CS & IT ENGINEERING

Operating System

Memory Management

Lecture -6

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Virtual Memory

Topic Page Replacement Policies

Topics to be Covered

Topic Multilevel Paging

Topic Inverted Paging

Topic : Question

[GATE-2016]

#Q. Consider a computer system with ten physical page frames. The system is provided with an access sequence $a_1, a_2, \ldots, a_{20}, a_1, a_2, \ldots, a_{20}$ where each ai number. The difference in the number of page faults between the last-in-first-out page replacement policy and the optimal page replacement policy is _____ 1

LIFO = 20 + 11 = 31

optimal = 20 + 10 = 30

1

#Q. Assume that a main memory with only 4 pages, each of 16 bytes, is initially empty. The CPU generates the following sequence of virtual addresses and uses the Least Recently Used (LRU) page replacement policy.

0, 4, 8, 20, 24, 36, 44, 12, 68, 72, 80, 84, 28, 32, 88, 92

0 1 2 0 4 5 1 2 5

How many page faults does this sequence cause? What are the page numbers of the pages present in the main memory at the end of the sequence?

$$P = \left\lfloor \frac{V.A.}{\text{Page size}} \right\rfloor$$

0	0	0	0	0	0	2
	1	1	1	5	5	5
		2	2	2	1	1
			4	4	4	4

A 6 and 1,2,3,4

B 7 and 1,2,4,5

C 8 and 1,2,4,5

D 9 and 1,2,3,5

Topic : Let's Take a Simple Example

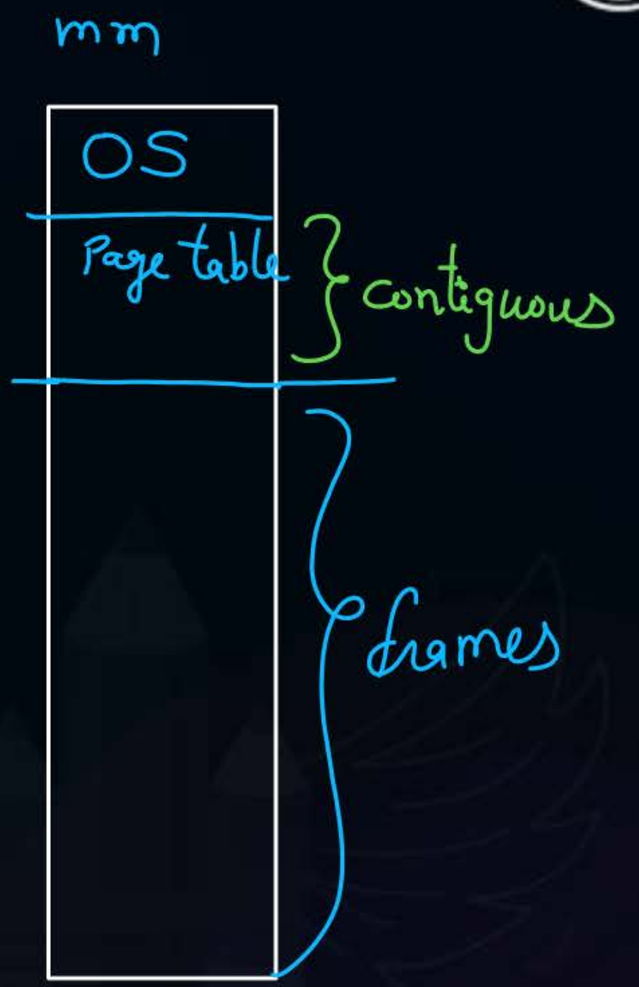
Process Size = 32B

Page size = 4B

} no. of pages = $\frac{32B}{4B} = 8$

Page table entry size = 1B

Page Table Size = ? = $8 * 1B = 8B$



Topic : Now What?

Can you keep entire page table in single page? ⇒ no

If not then how to access the specific entry?

Topic : Page Table in Memory

mm (16 frames)

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

Page 0 — Page Table

000	0101
001	1001
010	0110
011	0010
100	1111
101	1101
110	0011
111	1011

Page 1

8 bytes

P.T. of P.T. (outer P.T.)

0	0100
1	1100

P.T. Page 0

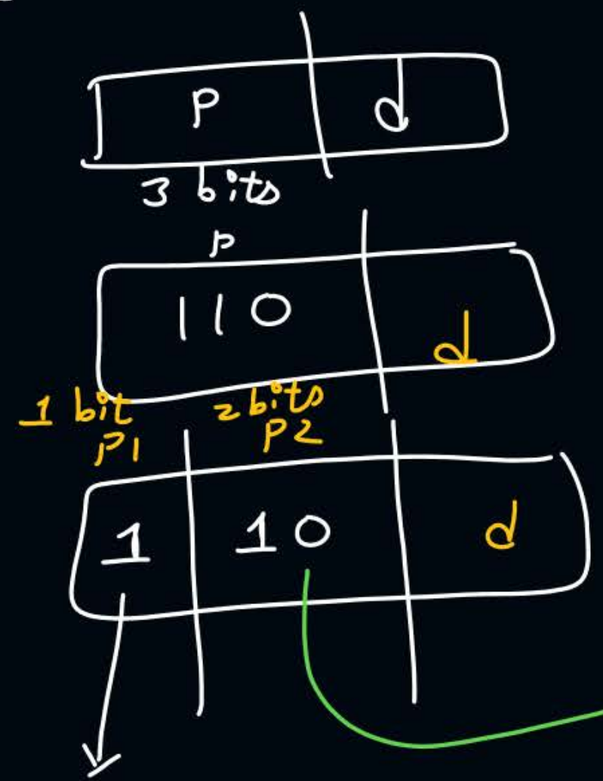
00	0101
01	1001
10	0110
11	0010

P.T. Page 1

00	1111
01	1101
10	0011
11	1011

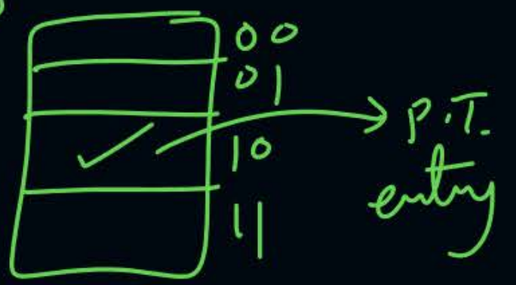
0000	
0001	
0010	Page 011
0011	Page 110
0100	P.T. Page 0
0101	Page 000
0110	Page 010
0111	
1000	
1001	Page 001
1010	
1011	Page 111
1100	P.T. Page 1
1101	Page 101
1110	
1111	Page 100

Assume L.A.



search in outer P.T. ⇒ which page of P.T. page to be accessed

→ frame no. ⇒ get P.T. page

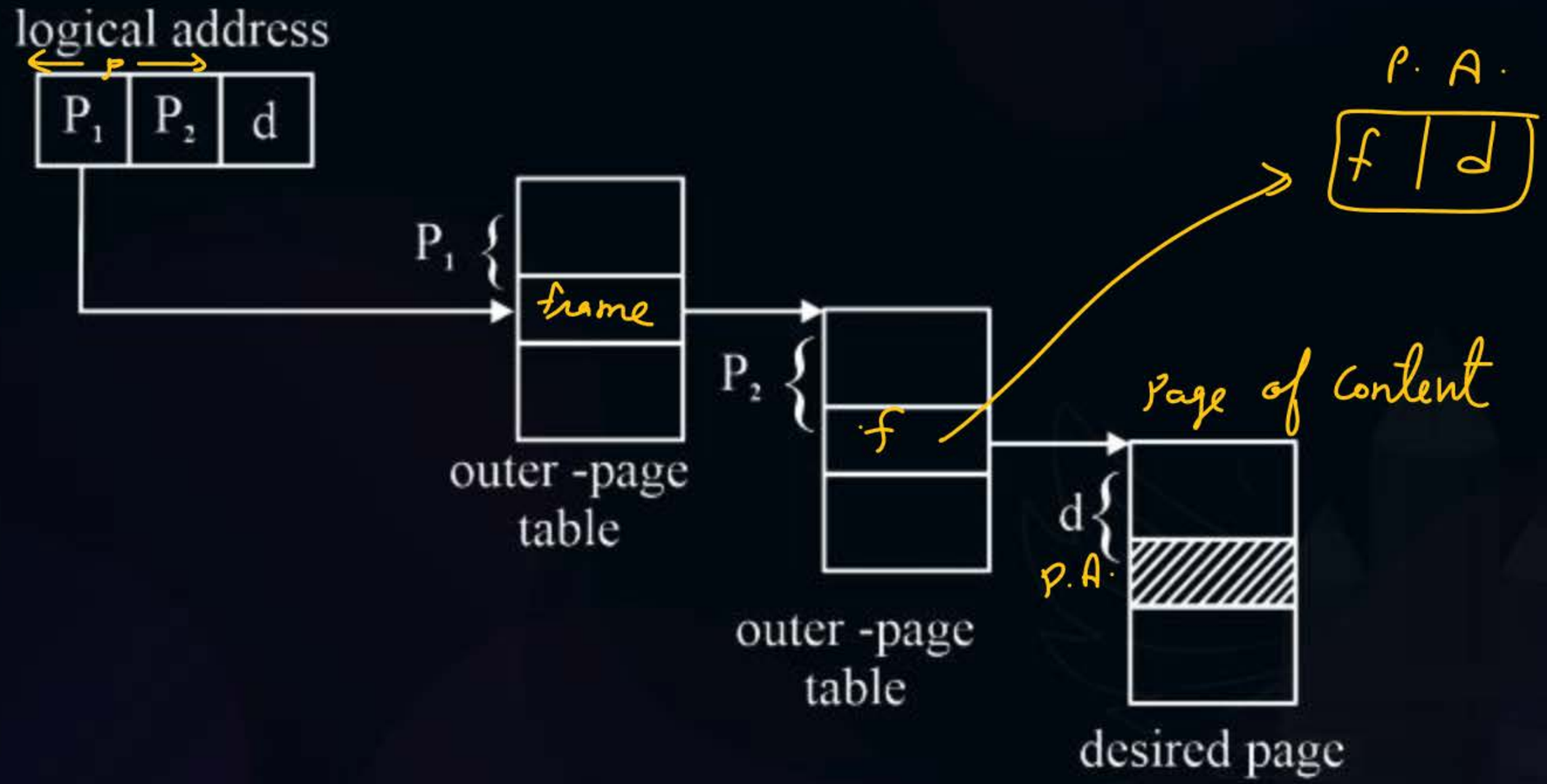


$$\text{no. of P.T. entries per page} = \frac{\text{Page size}}{\text{P.T. entry size}}$$

$$= \frac{4B}{1B}$$

$$= 4 = 2^2$$

Topic : Multilevel Paging



Topic : Multilevel Paging

introduced to save space in mm by keeping a few required page table pages only, in mm and remaining in disk.

Increase levels of page table paging until outer most page table is stored in single page.

example:-

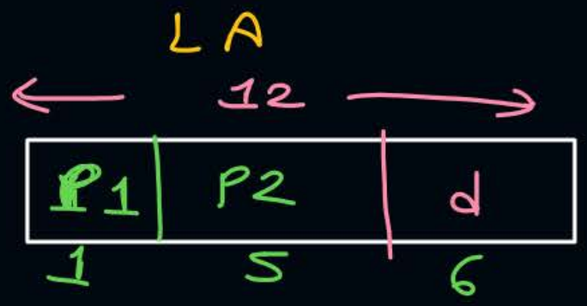
Page size = 4 bytes

P.T. entry size = 1 B

no. of P.T. entries per page = $\frac{4B}{1B}$

$= 4 = 2^2$

Ex:-



Page size = 64 B = 2^6 B ⇒ d = 6 bits

L.A. = 12 bits

P.T. entry = 2 bytes

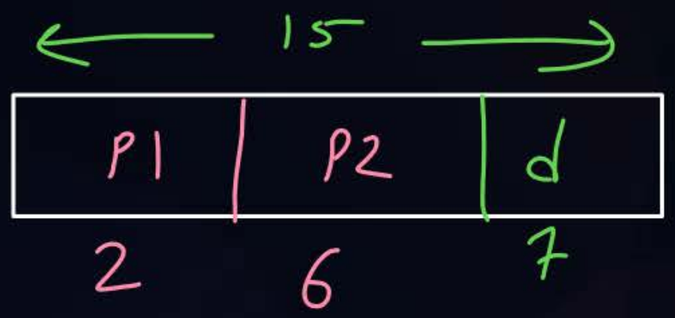
No. of P.T. entries per page = $\frac{64\,B}{2B}$

= 32 = 2^5

Topic : Question

VA = 15 bits
Page size = 128 bytes $= 2^7$ B $\Rightarrow$ d = 7 bits
Page table entry size = 2 bytes

Number of levels in multilevel page 2 ?



no. of P.T. entries per page $= \frac{128\,B}{2\,B}$

$= 64$

$= 2^6$

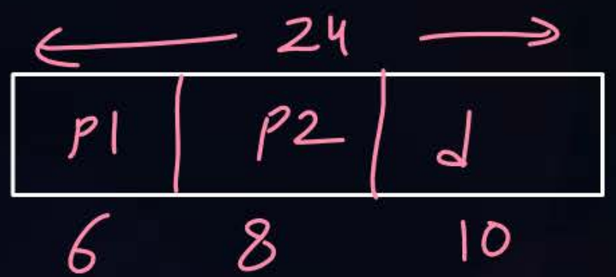
Topic : Question

VA = 24 bits

Page size = 1Kbytes ⇒ d = 10 bits

Page table entry size = 4 bytes

Number of levels in multilevel page 2 ?



no. of P.T. entries per page = $\frac{1KB}{4B}$

$= 2^8$

Topic : Question

VA = 29 bits

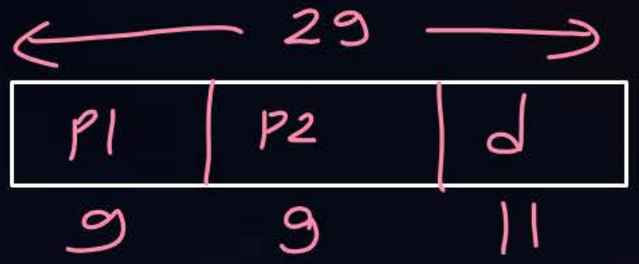
Page size = 2Kbytes = 2^{11} B

Page table entry size = 4 bytes

no. of P.T. entries per page = $\frac{2^{11}\text{ B}}{4\text{ B}}$

$= 2^9$

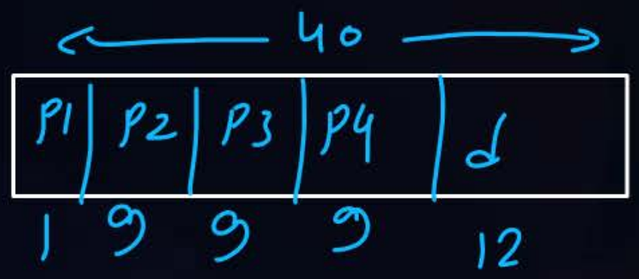
Number of levels in multilevel page 2 ?



Topic : Question

VA = 40 bits
Page size = 4Kbytes $= 2^{12}$ B $\Rightarrow$ d = 12 bits
Page table entry size = 8 bytes

Number of levels in multilevel page 4 ?



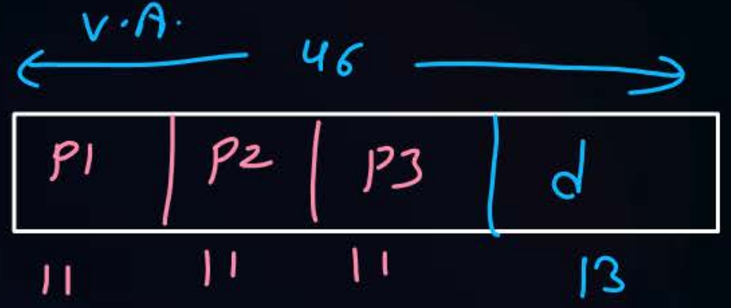
no. of P.T. entries per page $= \frac{2^{12}\text{ B}}{8\text{ B}} = 2^9$

Topic : Question

Ans = 3

#Q. Consider a virtual memory system with physical memory of 8GB, a page size of 8KB and 46-bit virtual address. Assume every page table exactly fits into a single page. If page table entry size is 4B then how many levels of page tables would be required.

2^{13} B



no. of P.T. entries per page $= \frac{8\,KB}{4B}$

$= 2K = 2^{11}$

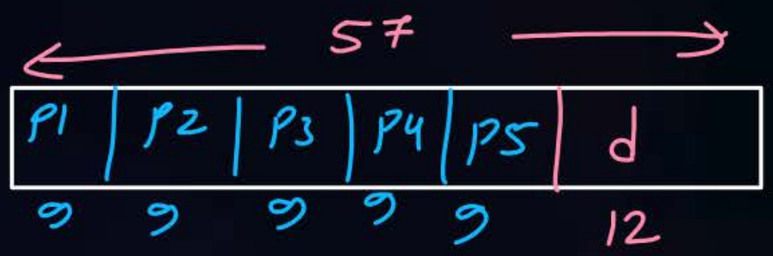
Topic : Question

GATE-2024

#Q. Consider a computer system with 57-bit virtual addressing using multi-level tree-structured page tables with L levels for virtual to physical address translation. The page size is 4KB(1KB=1024B) and a page table entry at any of the levels occupies 8 bytes.

The value of L is ___5___?

$4KB \rightarrow 2^{12}B \Rightarrow d = 12$



no. of P.T. entries per page $= \frac{2^{12}B}{8B} = 2^9$

⇒ Note :-

sometimes to restrict number of levels, outer P.T. may store entries more than one page size.

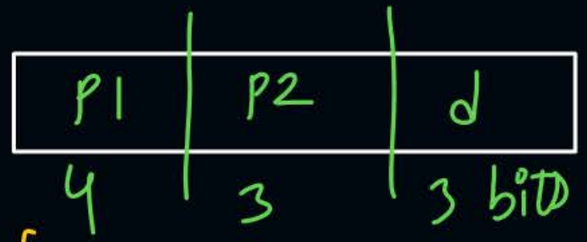
exc :-

L.A. = 10 bits

Page size. = 8 bytes

Page Table entry = 1 B

entries per page = $\frac{8B}{1B}$ = 8



outer P.T. can not be stored in a frame and stored in separate area.

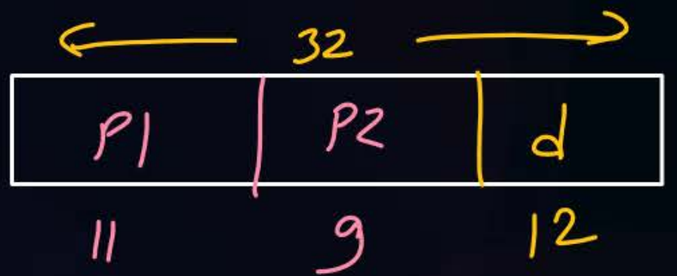
Topic : Question

[GATE-2025]

Ans = 11

#Q. A computer system supports a logical address space of 2^{32} bytes. It uses two-level hierarchical paging with a page size of 4096 bytes. A logical address is divided into a b-bit index to the outer page table, an offset within the page of the inner page table, and an offset within the desired page. Each entry of the inner page table uses eight bytes. All the pages in the system have the same size. The value of b is __11__?

L.A. = 32 bits



Page = 4096 B = 2^{12} B $\Rightarrow$ d = 12 bits

P.T. entries Per page = $\frac{4096\ B}{8\ B}$

$= 2^9$

Topic : Question

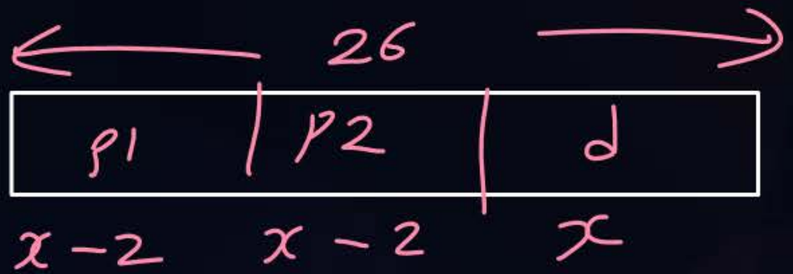
VA = 26 bits

Page table entry size = 4 bytes

2-level paging

Outer page table fits into a page exactly

Page size = __1__ Kbytes?



$(x-2)+(x-2)+x = 26$ bits

$3x-4 = 26$

$x = 10$ bits

assume,

Page size $= 2^x$ Bytes $= 2^{10}$ B $= 1$ KB

$\Downarrow$

$d = x$ bits

P.T. entries per page $= \frac{2^x B}{4B}$

$= \frac{2^x}{2^2}$

$= 2^{x-2}$

$\Downarrow$

P.T. searching $= x-2$

Topic : Question

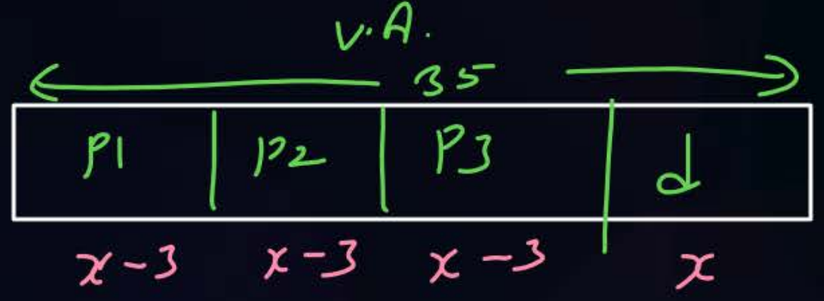
VA = 35 bits

Page table entry size = 8 bytes

3-level paging

Outer page table fits into a page exactly

Page size = __2__ Kbytes?



$3(x-3)+x=35$

$4x-9=35$

$4x=44$

$x=11 \Rightarrow$ Page $=2^{11}B=2KB$

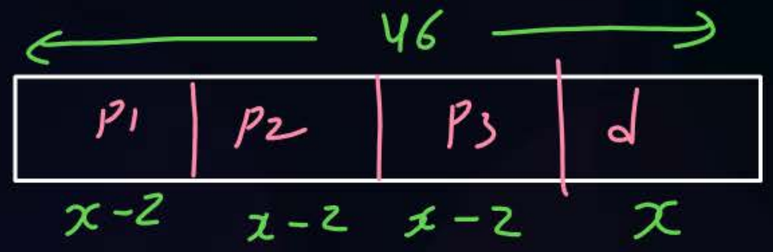
Topic : Question

Ans = 8

GATE - PYQ

#Q. A computer uses 46–bit virtual address, 32–bit physical address, and a three–level paged page table organization. The page table base register stores the base address of the first–level table (T1), which occupies exactly one page. Each entry of T1 stores the base address of a page of the second–level table (T2). Each entry of T2 stores the base address of a page of the third–level table (T3). Each entry of T3 stores a page table entry (PTE). The PTE is 32 bits in size. What is the size of a page in KB in this computer?

4 Bytes



$3(x-2)+x = 46$

$x = 13$

Page size $= 2^{13}B = 8KB$

Ques) L.A. = 36 bits

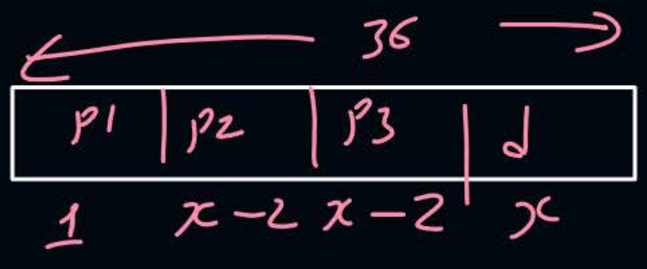
P.T. entry = 4 Bytes

3 level paging

outermost Page table can be searched using = 1 bit

page size = 8 KB ?

Soln



$x = 13$

page size $= 2^{13}$ B $= 8$ KB

Ques)

P.T. entry = 8B

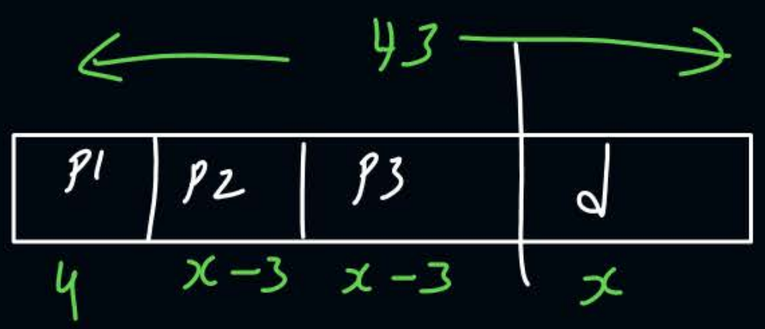
3-Level paging

L.A. = 43 bits

outer P.T. stores (16) P.T. entries only.

2^4 ⇒ P1 = 4 bits

Page size = 32 KB ?



$4 + (x-3)2 + x = 43$

$3x = 45$

$x = 15$

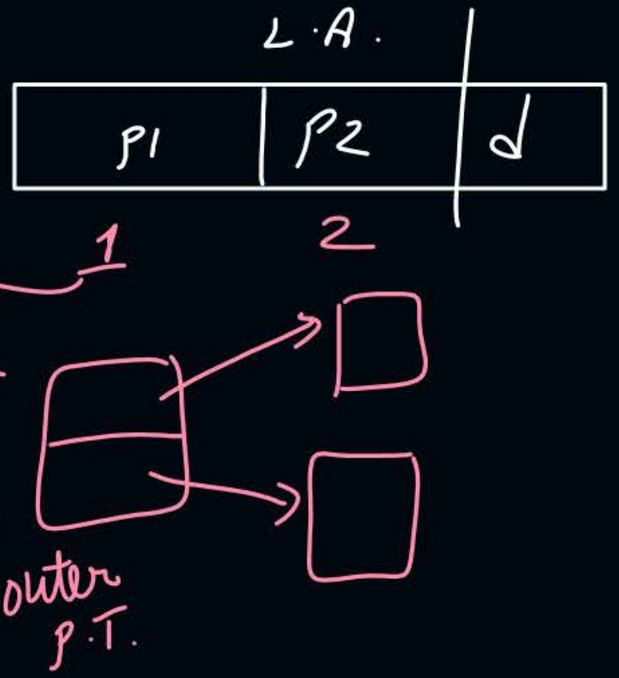
Page $= 2^{15}$ B $= 32$ KB

P.T. size across all level:-

ex:-

Page = 4B
P.T. entry = 1B
} entries per page = $\frac{4B}{1} = 4 = 2^2$

2 - level paging



No. of pages needed to store outer most P.T. = 1

—11— inner P.T. = $2^{p1} = 2^1 = 2$

Total P.T. = 3 pages

P.T. size across levels = 3 * page size
= 3 * 4B
= 12B

Topic : Question

Size of page tables across all levels?

P1	P2	D
1	8	10

Page size $= 2^{10}$ B $= 1$ KB

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

—11— Inner P.T. $= 2^{1} = 2$

= 3 pages

P.T. size = 3 * 1KB

= 3 KB Ans.

Topic : Question

Ans = 20 KB

Size of page tables across all levels?

P1	P2	D
2	9	12

→ Page size = 2^{12} B = 4 KB

no. of pages to store outer P.T. = 1

—— 11 —— Inner P.T. = $2^2 = 4$

Total = 5 pages

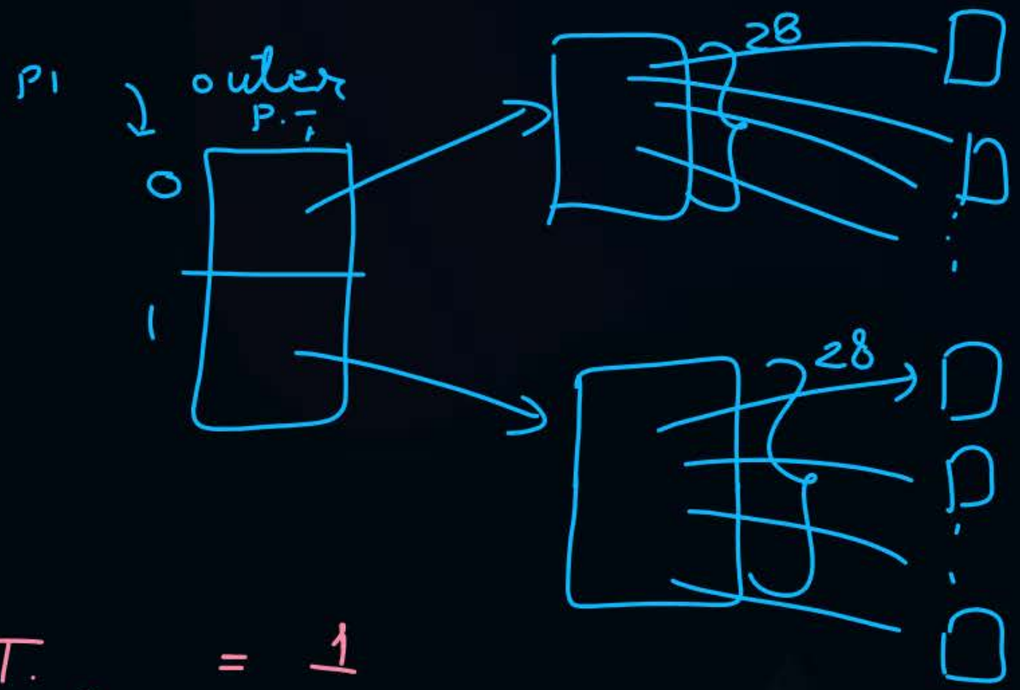
P.T. size across all level = 5 * 4KB

= 20 KB

Topic : Question

Size of page tables across all levels?

P1	P2	P3	D
1	8	8	10



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

—11— middle P.T. $= 2^{P_1} = 2^1 = 2$

—11— Innermost P.T. $= 2^{P_1} * 2^{P_2} = 2^1 \cdot 2^8 = 512$

Total = 515

P.T. size = 515 * 1KB

= 515 KB

32-bits OS

⇓

max V.A. for any process = 32 bits

max process size = 4 GB

64-bits OS

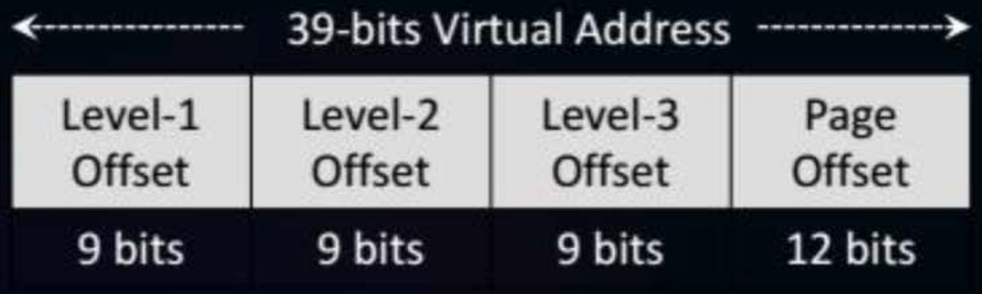
⇒ 64 bits

⇒ 2^{64} B

Topic : Question

GATE - PYQ

#Q. [redacted] Consider a three-level page table to translate a 39-bit virtual address to a physical address as shown below:



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 4108 KB across all levels?

2^{31} B

31 - bits

No. of pages for outer most P.T. = 1

—— 11 —— middle P.T. = $2^1 = 2$

—— 11 —— Inner most P.T. = $2^1 * 2^9 = 1024$

Total = 1027

P.T. size = 1027 * 4B

= 4108 kB

Topic : Question

H.W.

[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 _____?

2 mins Summary

Topic Multilevel Paging

Topic Inverted Paging

Happy Learning

THANK - YOU