CS & IT ENGINEERING

Operating System

Virtual Memory

Lecture - 06

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Multilevel Paging

Topic Access Time in Multilevel Paging

Topics to be Covered

Topic Access Time in Multilevel Paging

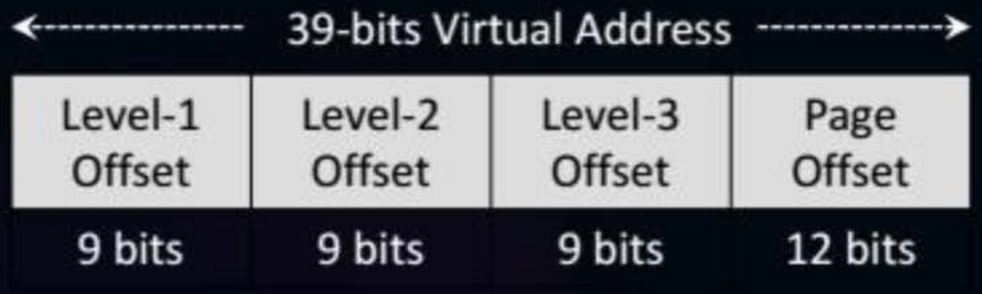
Topic Inverted Paging

Topic Hashed Page Table

Topic : Question

Ans = 4108

#Q. ~~Size of page~~ Consider a three-level page table to translate a 39-bit virtual address to a physical address as shown below:



max process size = 2^{39} B
= 512 GB

The page size is 4 KB = (1KB = 2^{10} bytes) and page table entry size at every level is 8 bytes. A process P is currently using 2 GB (1 GB = 2^{30} bytes) virtual memory which OS mapped to 2 GB of physical memory. The minimum amount of memory required for the page table of P across all levels is _____ KB across all levels?

no. of pages for outer P.T. = 1

—11— middle PT = $2^1 = 2$

—11— Inner PT = $2^1 \cdot 2^9 = 2^{10} = 1024$

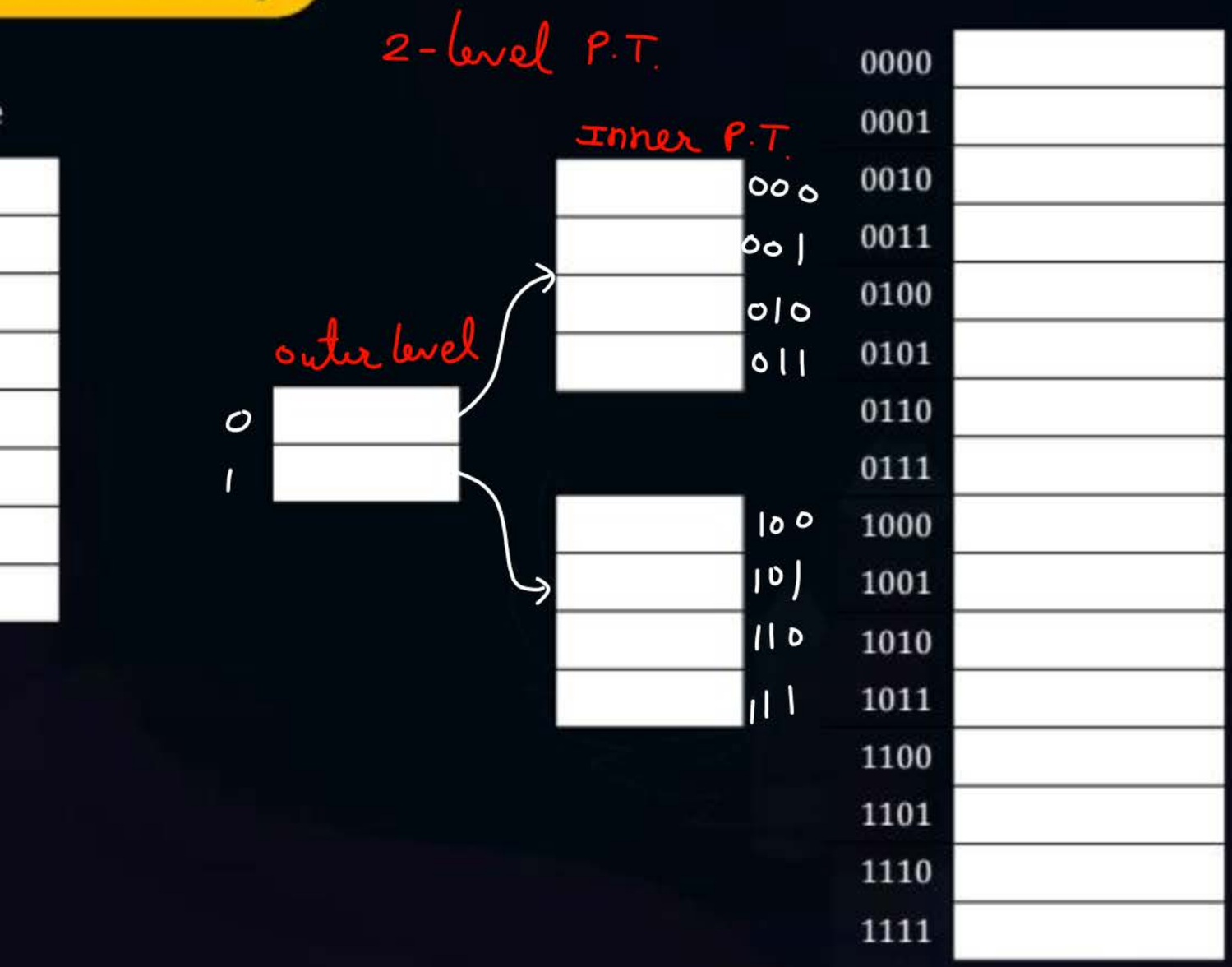
Total = 1027

Total space = 1027 * 4kB = 4108 kB

Topic : Page Table in Memory

Process
Page 000
Page 001
Page 010
Page 011
Page 100
Page 101
Page 110
Page 111

Page Table	
000	
001	
010	
011	
100	
101	
110	
111	



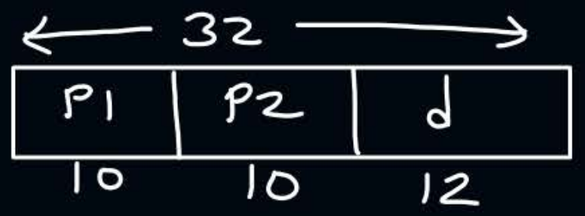
Topic : Question

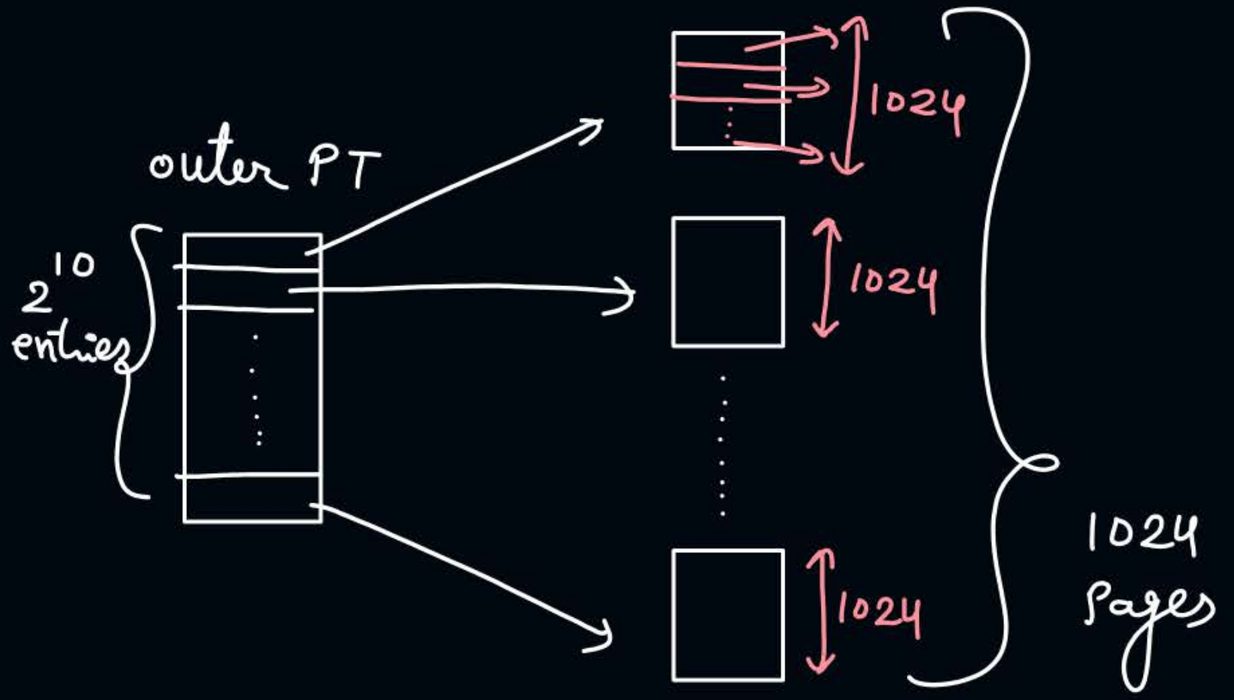
GATE- 2024

#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 _____?





no. of P.T.E. Per page $= \frac{4kB}{4B}$

$= 1k$

$= 2^{10}$

$X = 3$ (1 outer PT page + 2 inner P.T. Page) $\Rightarrow$ 2000 entries of PT across 2 inner P.T. Pages

$\underline{Y = 1025}$ (1 outer PT page + 1024 —11—)

1028

Topic : EMAT With Multilevel Paging

Single level paging :-

EMAT = 2 * tmm

(1 for PT
1 for content)

2-level paging :-

EMAT = 3 * tmm

[2 for PT
1 for content]

3-level paging :-

EMAT = 4 * tmm

2-level paging with TLB :-

$$EMAT = H * (t_{TLB} + t_{mm}) + (1 - H)(t_{TLB} + 2 * t_{mm} + t_{mm})$$

$2 * t_{mm}$: for PT

t_{mm} : for content

Topic : Question

#Q. A processor uses 2–level page tables for virtual to physical address translation. Page tables for both levels are stored in the main memory. Virtual and physical addresses are both 32 bits wide. The memory is byte addressable. For virtual to physical address translation, the 10 most significant bits of the virtual address are used as index into the first level page table while the next 10 bits are used as index into the second level page table. The 12 least significant bits of the virtual address are used as offset within the page. Assume that the page table entries in both levels of page tables are 4 bytes wide. Further, the processor has a translation look-aside buffer (TLB), with a hit rate of 96%. The TLB caches recently used virtual page numbers and the corresponding physical page numbers. The processor also has a physically addressed cache with a hit rate of 90%. Main memory access time is 10 ns, cache access time is 1 ns, and TLB access time is also 1 ns.

Assuming that no page faults occur, the average time taken to access a virtual address is approximately (to the nearest 0.5 ns)?

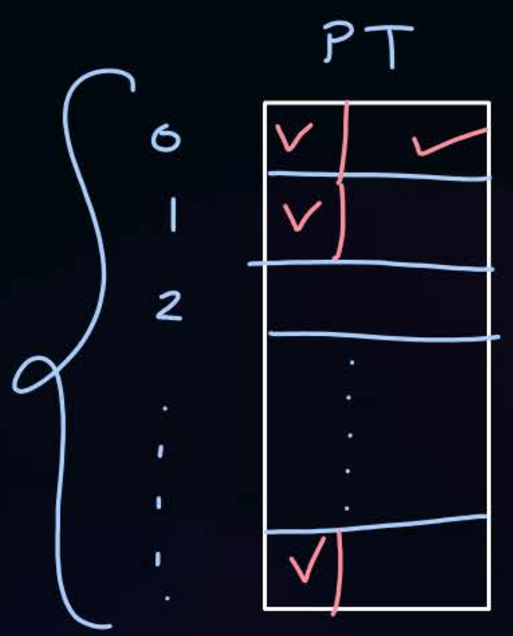
if physical add. is available then

$$\text{Content access time} = 0.9 * 1 + 0.1(1 + 10)$$
$$= 0.9 + 1.1$$
$$= 2 \text{ ns}$$

$$\text{E.M.A.T.} = 0.96(1 + 2) + 0.04(1 + 2 * 10 + 2)$$
$$= 3.8 \text{ ns}$$
$$= 4.0 \text{ ns}$$

Topic : Problem in Virtual Memory

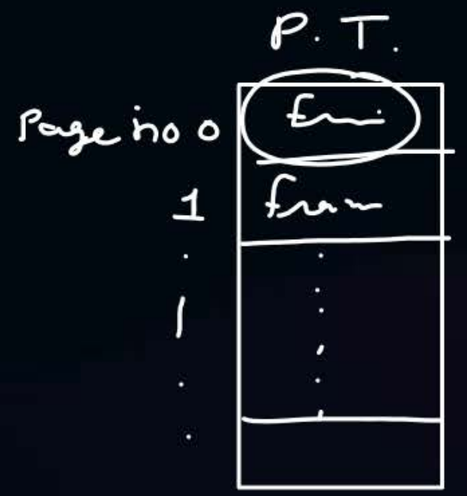
Page table is too large and so many page table entries are invalid.

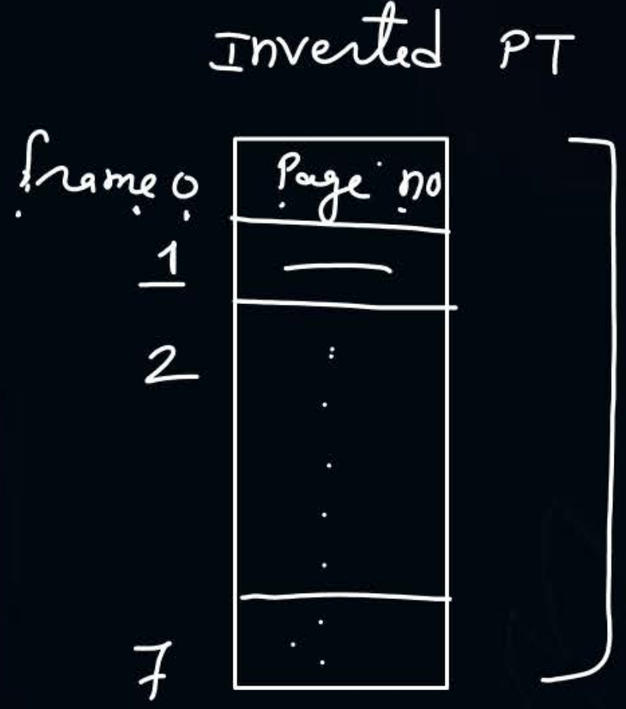


⇓

⇒ Hence a lot of space of mm wasted for keeping invalid PT entries.

Topic : Inverted Page Table





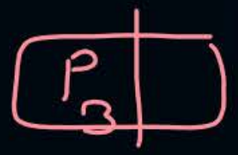
Topic : Inverted Page Table

Process

0	Page 0
1	
2	
3	
4	
5	
6	
7	page 7

Page Table

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



no. of PT entries = No. of frames

↓

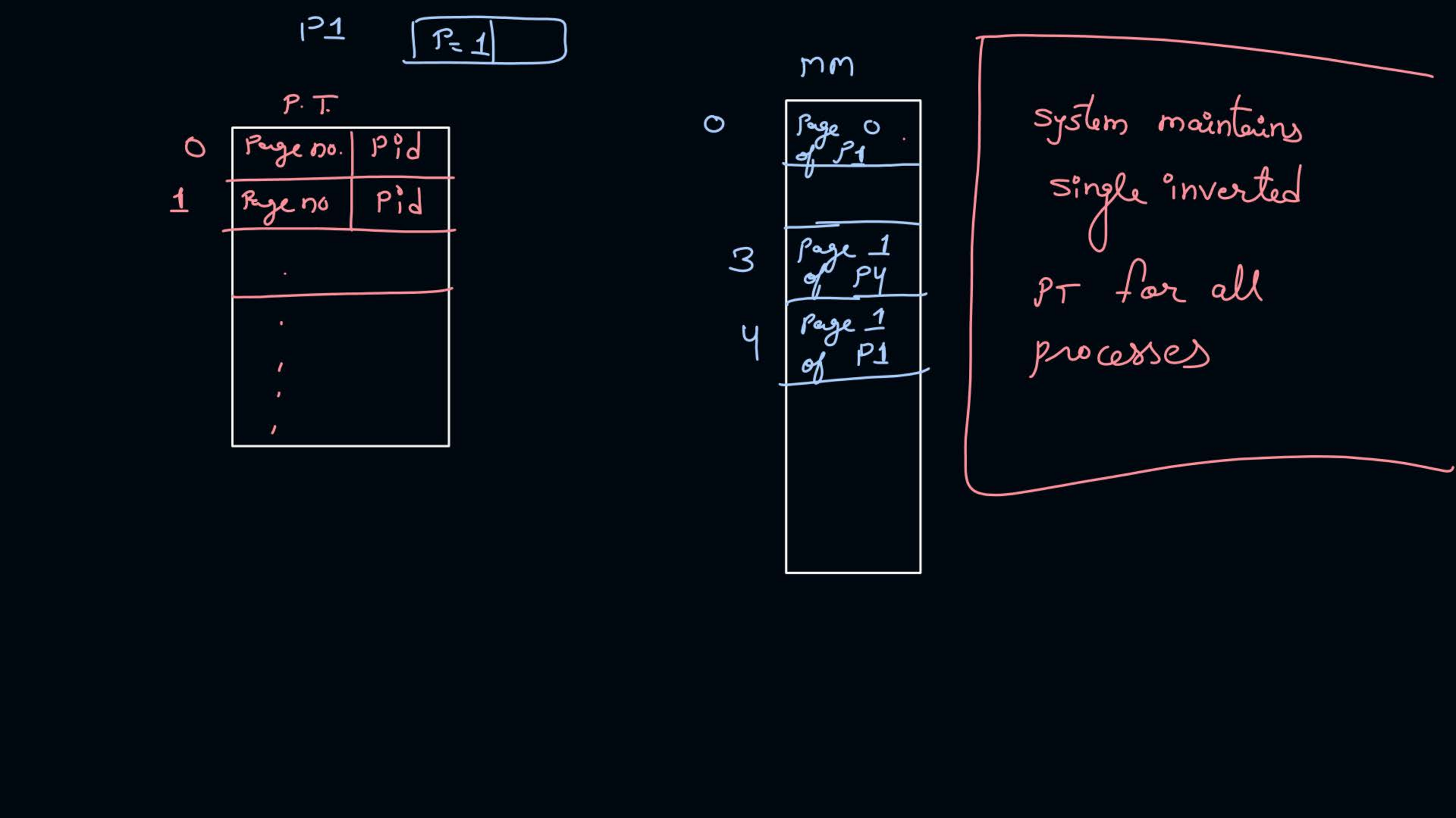
Inverted Page Table

0	6
1	3
2	7
3	1

frame no.

Physical Memory

0	Page 6
1	Page 3
2	Page 7
3	Page 1

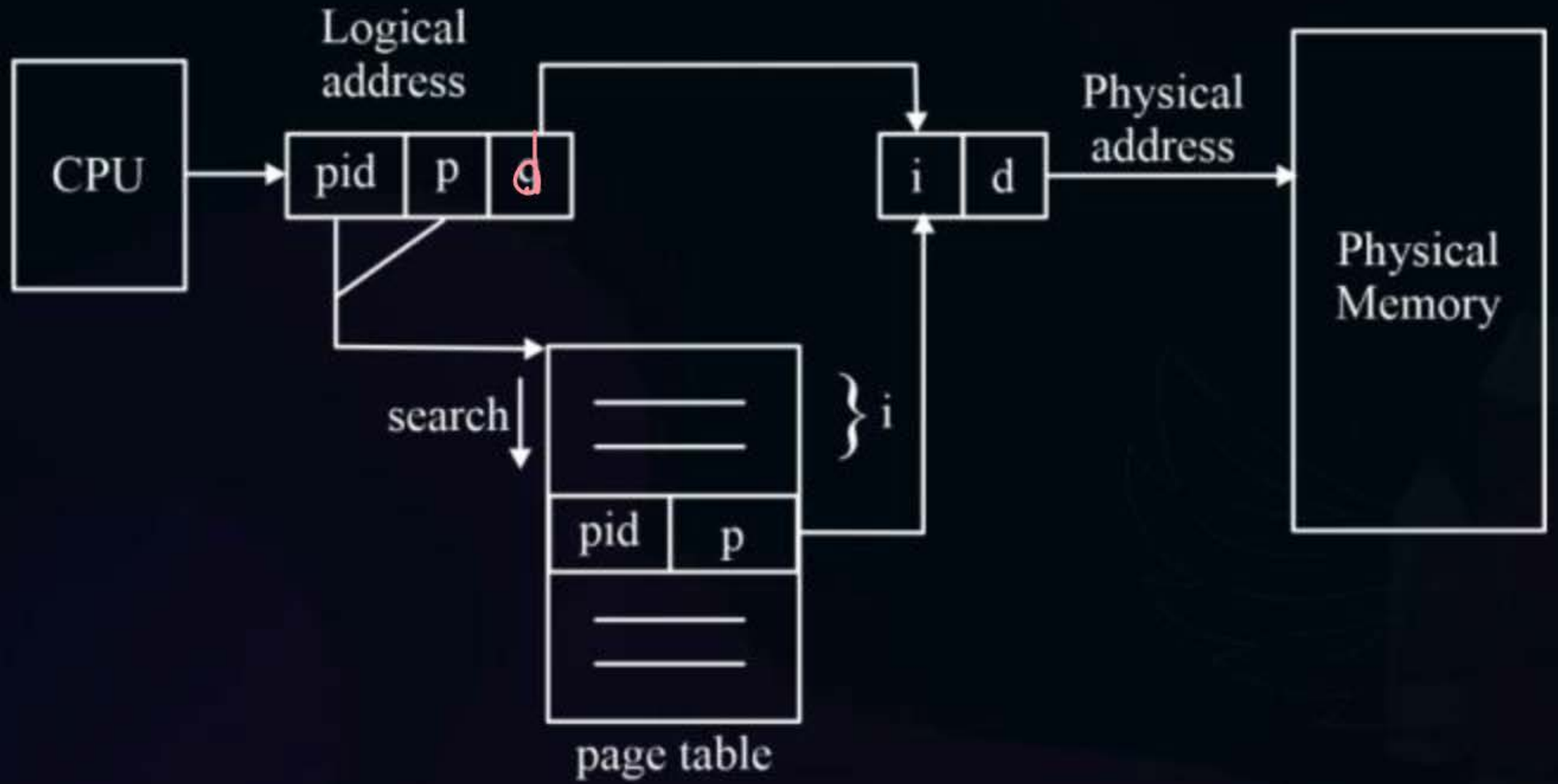

P1
P=1
P.T.
0
Page no.
Pid
1
Page no
Pid
MM
0
Page 0 of P1
3
Page 1 of P4
4
Page 1 of P1
System maintains single inverted PT for all processes

Topic : Inverted Page Table

Each entry in the page table contains the following fields:

1. Page number
2. Process id
3. Control bits → V/I, Replacement bits, dirty bit
4. Chained pointer

Topic : Inverted Page Table

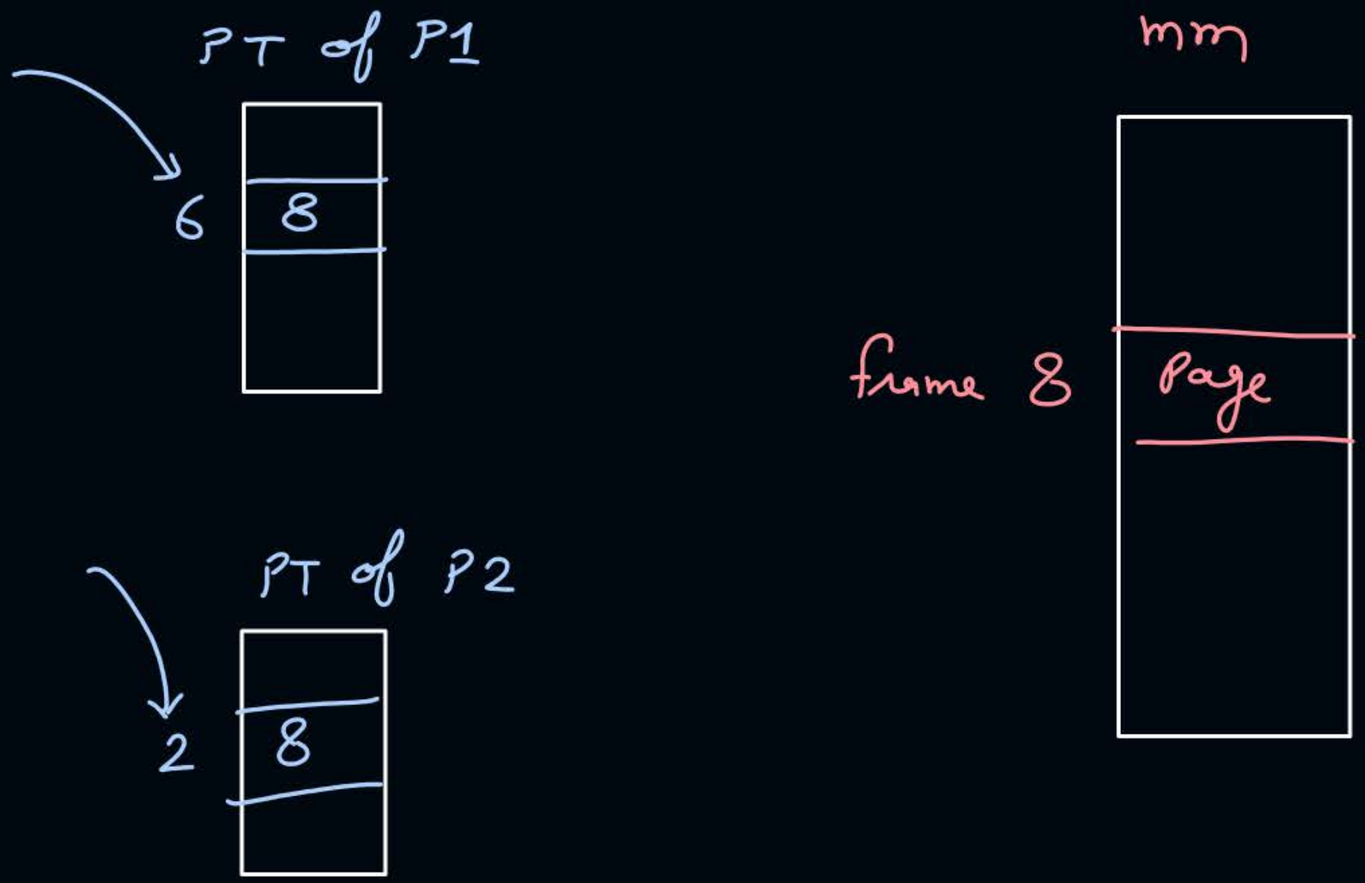


Topic : Inverted Page Table

Advantages & Disadvantages:

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

shared page :-

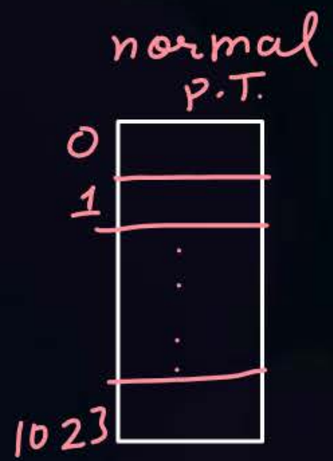


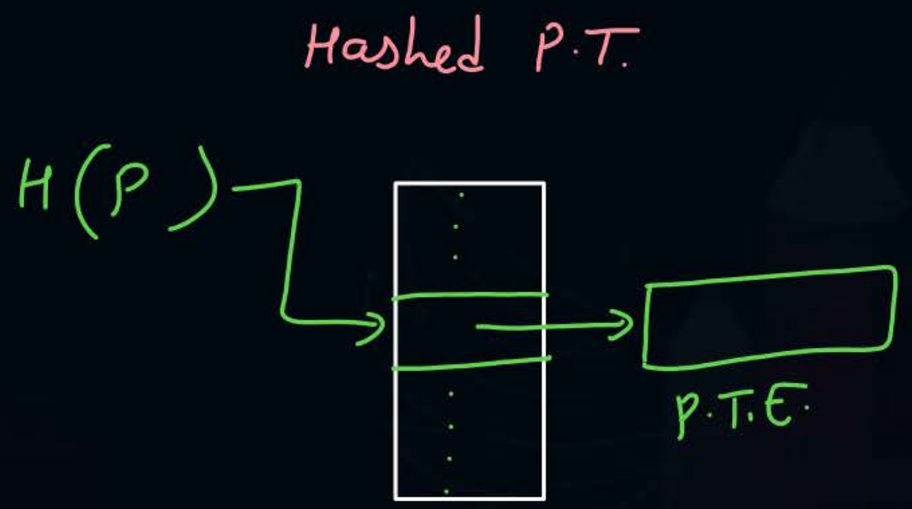
Topic : Hashed Page Table

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.

1024 Pages in a process ⇓ numbered 0 to 1023





ex:-

Process = 1024 Pages

PT = 100 slots

example

$H(P) = P \% 100$

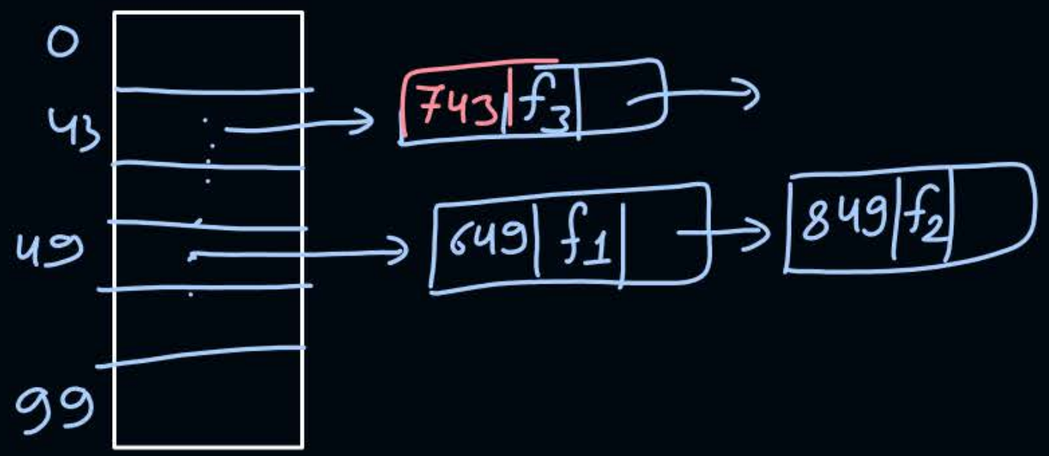
CPU Req:-

P ⇒ slot

649 ⇒ 649 % 100 = 49

849 ⇒ 849 % 100 = 49

743 ⇒ 743 % 100 = 43



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.

Topic: Impact of Page Size on Page Table Size

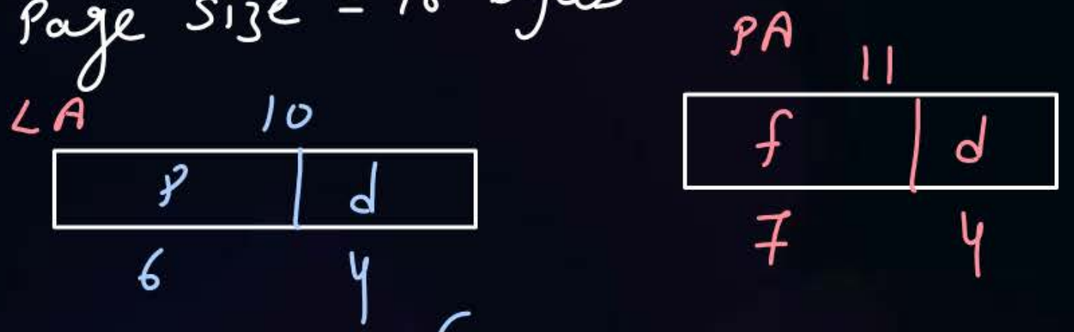
ex:-

Process size = 1kB = 2^{10} B

MM size = 2kB = 2^{11} B

Assumption 1:-

Page size = 16 Bytes

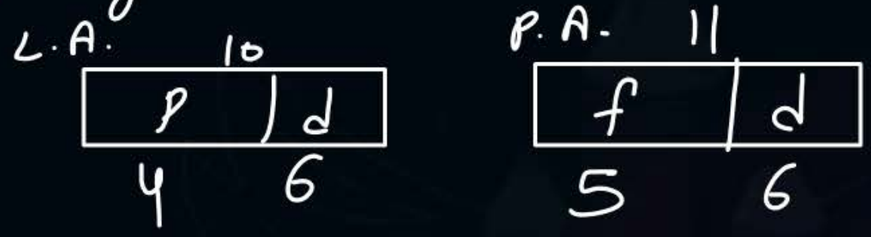


P.T. Size = 2^{6} * 7 bits

= 448 bits

Assumption 2:-

Page size = 64B



P.T. size = 2^{4} * 5 bits

= 80 bits

when Page size increases, P.T. size decreases

$$\text{P.T. size} \propto \frac{1}{\text{Page size}}$$

{ Process size is constant

Topic : Question

[GATE-2021]

#Q. Which one of the following statements is FALSE?

A The TLB performs an associative search in parallel on all its valid entries using page number of incoming virtual address.

B If the virtual address of a word given by CPU has a TLB hit. but the subsequent search for the word results in a cache miss, then the word will always be present in the main memory.

C The memory access time using a given inverted page table is always same for all incoming virtual addresses.

D In a system that uses hashed page tables, if two distinct virtual addresses VI and V2 map to the same value while hashing, then the memory access time of these addresses will not be the same.

Topic : Question

[GATE-1998]

#Q. If an instruction takes i microseconds and a page fault takes an additional j microseconds, the effective instruction time if on the average a page fault occurs every k instruction is:

A $i+\frac{i}{k}$

B $i+(j\times k)$

C $\frac{i+j}{k}$

D $(i+j)\times k$

Topic : Question

[GATE-1996]

#Q. A demand paged virtual memory system uses 16 bit virtual address, page size of 256 bytes, and has 1 Kbyte of main memory. page replacement is implemented using the list, whose current status (page number is decimal) is

For each hexadecimal address in the address sequence given below,

010D, 10FF, 11 BO

indicate

i. the new status of the list
ii. page faults, if any, and
ii. page replacements, if any.

2 mins Summary

Topic Access Time in Multilevel Paging

Topic Inverted Paging

Topic Hashed Page Table

Happy Learning

THANK - YOU