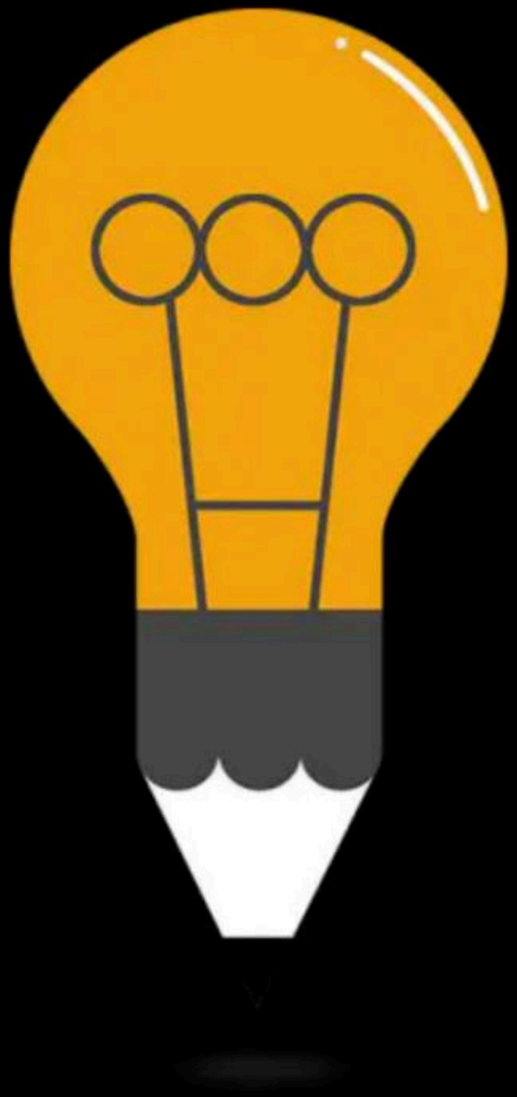
unacademy

Paging & Address Calculation

Comprehensive Course on Operating System for GATE - 2024/25

Vishvadeep Gothi • Lesson 25 • May 17, 2023

unacademy



Operating System

Paging

By: Vishvadeep Gothi

MMT

Contiguous

non-contiguous

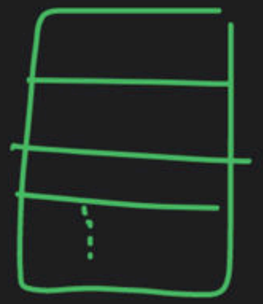
non-contiguous MMT

Paging

each process divided into equal size partitions

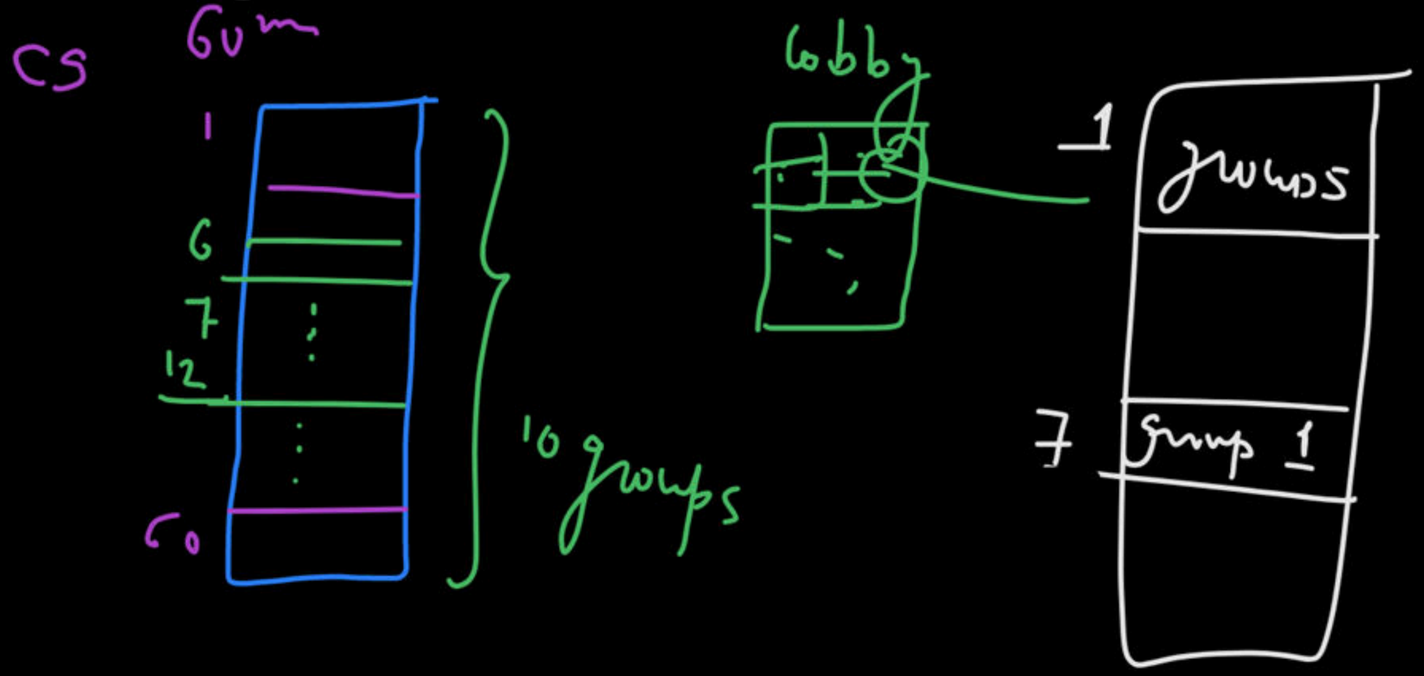
Segmentation

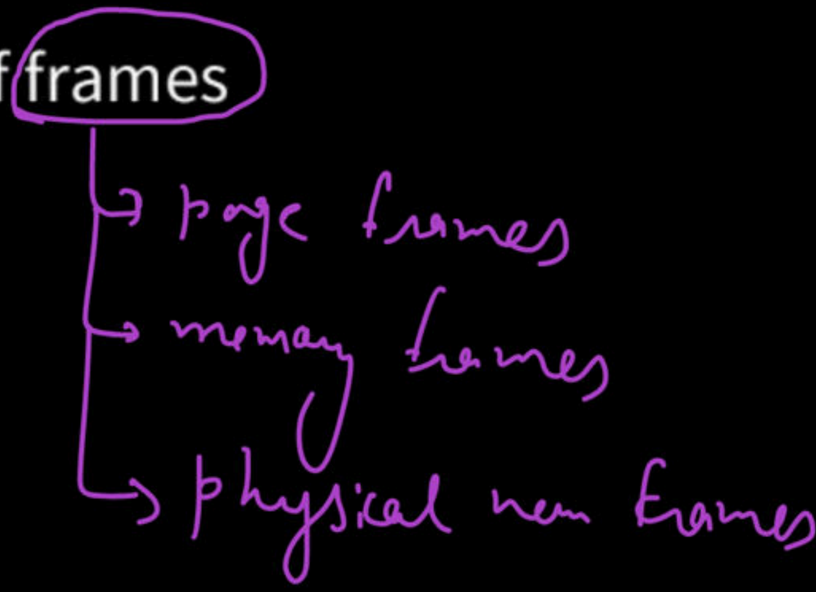
each process divided into variable size partitions



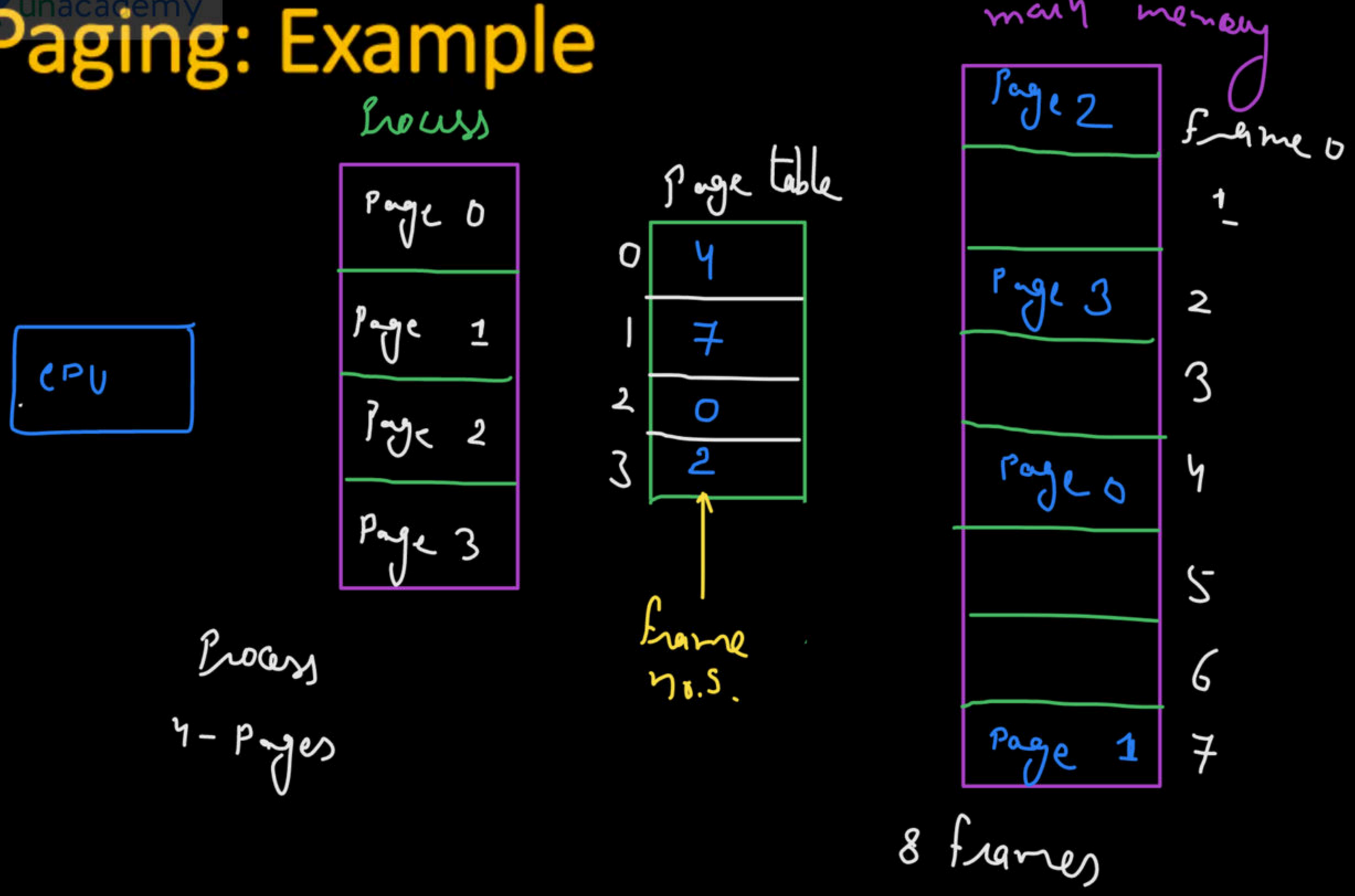
Paging

- Process is divided in equal size of pages
- Physical memory is divided in same equal size of frames
- Pages are scattered in frames





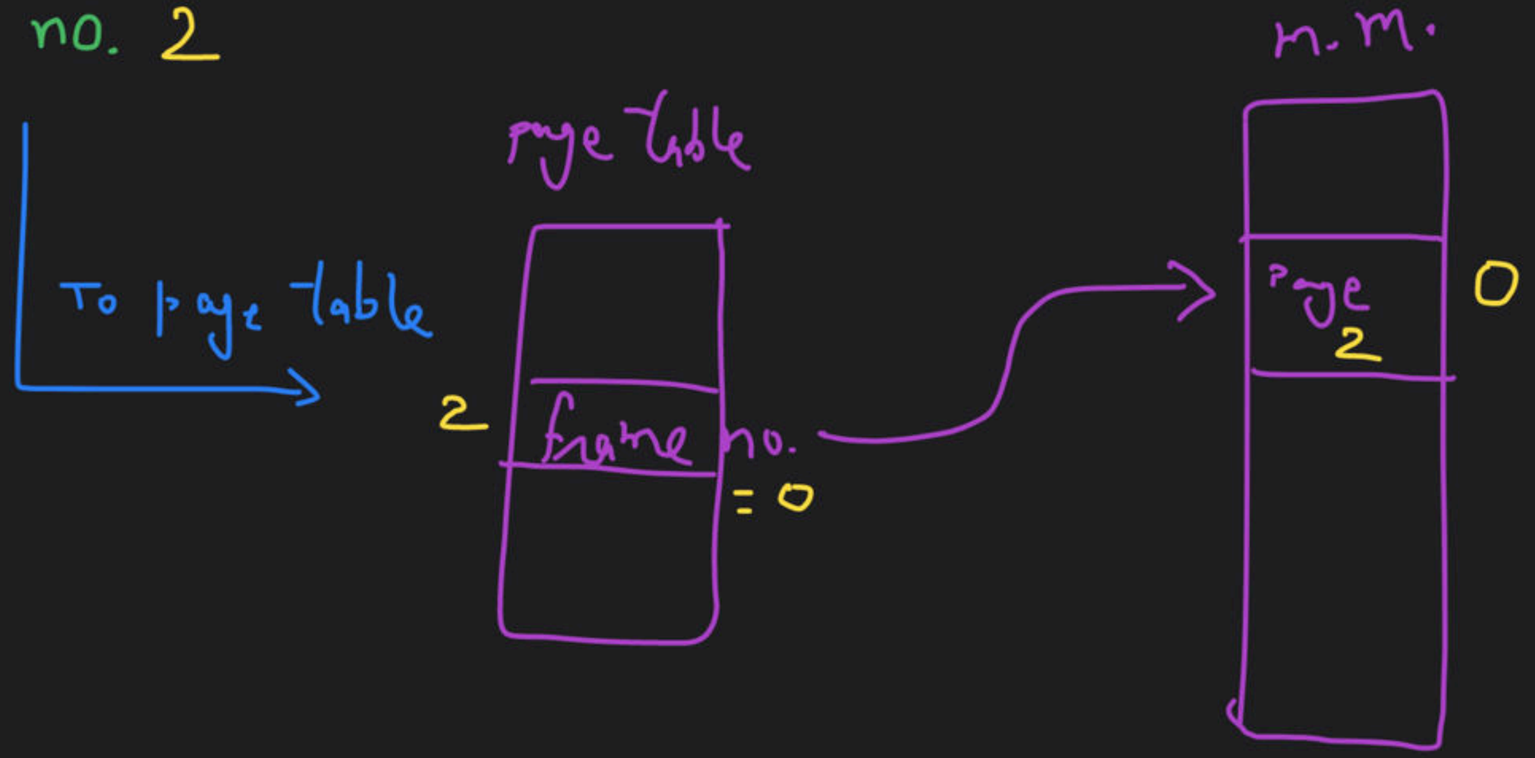
Paging: Example



CPU requests for main memory content

↓

Page no. 2

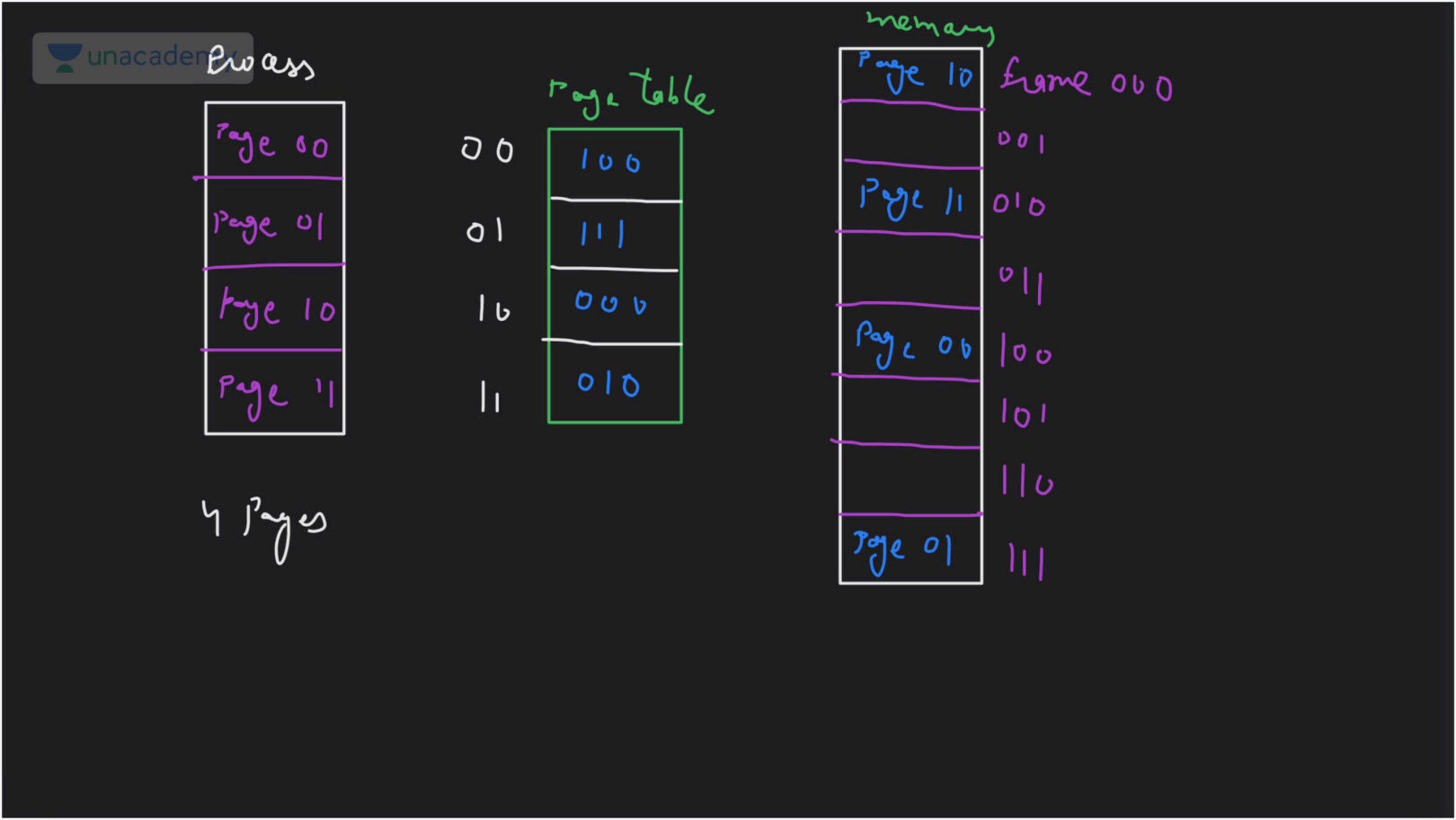


Paging

- Process is divided in equal size of pages
- Physical memory is divided in same equal size of frames
- Processor will have a view of process and its pages
- Pages are scattered in frames
- Page table is used to map a process page to a physical frame
- Number of entries in page table = Number of pages in process
- OS maintains a page table for each process

Each P.T. entry contains frame no. + extra bits

Page table size = no. of entries in P.T. * 1 entry size
= no. of pages in process * 1 entry size


unacademy
Process
Page 00
Page 01
Page 10
Page 11
4 Pages
Page table
00
01
10
11
100
111
000
010
Memory
Page 10
Frame 000
001
Page 11
010
011
Page 00
100
101
110
Page 01
111

Example: Page size = 4 Bytes

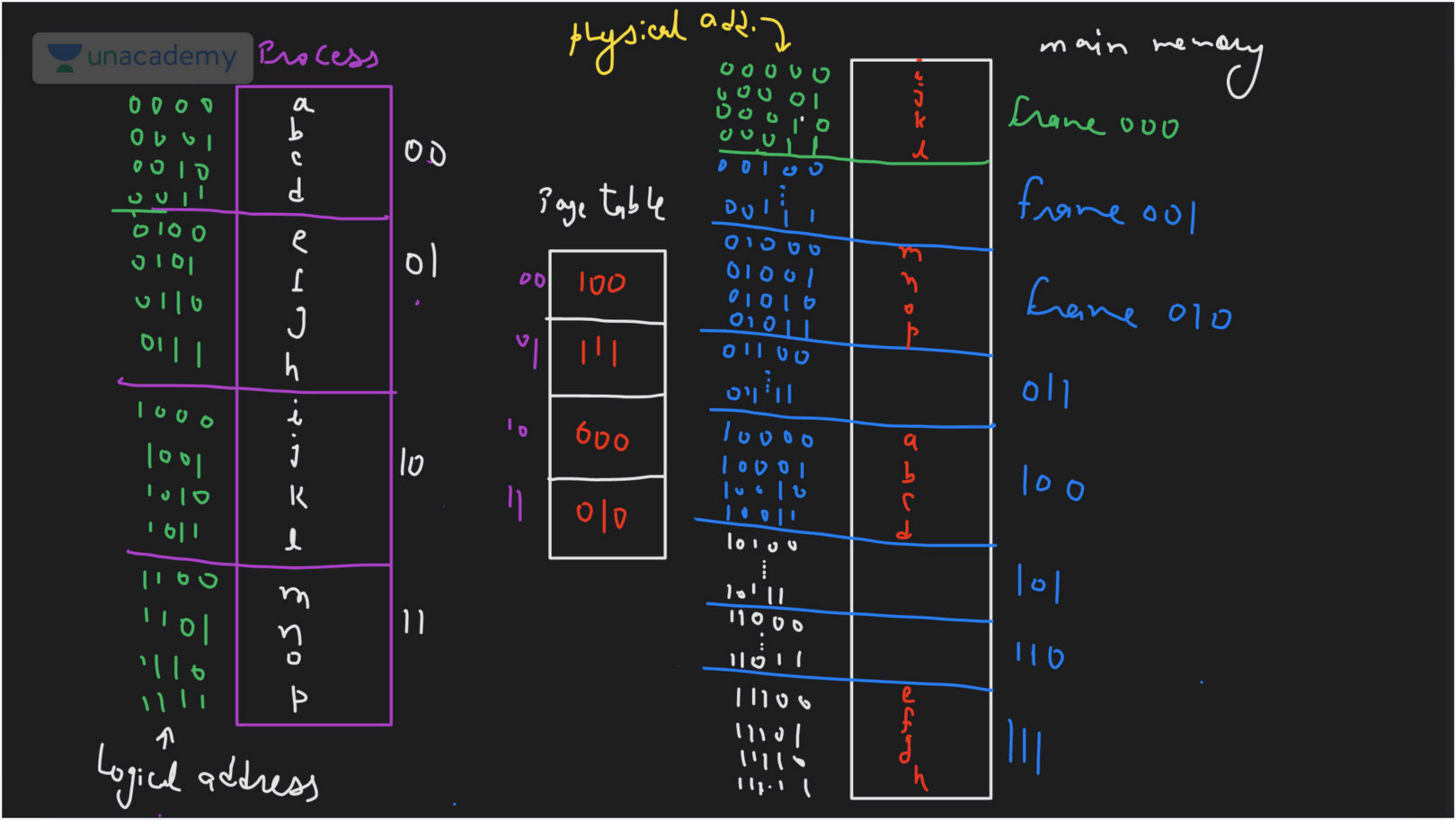
Process size = 4 pages = 4 * 4 bytes = 16 bytes

Main memory = 8 frames = 8 * 4 bytes = 32 bytes ⇒ mm address = 5-bits

---

Process size = logical address space

m.m. size = physical address space


unacademy
Process
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
00
01
10
11
Logical address
Page table
00
100
01
111
10
000
11
010
physical add.
main memory
frame 000
frame 001
frame 010
011
100
101
110
111

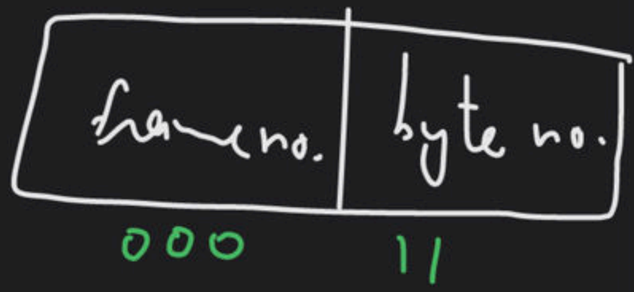
⇒ CPU always generates logical address (virtual address)

L.A. ⇒ 10 11 ⇒ byte 11 of page 10

10 → page no.

11 → byte no.

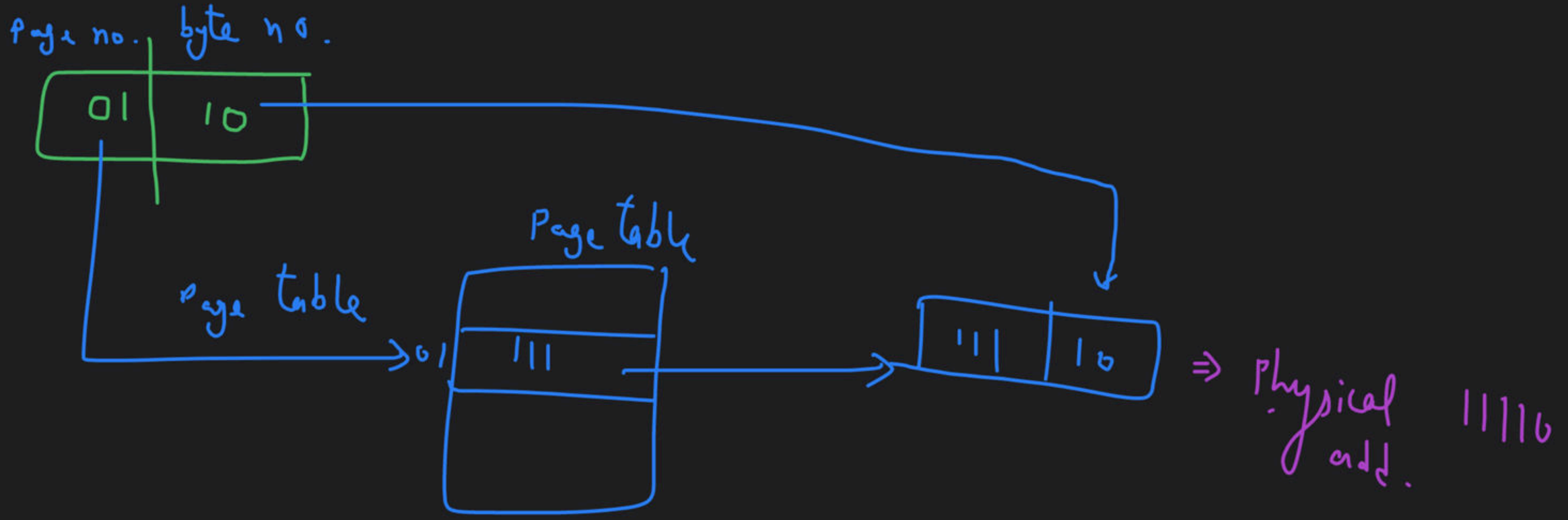
page no. → search page no. 10 in Page table ⇒ frame 000 ⇒ goto frame no. 000 & access byte 11



physical add.

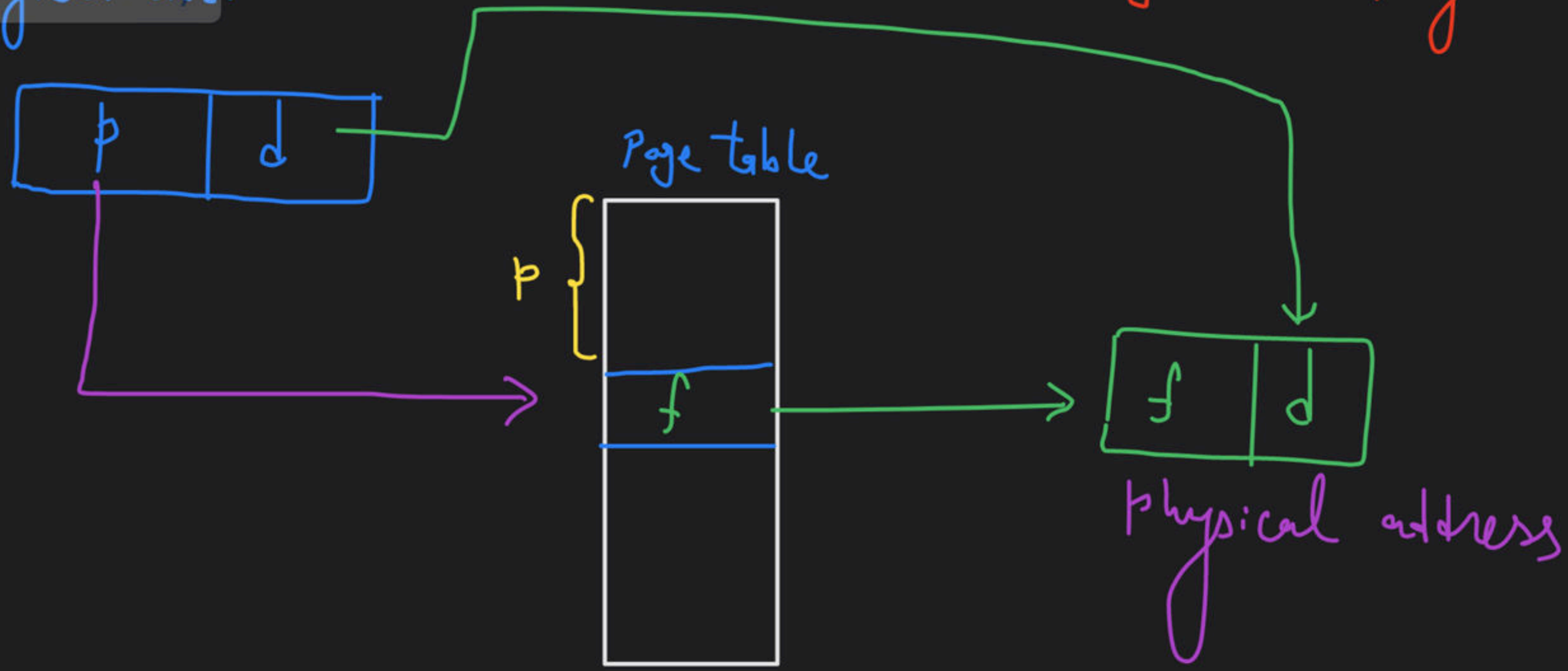
ex. (2)

CPU generates logical add. 0110



logical add.

logical to physical address translation



p = page no.

d = offset (byte no./ line no.)

Ques

Process size 64 Bytes

Pages size = 4 bytes

no. of pages = ?

no. of bits in page no. = ?

$$\text{no. of pages} = \frac{\text{Process size}}{\text{Page size}}$$

$$= \frac{64\,B}{4\,B}$$

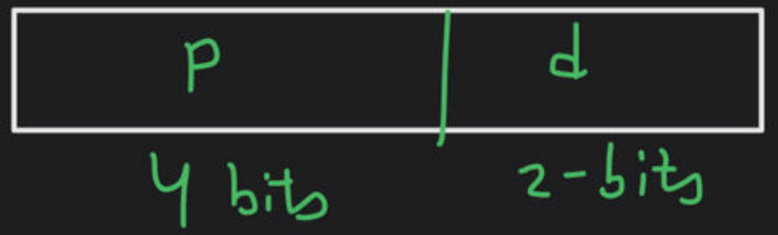
$$= 16 \text{ pages} \Rightarrow 2^4 \text{ Pages} \Rightarrow \text{page no.}$$

4-bits

no. of bits

byte no. = 2-bits

L.A.



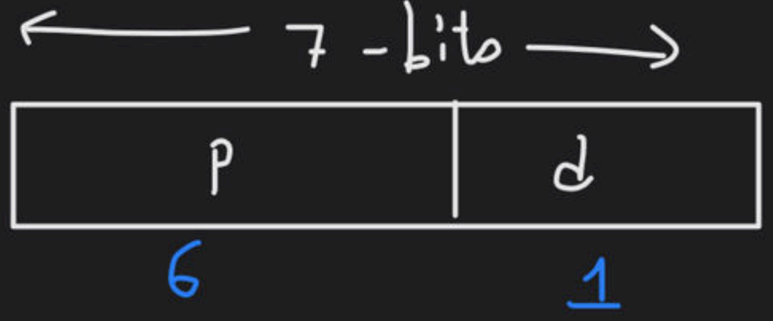
Ques

Process size = 128 Bytes $\Rightarrow 2^7$ bytes =

Page size = 2 byte = 2^1 byte

L.A. size = 7 -bits

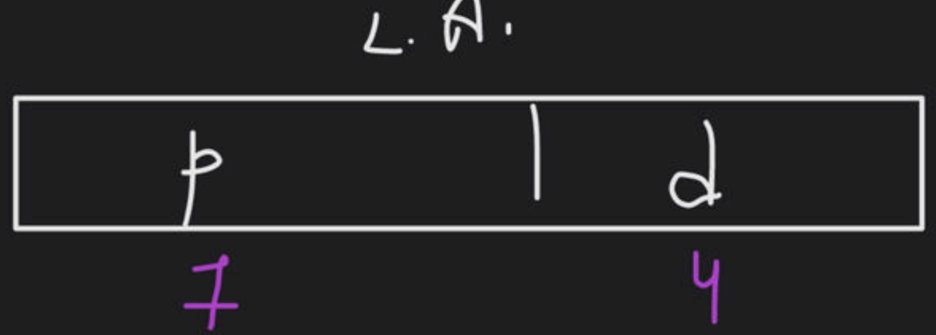
no. of pages = $\frac{128B}{2B} = 64$



no. of pages = 2^6 = 64 pages

Ques Consider a paging system with page size 16 bytes and a process having 128 pages.

Soln



128 * 16 bytes

$2^7 * 2^4$ bytes

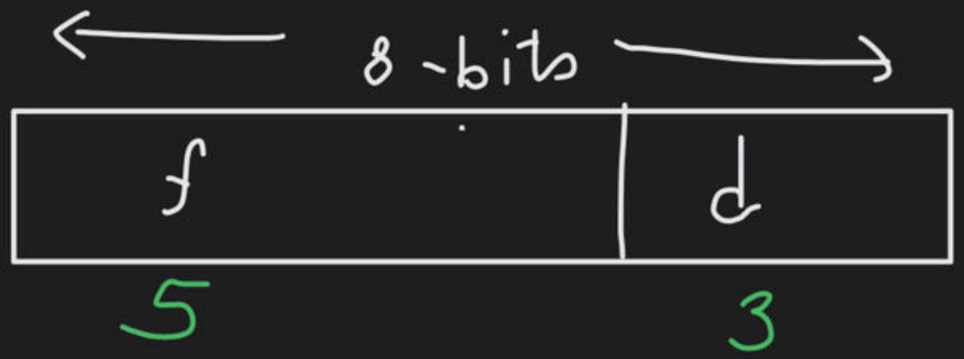
2^{11} bytes

L.A. Size = 11 bits

Process size = 2^{11} bytes = 2048 bytes = 2 kbytes

Ques Physical memory = 256 bytes ⇒ physical add. = 8-bits

Page Size = 8 bytes ⇒ 2^3 bytes ⇒ d = 3-bits



no. of frames = 2^5 = 32 frames

unacademy

$$K = 2^{10}$$

$$M = 2^{20}$$

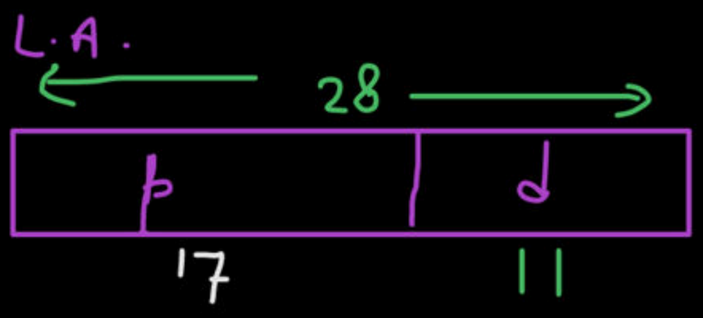
$$G = 2^{30}$$

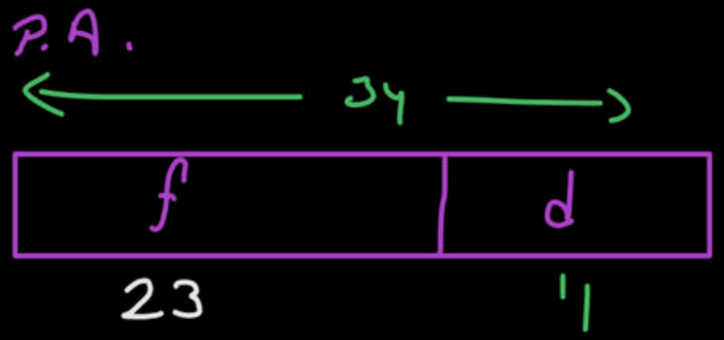
Question

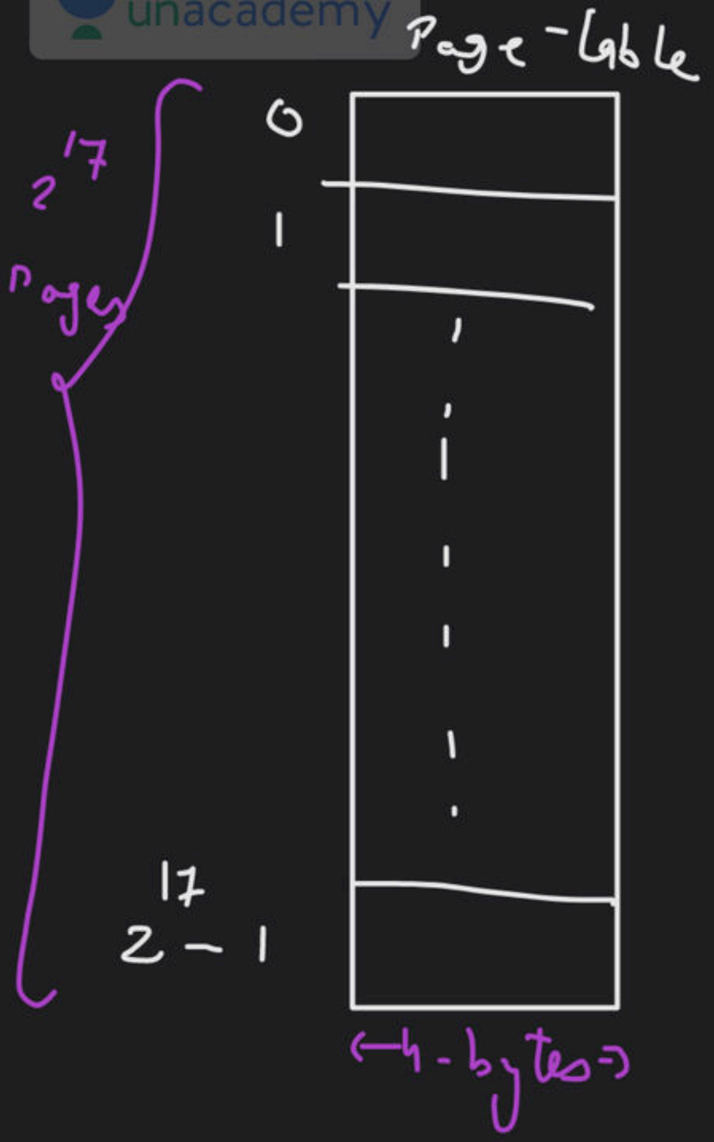
Consider a paged memory system where the logical address of 28 bits and physical address of 34 bits. If each page table entry is of 4 bytes and page size is 2KB then:

1. Number of pages in process? 2^{17}
2. Number of frames in main memory? 2^{23}
3. Number of bits for page number? 17 bits
4. Number of for frames? bits => 23 bits
5. Number of entries in page table? 2^{17}
6. Page table size?

2KB → 2^{11} Bytes ⇓ d = 11-bits







Page table size = no. of pages * P.T. entry size

$= 2^{17} * 4$ bytes

$= 2^{19}$ Bytes

$= 512$ kBytes

P.T. entry — 32-bits

frame no.	extra bits
← 23-bits →	← 9-bits →

Ques L.A. = 32-bits

P.A. = 34-bits

Page size = 16 Kbytes $\Rightarrow 2^{14}$ bytes

Ans : 12

no. of protectn-bits per page-table entry = ____ bits

in page table entry size = 4 bytes = 32 bits

Soln :-

P.T. entry

32-bits

frame no.	extra bits
20	12-bits

Physical add.

34-bits

f	d
20	14

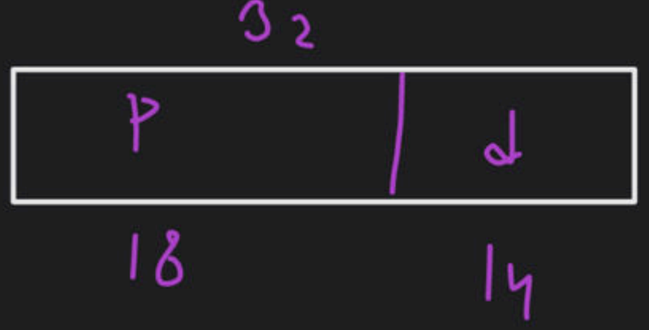
P.T. Size = _____ in last Questn ?

= no. of pages * P.T. entry size

= 2^{18} * 4 bytes

= 2^{20} bytes = 1MBytes

1A.



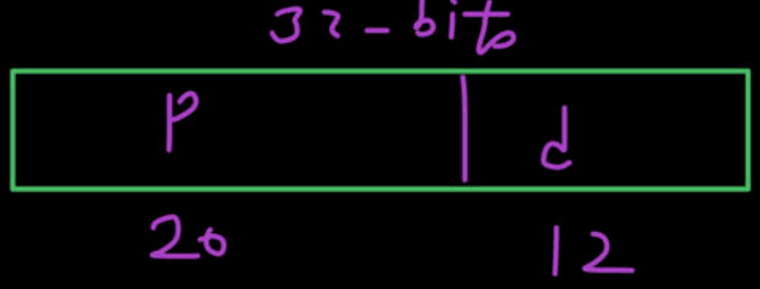
Question GATE-2001

→ physical add. = 26 bits

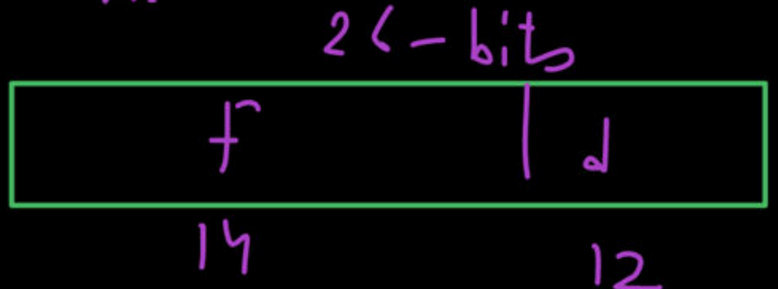
Consider a machine with 64 MB physical memory and a 32-bit virtual address space. If the page size s 4 KB, what is the approximate size of the page table?

1. 16 MB
2. 8 MB
3. 2 MB
4. 24 MB

L.A.



P.A.



P.T. size $\geq 2^{20} * 14$

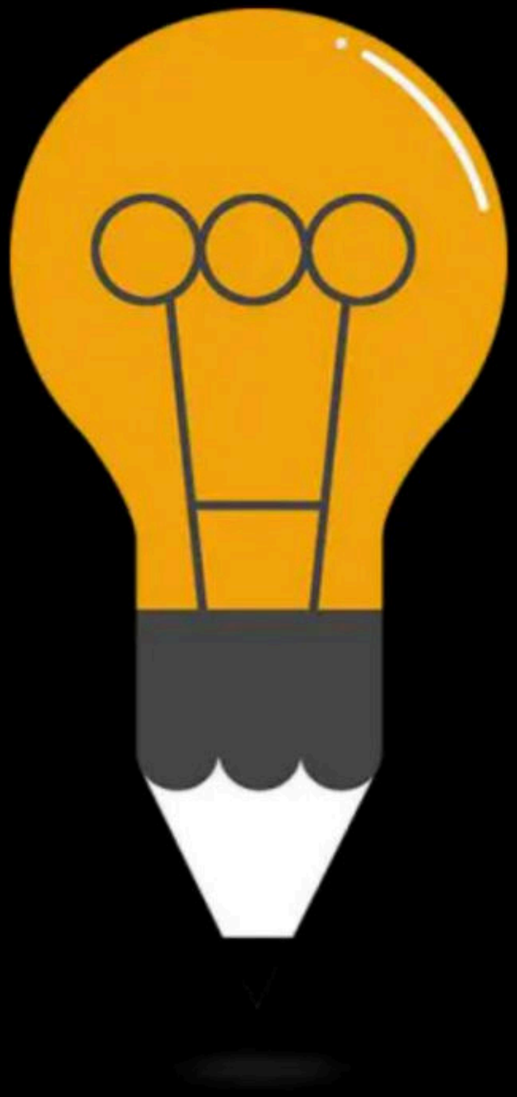
14 M bits $\cong$ 2 M bytes

Question GATE-2015

H. work

A computer system implements 8 kilobyte pages and a 32−bit physical address space. Each page table entry contains a valid bit, a dirty bit, three permission bits, and the translation. If the maximum size of the page table of a process is 24 megabytes, the length of the logical address supported by the system is _____ bits?

unacademy



DPP

By: Vishvadeep Gothi

Question 1

Consider a paged memory system where the process size is 16MB and main memory size is 4GB. The page size is 2KB.

1. Number of pages in process?
2. Number of frames in main memory?
3. Number of bits for page number?
4. Number of bits for frames?
5. Number of entries in page table?
6. Page table size?

Question 2

Consider a paged memory system where the process size is 128MB and main memory size is 2GB. The page size is 1KB.

1. Number of pages in process?
2. Number of frames in main memory?
3. Number of bits for page number?
4. Number of bits for frames?
5. Number of entries in page table?
6. Page table size?

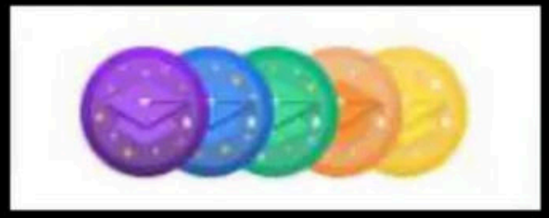
Question 3

Consider a paged memory system where the logical address is 25 bits and physical address is 33 bits. The page size is 4KB.

1. Number of pages in process?
2. Number of frames in main memory?
3. Number of bits for page number?
4. Number of bits for frames?
5. Number of entries in page table?
6. Page table size?

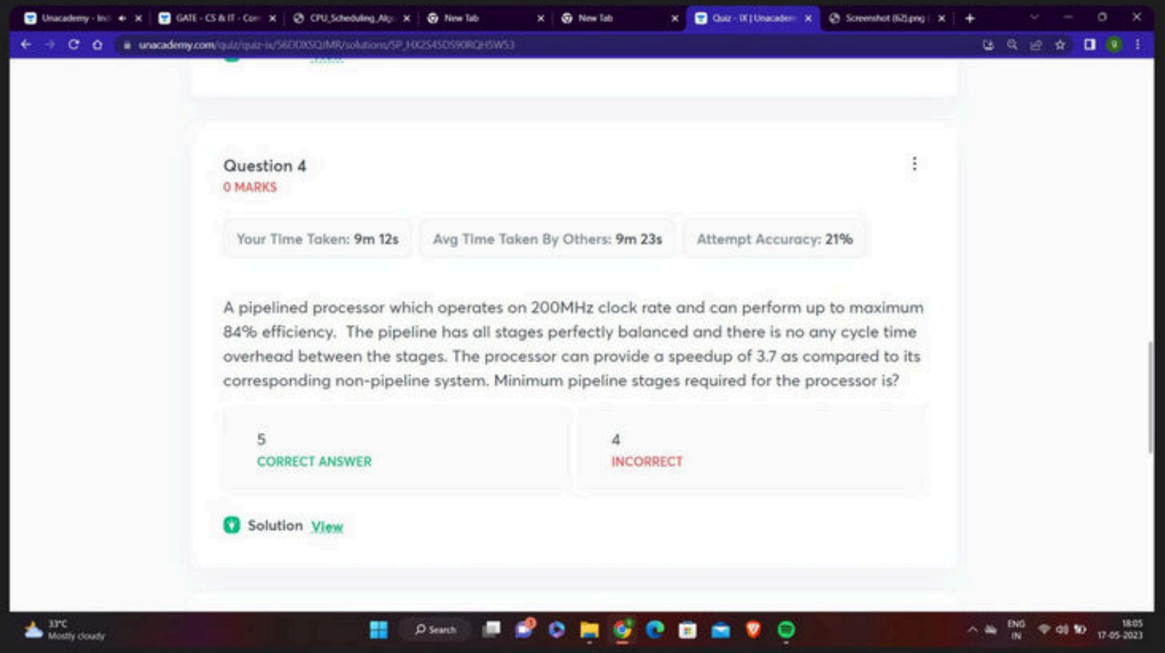
unacademy

Happy Learning.!



▲ 1 • Asked by Meghansh

Please help me with this doubt



Question 4

0 MARKS

Your Time Taken: 9m 12s | Avg Time Taken By Others: 9m 23s | Attempt Accuracy: 21%

A pipelined processor which operates on 200MHz clock rate and can perform up to maximum 84% efficiency. The pipeline has all stages perfectly balanced and there is no any cycle time overhead between the stages. The processor can provide a speedup of 3.7 as compared to its corresponding non-pipeline system. Minimum pipeline stages required for the processor is?

5 CORRECT ANSWER

4 INCORRECT

Solution View

speed up = efficiency * max speed up

$3.7 = 0.84 * k$

$k = \frac{3.7}{0.84}$

$= 4.4$

$k_{min} = 5$

$k \geq 4.4$