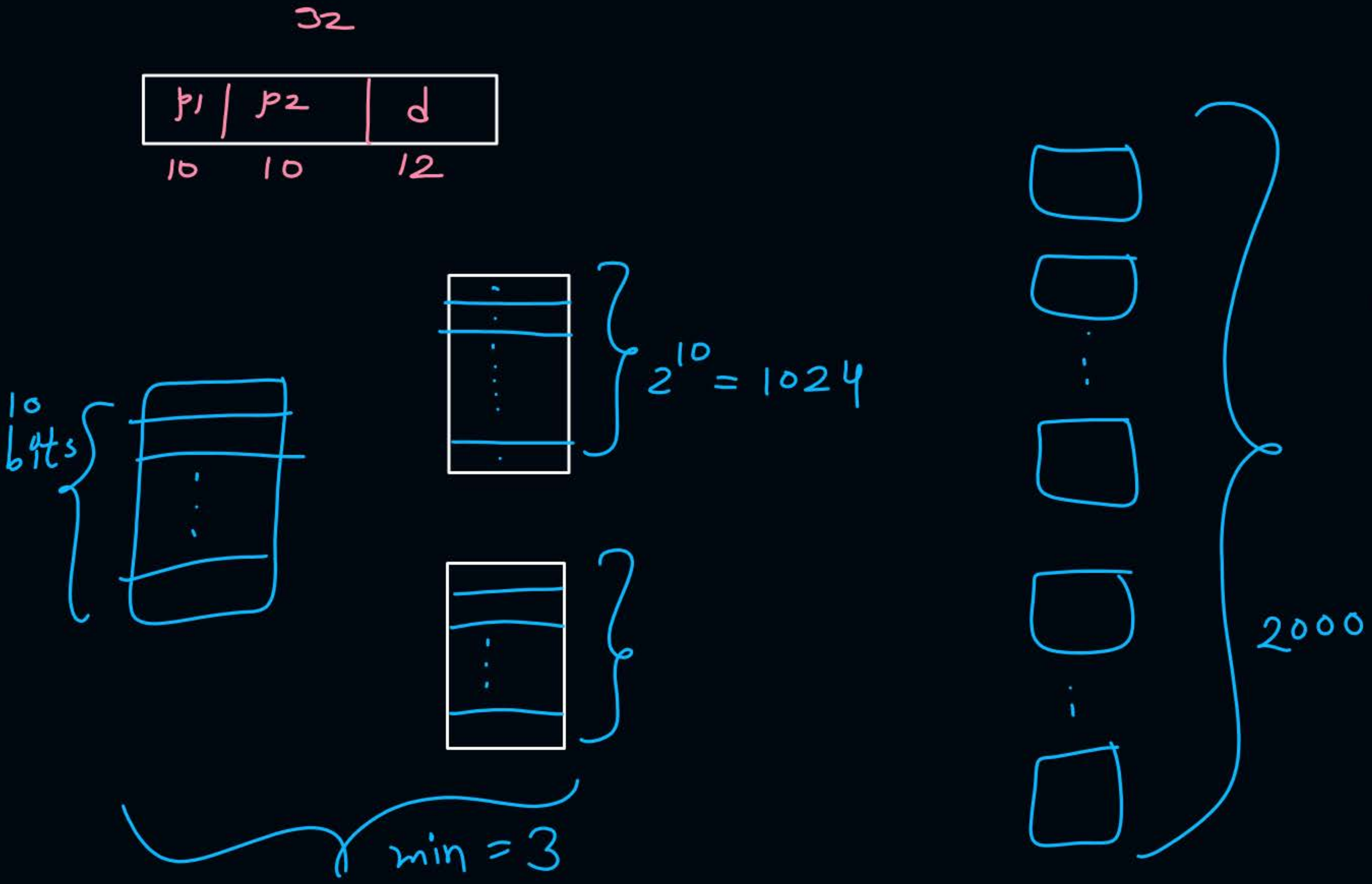
Topic : Question

[GATE-2024]

#Q. Consider a 32-bit system with 4 KB page size and page table entries of size 4 bytes each. Assume 1 KB = 2^{10} bytes. The OS uses a 2-level page table for memory management, with the page table containing an outer page directory and an inner page table. The OS allocates a page for the outer page directory upon process creation. The OS uses demand paging when allocating memory for the inner page table, i.e., a page of the inner page table is allocated only if it contains at least one valid page table entry.

An active process in this system accesses 2000 unique pages during its execution, and none of the pages are swapped out to disk. After it completes the page accesses, let X denote the minimum and Y denote the maximum number of pages across the two levels of the page table of the process.

The value of X + Y is 1028?



max

keep all inner P.T. page

$= 2^{10} = 1024$

outer = 1

1025 max pages

Ques) A system has LA 32 bits

page size 1 KB

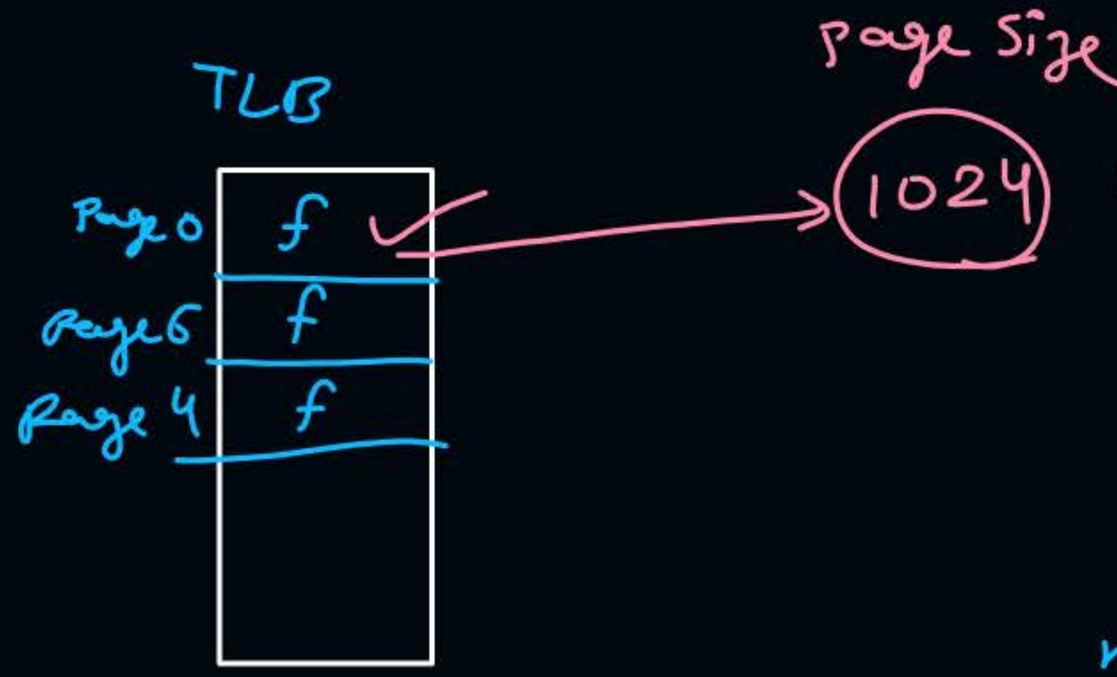
TLB with 128 entries

P.A. = 28 bits ⇒ P.A.S. = 2^{28} B

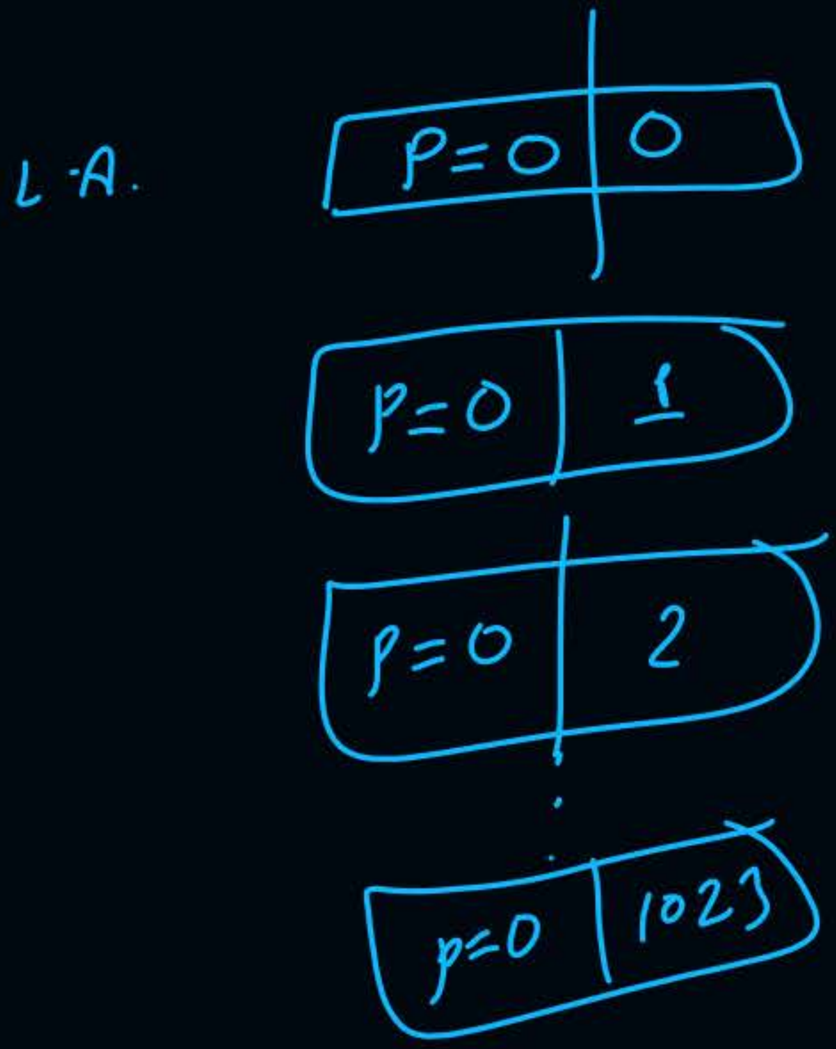
1. max No. of L.A. accessed by CPU without any TLB miss?

2. — " ———— " ———— " ———— any page fault?

1.)



LA can have a hit



max no. of L.A. hit in TLB w/o miss

$= 128 * 1024$

$= 128K$

$= 2^{17}$

2) as many as P.A. physical mem. has.

$$= 2^{28}$$

$$= 256M$$

Topic : LA to PA Translation

outer P.T.

0	10001
1	01111

Inner P.T. (frame no. ⇒ 01111)

000	00101
001	11100
010	00010
011	01001
100	01000
101	00110
110	10011
111	11011

(frame no. 10001)

000	10000
001	00100
010	11001
011	00111
100	10111
101	01100
110	11111
111	01010

Page size = 16 bytes

LA

← 8 bits →

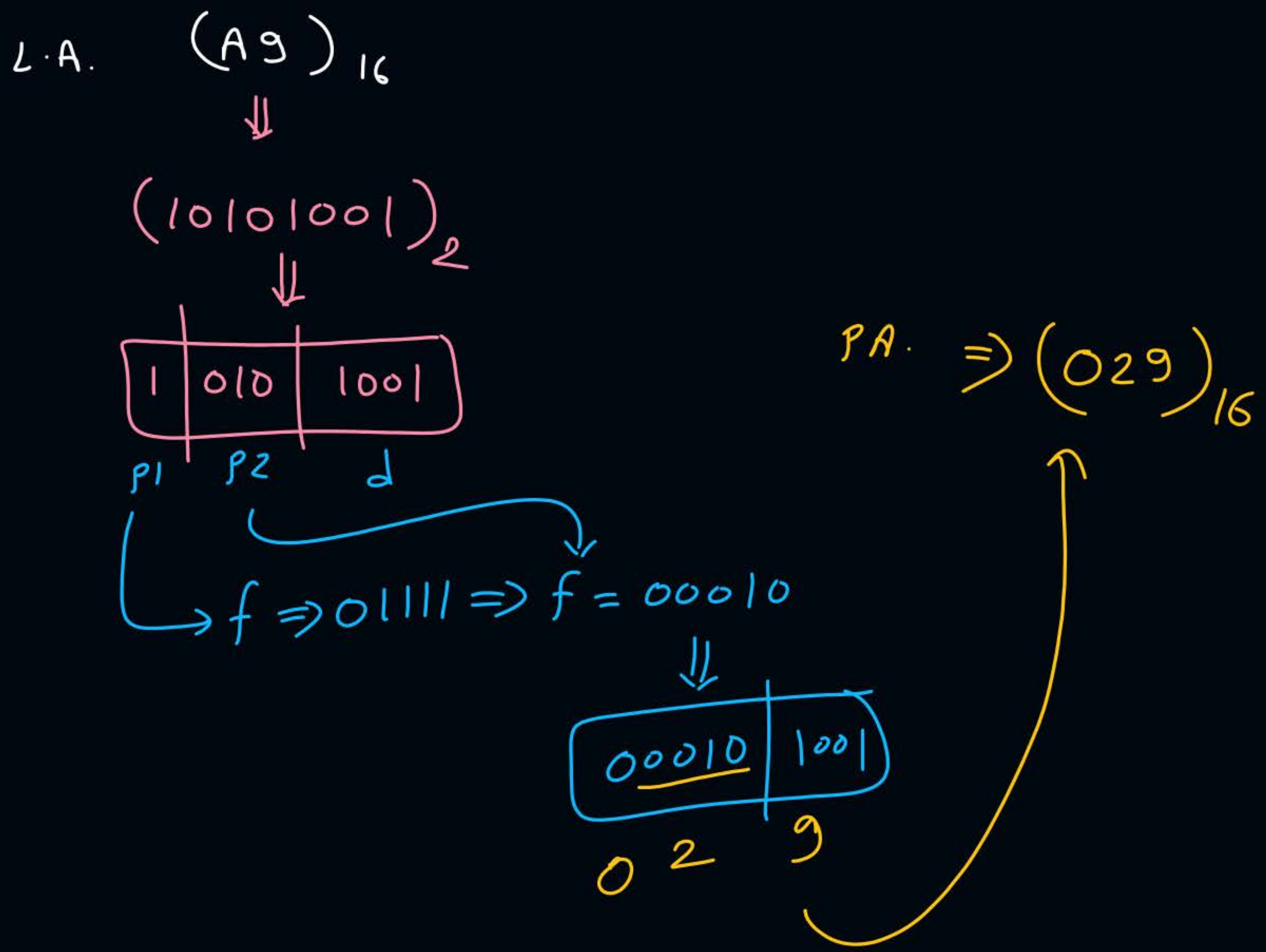
P1	P2	d
1	3	4

LA

1	001	0011
p1	p2	d

↳ search in outer P.T. ⇒ 01111 ⇒ search in frame 01111 with p2 = 001 ⇒ f = 11100

11100	0011


L.A. $(A9)_{16}$
$\Downarrow$
$(10101001)_2$
$\Downarrow$
1 | 010 | 1001
p1 p2 d
$f \Rightarrow 01111 \Rightarrow f = 00010$
$\Downarrow$
00010 | 1001
0 2 9
P.A. $\Rightarrow (029)_{16}$

⇒ As page size increases, P.T. size decreases

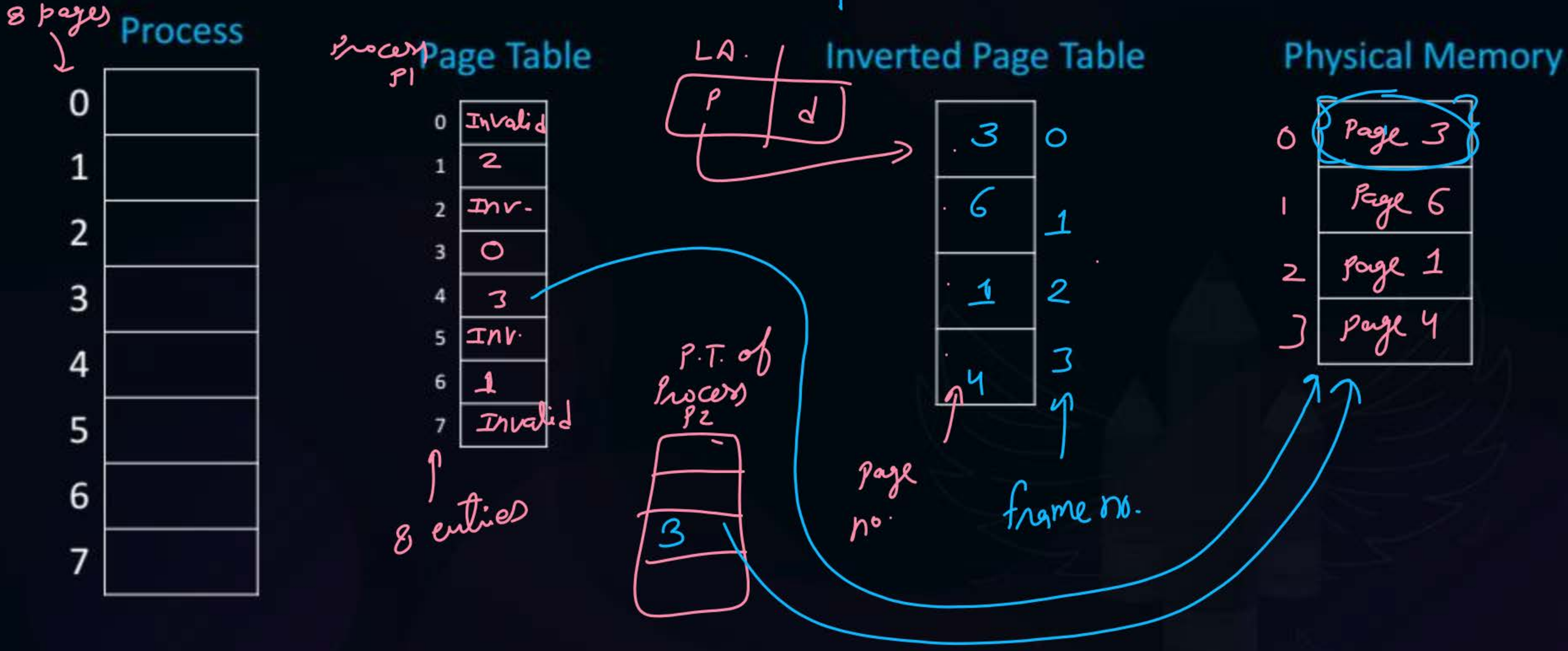
→ Big pages ⇒ more unreferenced data in mm.

→ Big pages ⇒ P.f. service time also higher

→ Page fault service may need DMA.

Topic : Inverted Page Table

no. of entries = no. of frames



Process P1 Page Table

0	Invalid
1	2
2	Inv.
3	0
4	3
5	Inv.
6	1
7	Invalid

Inverted Page Table

page no.	frame no.
3	0
6	1
1	2
4	3

Physical Memory

0	Page 3
1	Page 6
2	page 1
3	page 4

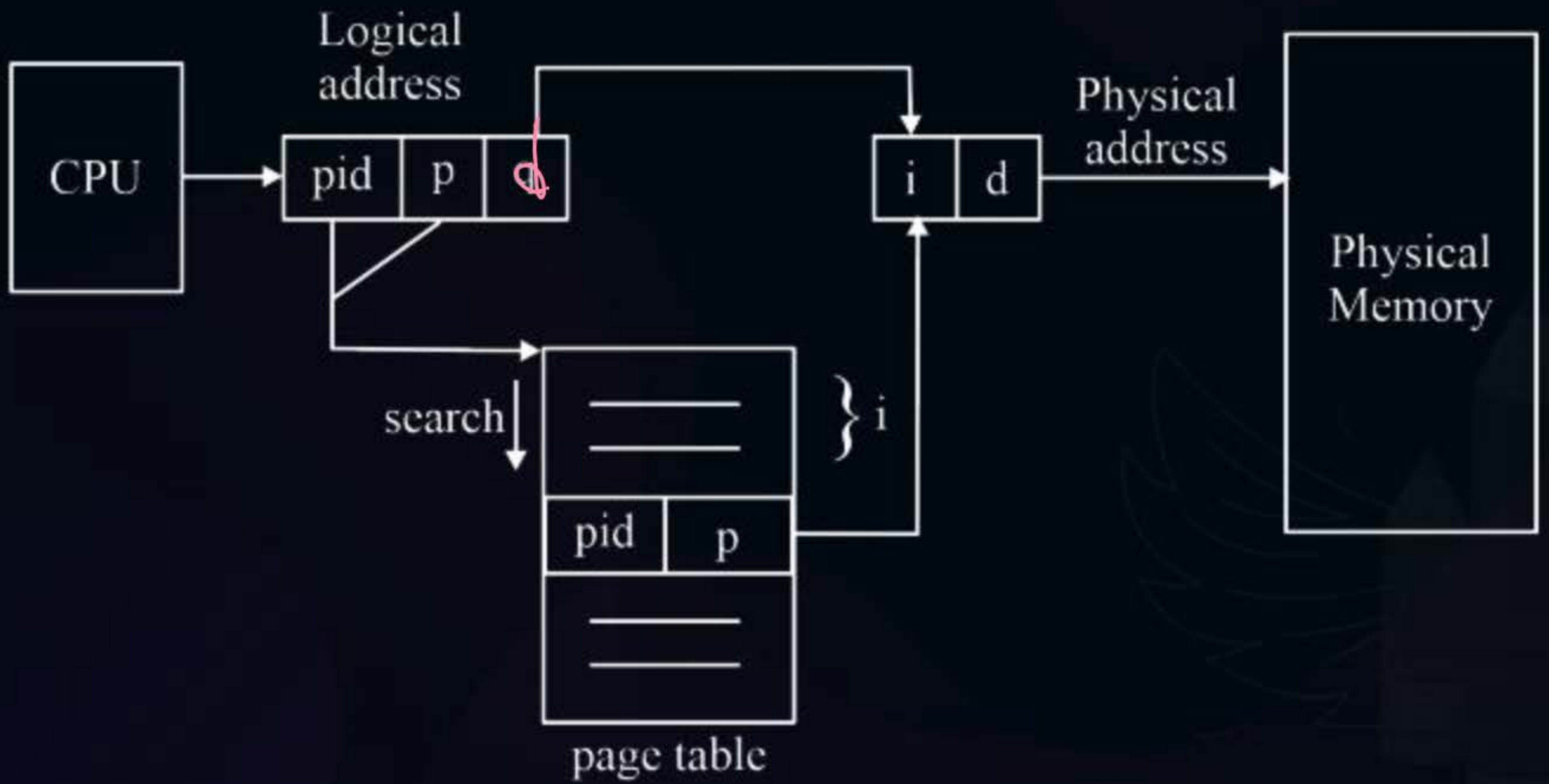
Topic : Inverted Page Table

→ one for all processes

Each entry in the page table contains the following fields:

1. Page number
2. Process id
3. Control bits
4. Chained pointer

Topic : Inverted Page Table



Topic : Inverted Page Table

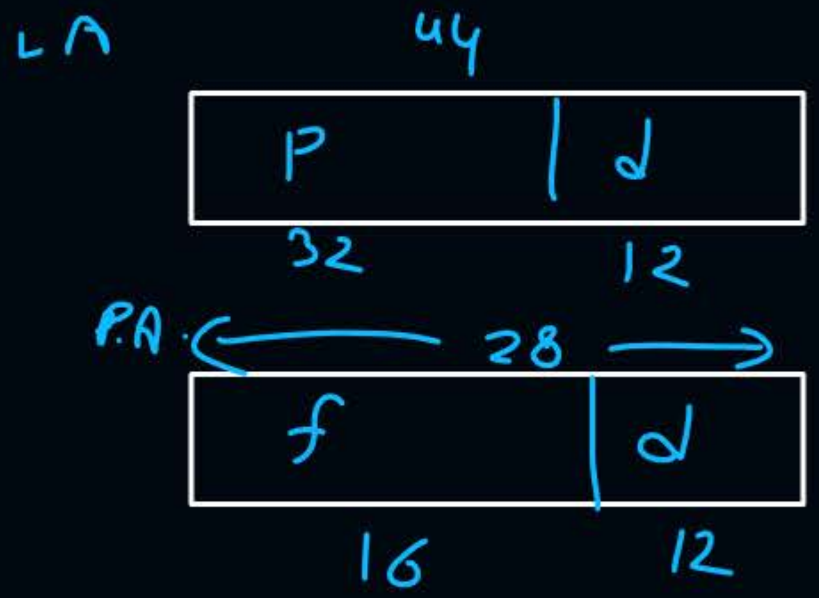
Advantages & Disadvantages:

1. Reduced memory space
2. Longer lookup time
3. Difficult shared memory implementation

Ques) L.A. = 44 bits

Pagesize = 4KB

P.A. = 28 bits



P.T. size = ______ ?

$= 2^{32} * 16$ bits

$= 2^{32} * 2$ B

= 8 GBytes

Inverted P.T. size = ______ ?

$= 2^{16} * 32$ bits

$= 2^{16} * 4$B

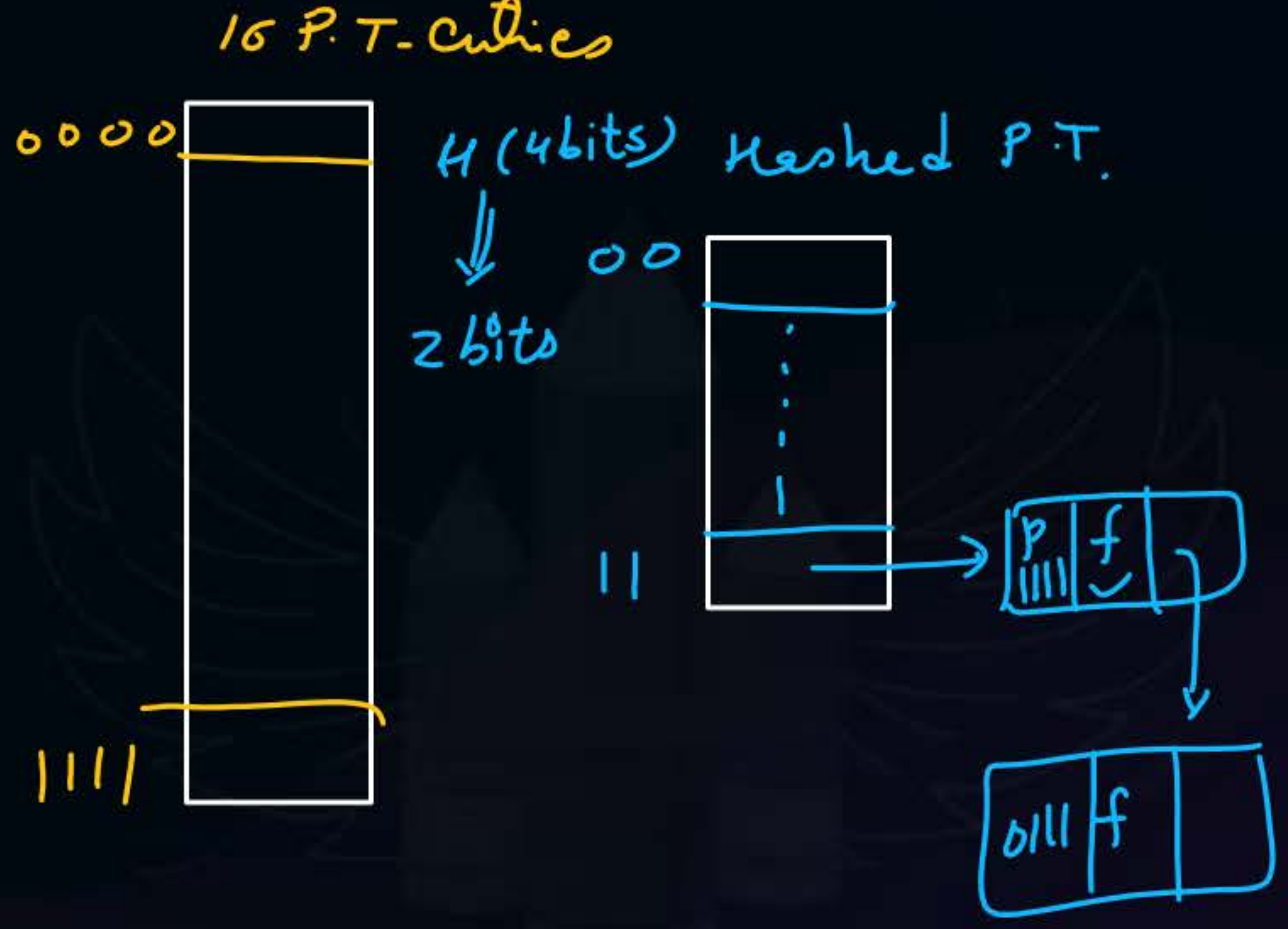
= 256 KB

Topic : Hashed Page Table

In this virtual page, the number is hashed into a page table

This Page table mainly contains a chain of elements hashing to the same elements.

$H(P) \Rightarrow$ index where P.T. entry stored



Topic Hashed Page Table

Each element mainly consists of :

1. The virtual page number
2. The value of the mapped page frame.
3. A pointer to the next element in the linked list.

2 mins Summary

Topic Inverted Paging

Topic Hashed Page table

Happy Learning

THANK - YOU