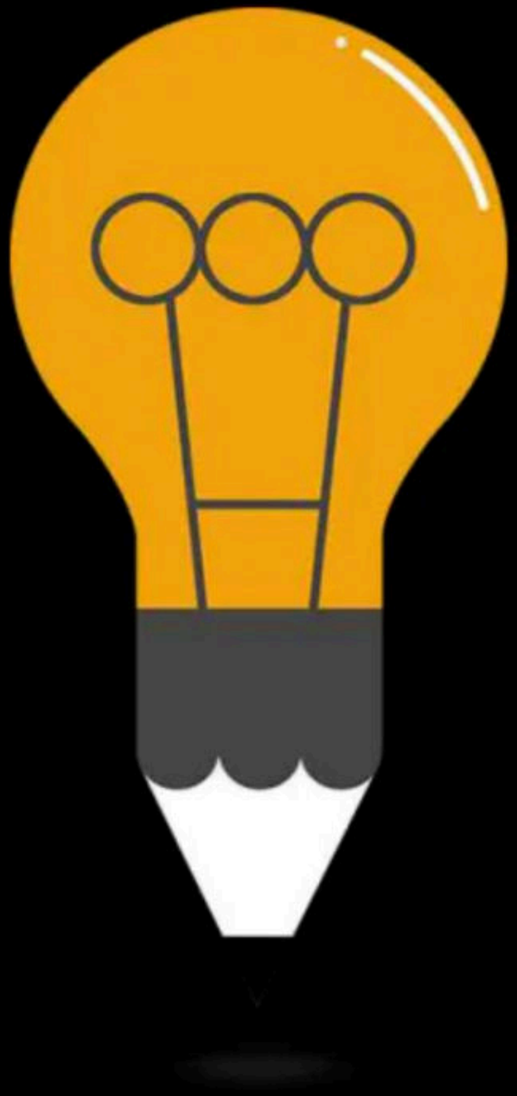
unacademy

Doubt Clearing Session & Questions on Paging

Comprehensive Course on Operating System for GATE - 2024/25

Vishvadeep Gothi • Lesson 37 • May 31, 2023

unacademy



Operating System
Doubts & Inverted Paging

By: Vishvadeep Gothi

▲ 1 • Asked by Siddhartha

sir, i want clarity.Sir,ptbr will not as it has current base address of page table of current process.Sir i want clarity on stack pointer?Sir , please explain.

Q.15 Which one of more of the following need to be saved on context switch form one thread (T1) of a process to another thread (T2) of the same process?

(a) Program counter
(b) Stack pointer
(c) General purpose register
(d) Page table base register

not shared

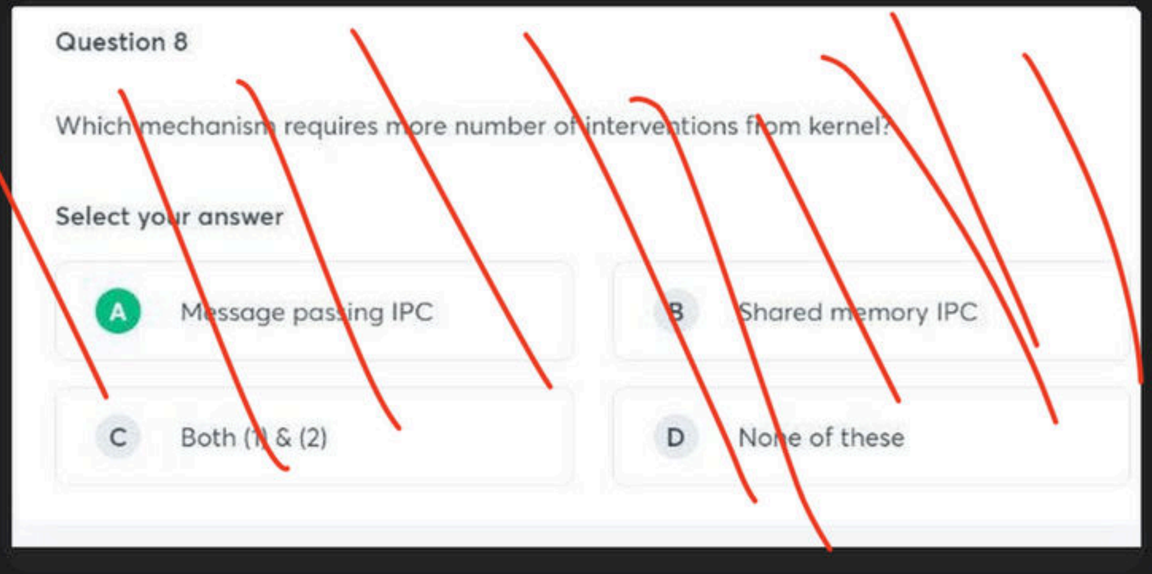
PC, ?

stack

unacademy

▲ 1 • Asked by Jai

Please help me with this doubt



unacademy

▲ 1 • Asked by Siddhartha

??

Q.14 Which one or more of the following option guarantee that a computer system transit from user mode to kernel mode

(a) Page fault
(b) System call
(c) Malloc call
(d) Function call

unacademy

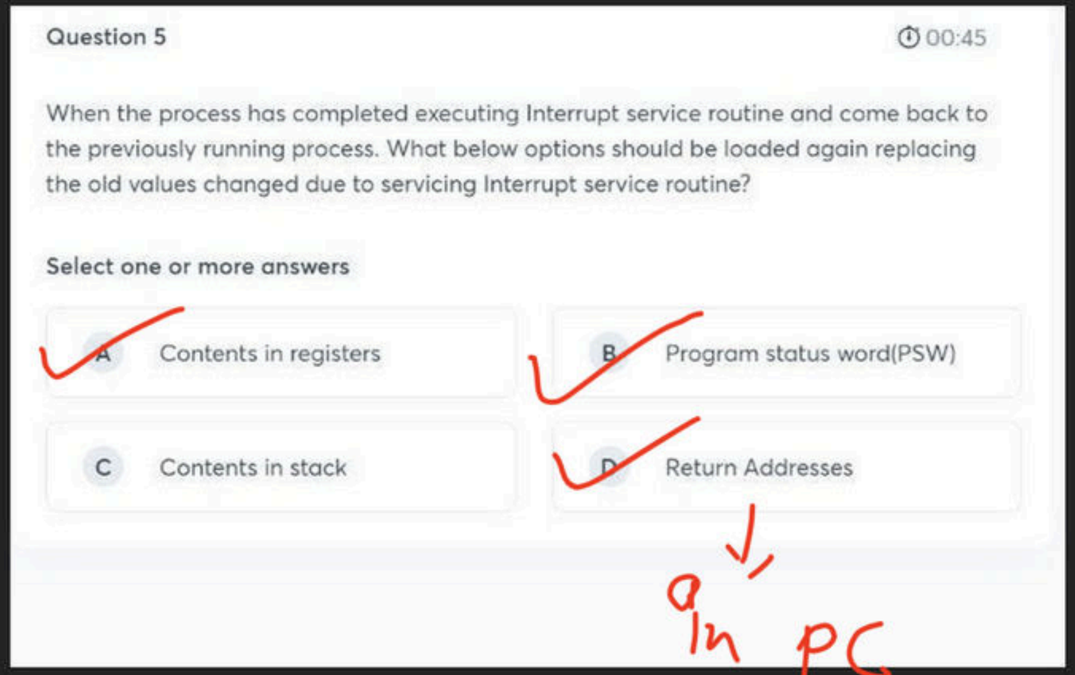
▲ 1 • Asked by Siddhartha

sir , is process id same as process address What is the difference between them?

unacademy

▲ 1 • Asked by Jai

Please help me with this doubt



unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt

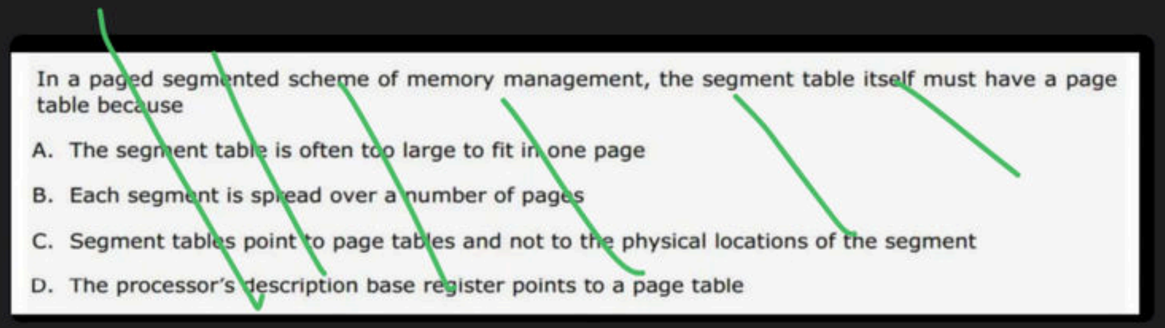
A multi-user, multi-processing operating system cannot be implemented on hardware that does not support

A. Address translation
B. DMA for disk transfer
C. At least two modes of CPU execution (privileged and non-privileged)
D. Demand paging

unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt



In a paged segmented scheme of memory management, the segment table itself must have a page table because

A. The segment table is often too large to fit in one page

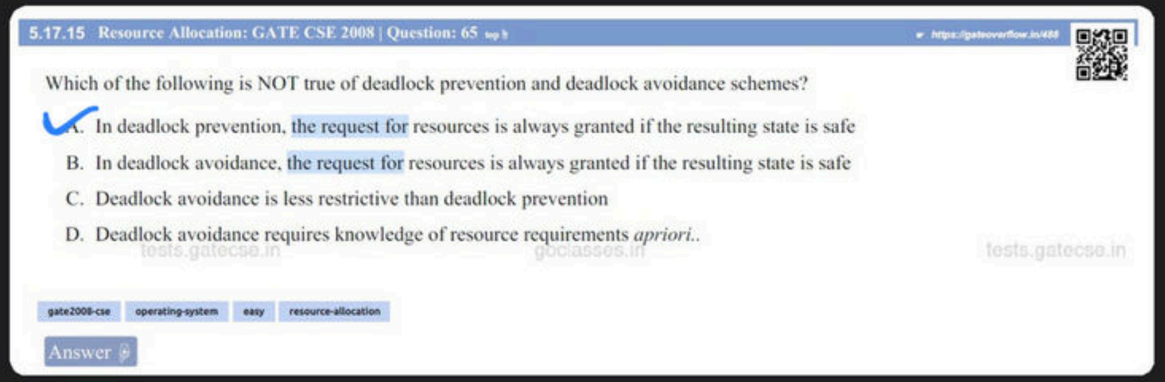
B. Each segment is spread over a number of pages

C. Segment tables point to page tables and not to the physical locations of the segment

D. The processor's description base register points to a page table

▲ 1 • Asked by Soham

Please help me with this doubt



5.17.15 Resource Allocation: GATE CSE 2008 | Question: 65 top ▸ https://gateoverflow.in/488

Which of the following is NOT true of deadlock prevention and deadlock avoidance schemes?

A. In deadlock prevention, the request for resources is always granted if the resulting state is safe

B. In deadlock avoidance, the request for resources is always granted if the resulting state is safe

C. Deadlock avoidance is less restrictive than deadlock prevention

D. Deadlock avoidance requires knowledge of resource requirements *apriori*..

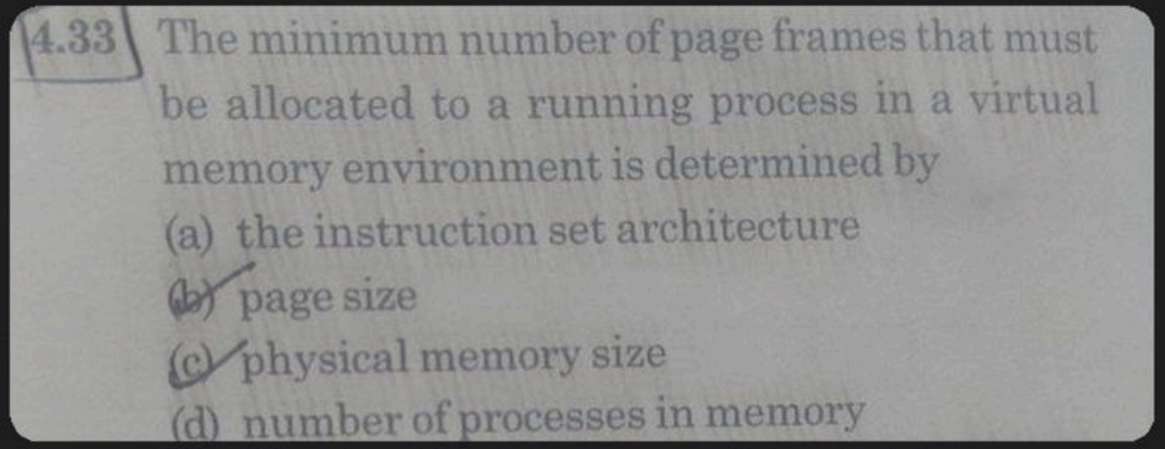
gate2008-cse operating-system easy resource-allocation

Answer

unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt



4.33 The minimum number of page frames that must be allocated to a running process in a virtual memory environment is determined by

(a) the instruction set architecture

(b) page size

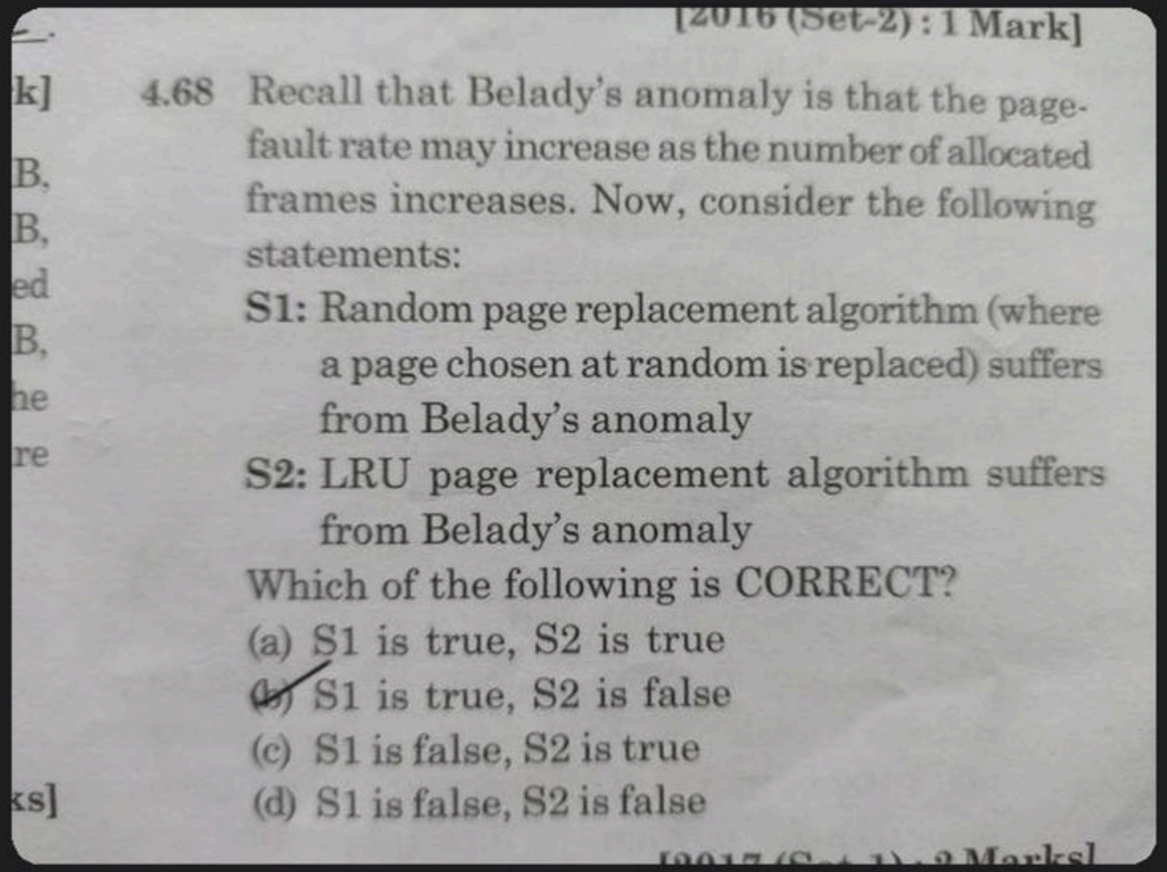
(c) physical memory size

(d) number of processes in memory

unacademy

▲ 1 • Asked by Shreyas

Isme b hona chahiye na



[2016 (Set-2) : 1 Mark]

4.68 Recall that Belady's anomaly is that the page-fault rate may increase as the number of allocated frames increases. Now, consider the following statements:

S1: Random page replacement algorithm (where a page chosen at random is replaced) suffers from Belady's anomaly

S2: LRU page replacement algorithm suffers from Belady's anomaly

Which of the following is CORRECT?

(a) S1 is true, S2 is true

(b) S1 is true, S2 is false

(c) S1 is false, S2 is true

(d) S1 is false, S2 is false

▲ 2 • Asked by Shreyas

what does last line mean?

In a two-level virtual memory, the memory access time for main memory, $t_M = 10^{-8}$ sec, and the memory access time for the secondary memory, $t_D = 10^{-3}$ sec. What must be the hit ratio, H such that the access efficiency is within 80 percent of its maximum value?

no. of page faults = 0

$p = 0 \Rightarrow$ E.M.A. = ideal access time

achieved access time = 80% of ideal access time

unacademy

▲ 1 • Asked by Shreyas

Please help me with this doubt

2.36 Consider the following proposed solution for the critical section problem. There are n processes: $P_0 \ldots P_{n-1}$. In the code, function *pmax* returns an integer not smaller than any of its arguments. For all i, $t[i]$ is initialized to zero.

Code for P_i:

```
do
{
    c[i]=1; t[i] = pmax (t[0],..., t[n–1]) +1; c[i] = 0;
    for every j ≠ i in {0,..., n – 1}
    {   while (c[j]);
        while (t[j] != 0 && t[j] <= t[i]);
    }
    Critical Section; t[i] = 0;
    Remainder Section;
} while (true);
```

Deadlock possible

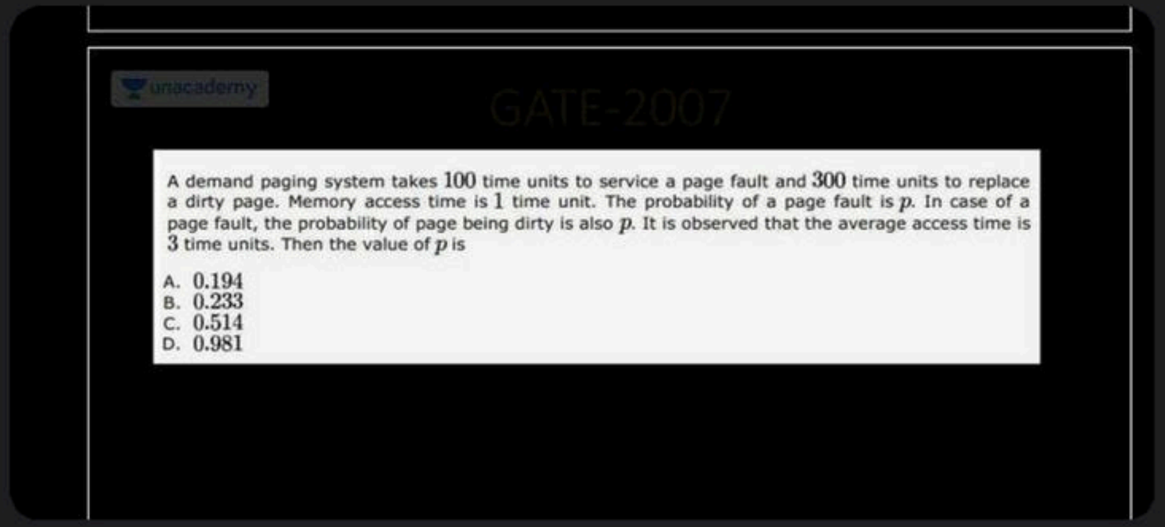
Which one of the following is TRUE about the above solution?

(a) At most one process can be in the critical section at any time
(b) The bounded wait condition is satisfied
(c) The progress condition is satisfied
(d) It cannot cause a deadlock

[2016 (Set-1) : 2 Marks]

▲ 1 • Asked by Riya

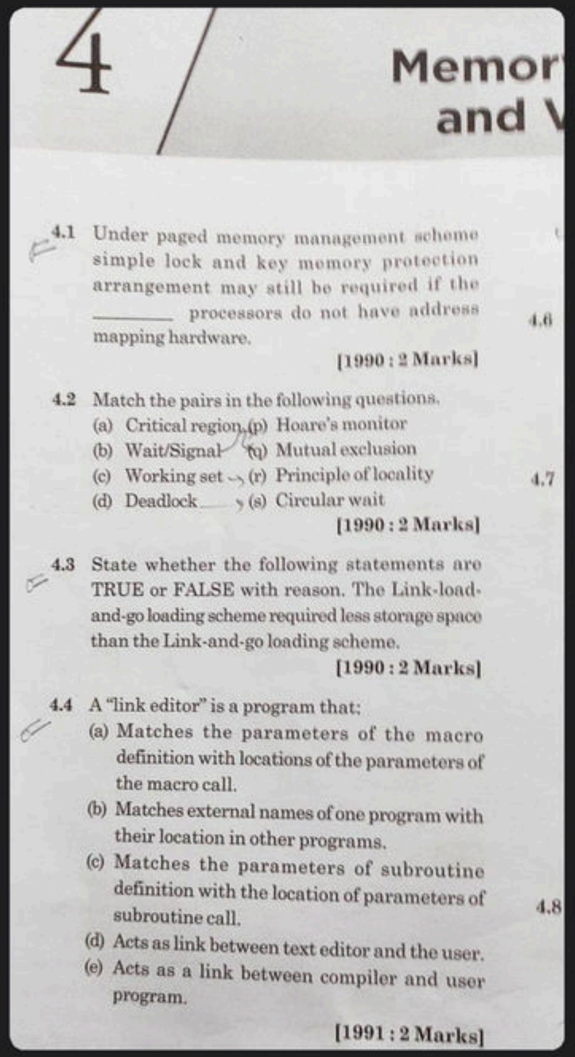
Please help me with this doubt



unacademy

▲ 1 • Asked by Soham

ye sab out of syllubs h sir??



4 Memor and V

4.1 Under paged memory management scheme simple lock and key memory protection arrangement may still be required if the _____ processors do not have address mapping hardware. 4.6

[1990 : 2 Marks]

4.2 Match the pairs in the following questions.

(a) Critical region (p) Hoare's monitor
(b) Wait/Signal (q) Mutual exclusion
(c) Working set (r) Principle of locality 4.7
(d) Deadlock (s) Circular wait

[1990 : 2 Marks]

4.3 State whether the following statements are TRUE or FALSE with reason. The Link-load-and-go loading scheme required less storage space than the Link-and-go loading scheme.

[1990 : 2 Marks]

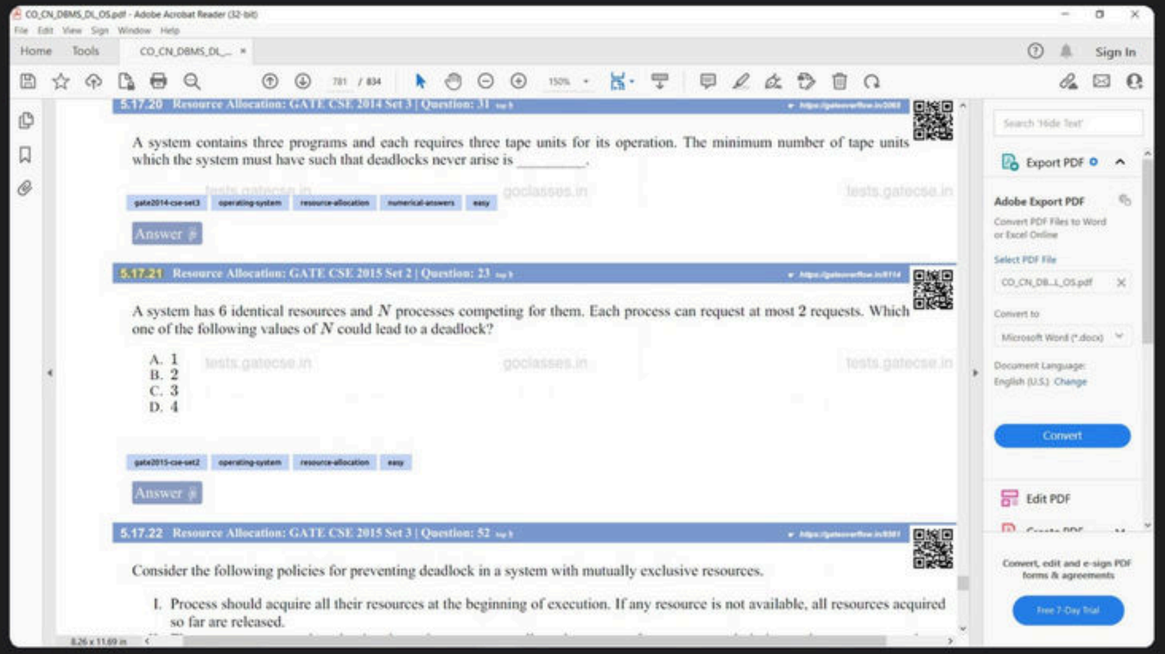
4.4 A "link editor" is a program that:

(a) Matches the parameters of the macro definition with locations of the parameters of the macro call.
(b) Matches external names of one program with their location in other programs.
(c) Matches the parameters of subroutine definition with the location of parameters of subroutine call. 4.8
(d) Acts as link between text editor and the user.
(e) Acts as a link between compiler and user program.

[1991 : 2 Marks]

▲ 1 • Asked by Soham

ye wala



unacademy

Quiz - 5

Ques - 1)

$256\,kB \Rightarrow 256\,k \times 8\ bits = 2^{16} * P.T.G.$

$P.T.E. = \dfrac{256\,k \times 8}{2^{16}}$

$= 32\text{-bits}$

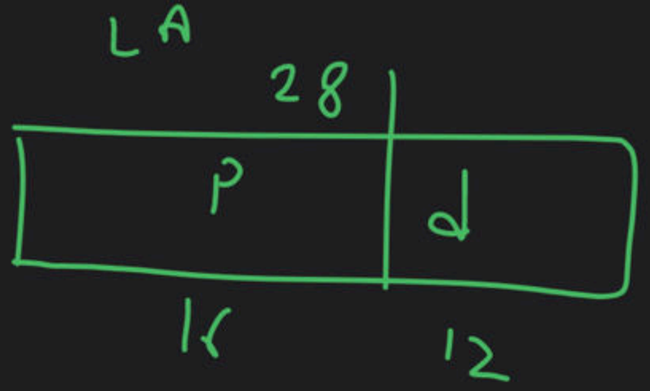
L.A.S. = 256 MB

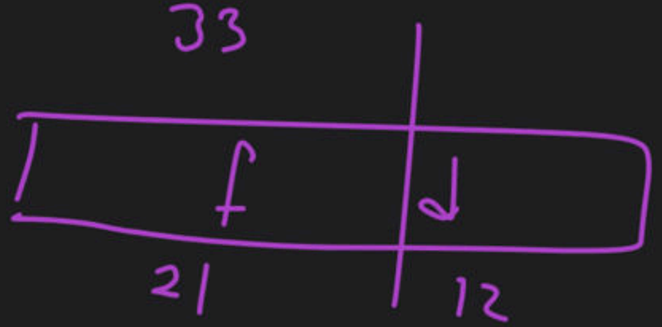
$\Downarrow$

2^{28} B

$\Downarrow$

L.A. = 28 bits





PTE = 21 + extra

$\Downarrow$

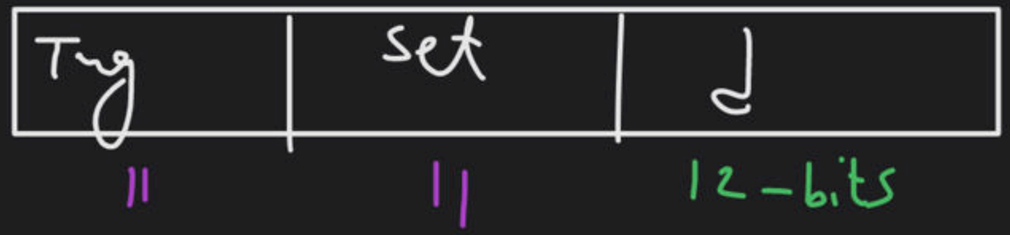
11-bits

Ques-2) d)

Ques-3) a, b, c, d

Ques-4)

LA.



LA = 11 + 11 + 12

= 34-bits ⇒ L.A.S.

$= 2^{34}$ bytes

= 16 GB.

⇒ Ans.

page = 4KB ⇒ d = 12 bit

no. of sets in TLB $= \frac{4096}{2} = 2048 = 2^{11}$ ⇒

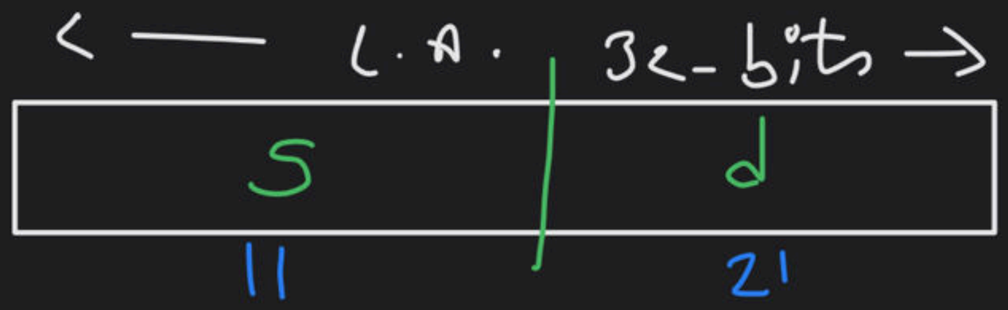
set = 11-bits

EMAT = 15 + 300 + 0.1 * 300

= 345 nsec

6) decrease page size

7)



d ⇒ based on largest seg. size

$d = \log_2(2M) = 21$

seg. sizes: 16KB to 2Mbytes

no. of segments $= 2^{11} = 2 * 2^{10}$

Question

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)

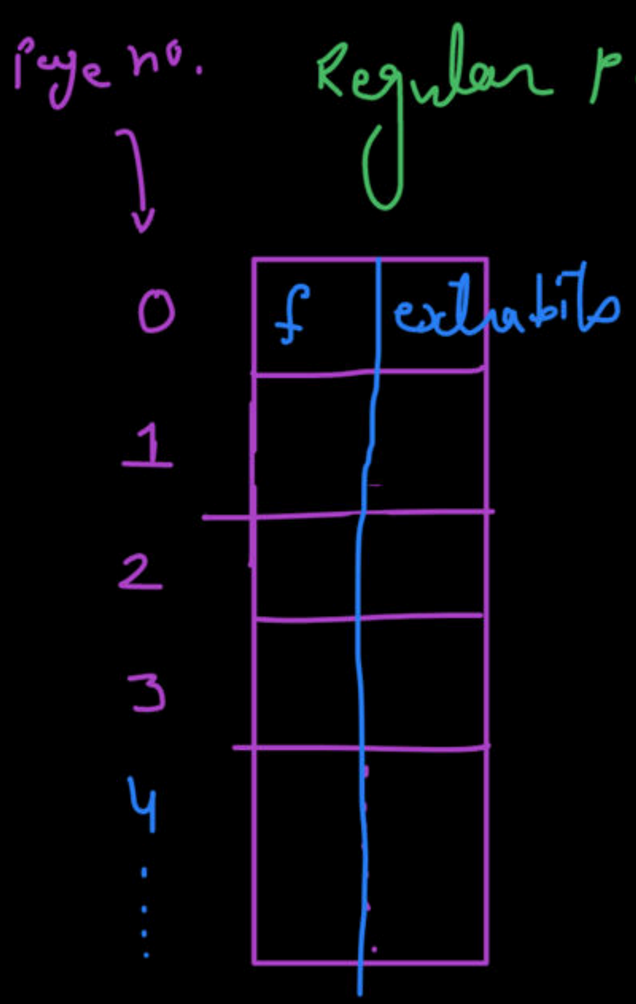
Problem in Virtual Memory

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

wastage of space to store large P. T.

because all pages are not in mm.

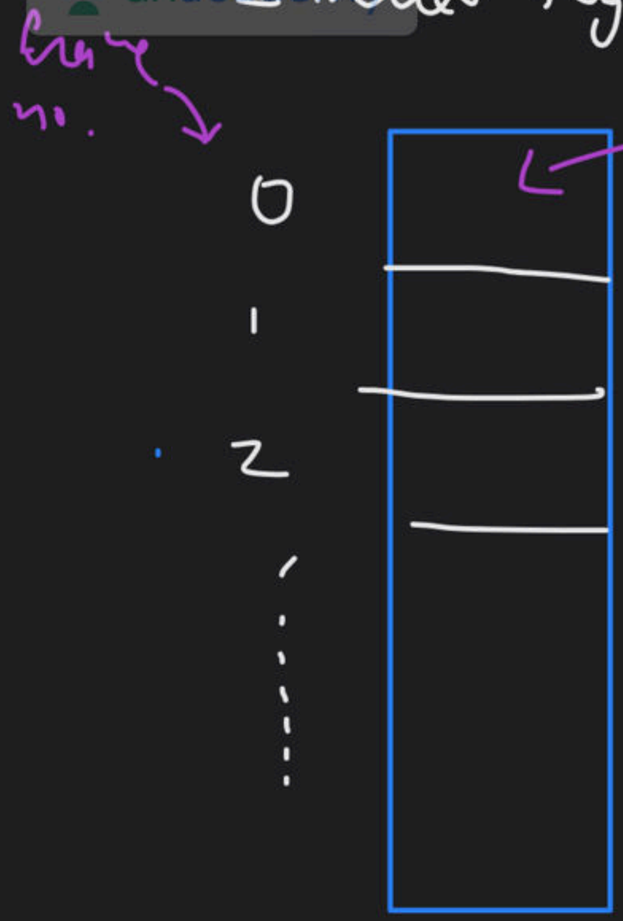
Inverted Page Table



P.T. will have total no. of entries for all processes included = no. of frames in mm.

So that there will not be any invalid entry (exstg) for any process in P.T.

Inverted Page-table

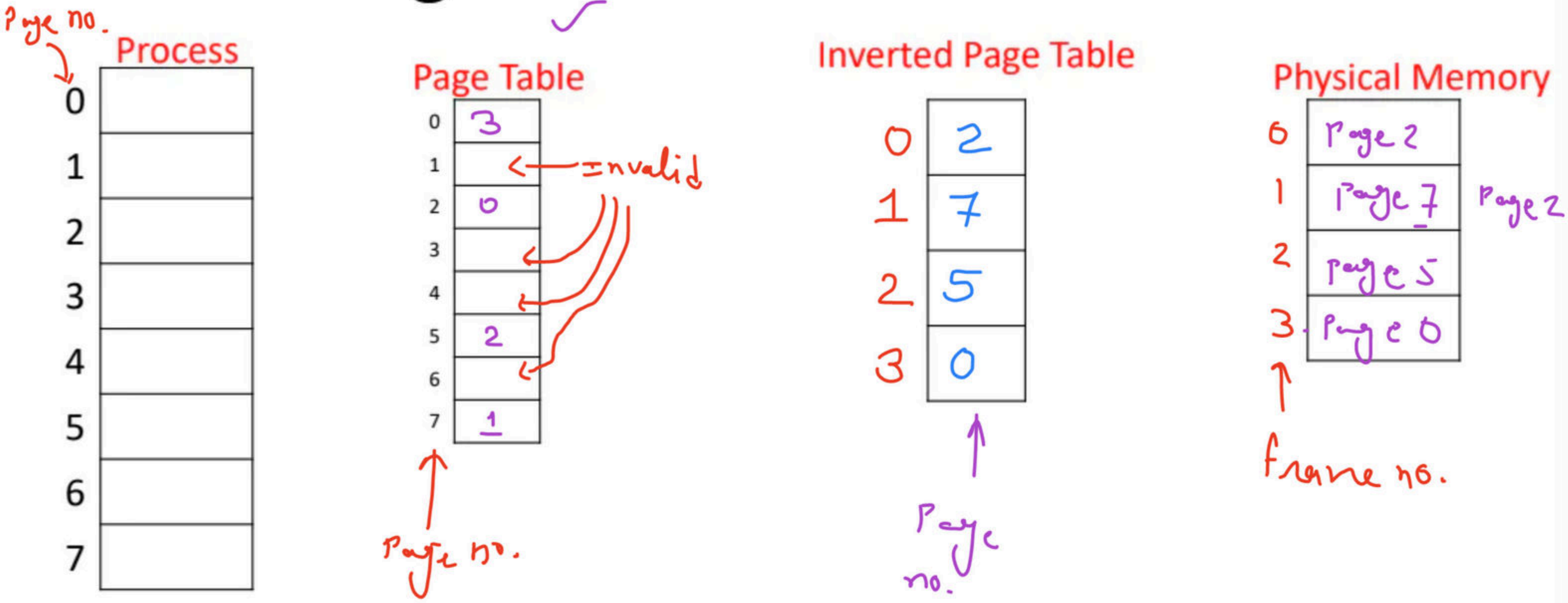


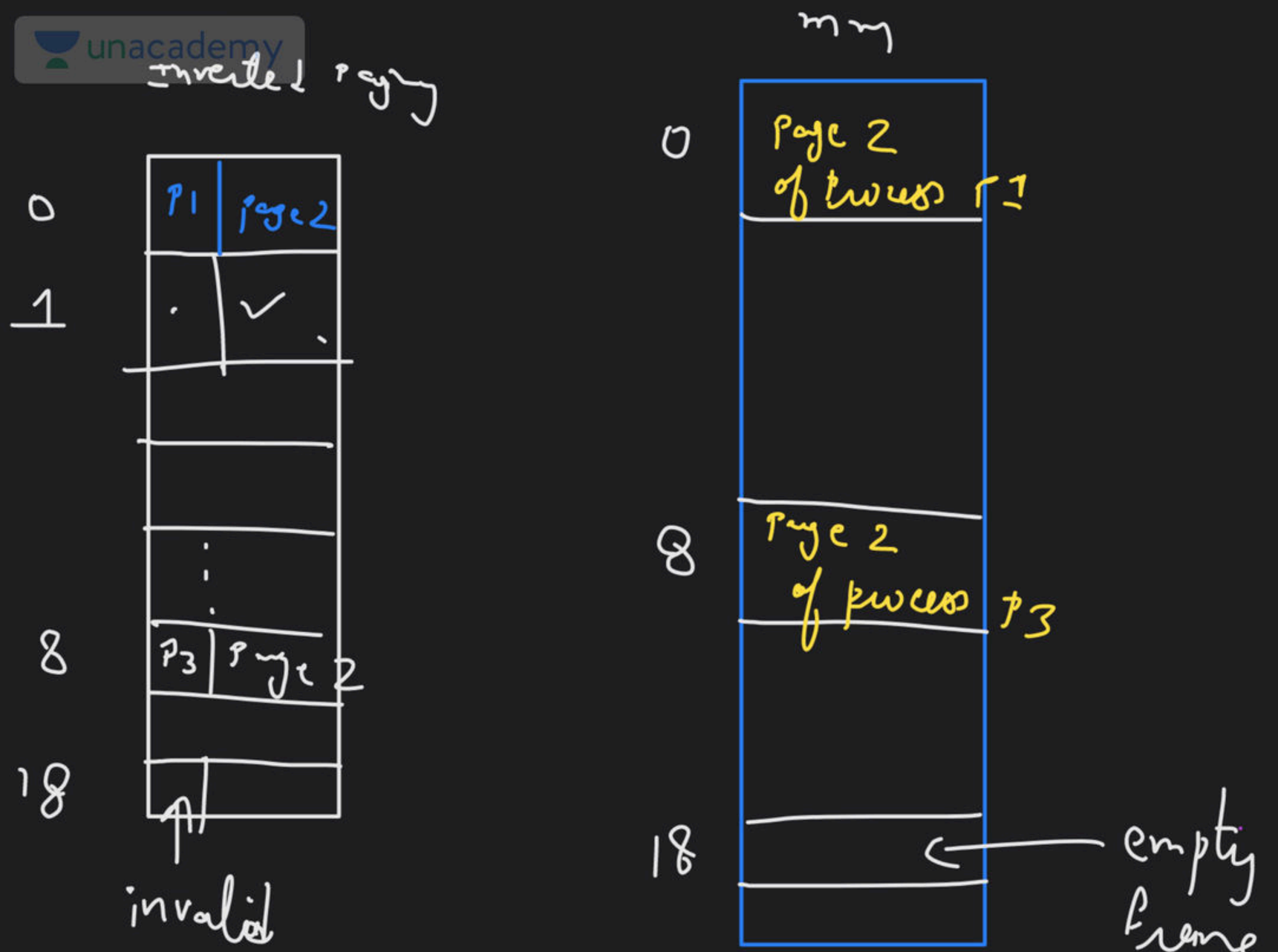
Page no. which is present on frame

no. of entries in inverted P.T. = no. of frames in mm.

⇒ single P.T. for all processes in maintained.

Inverted Page Table



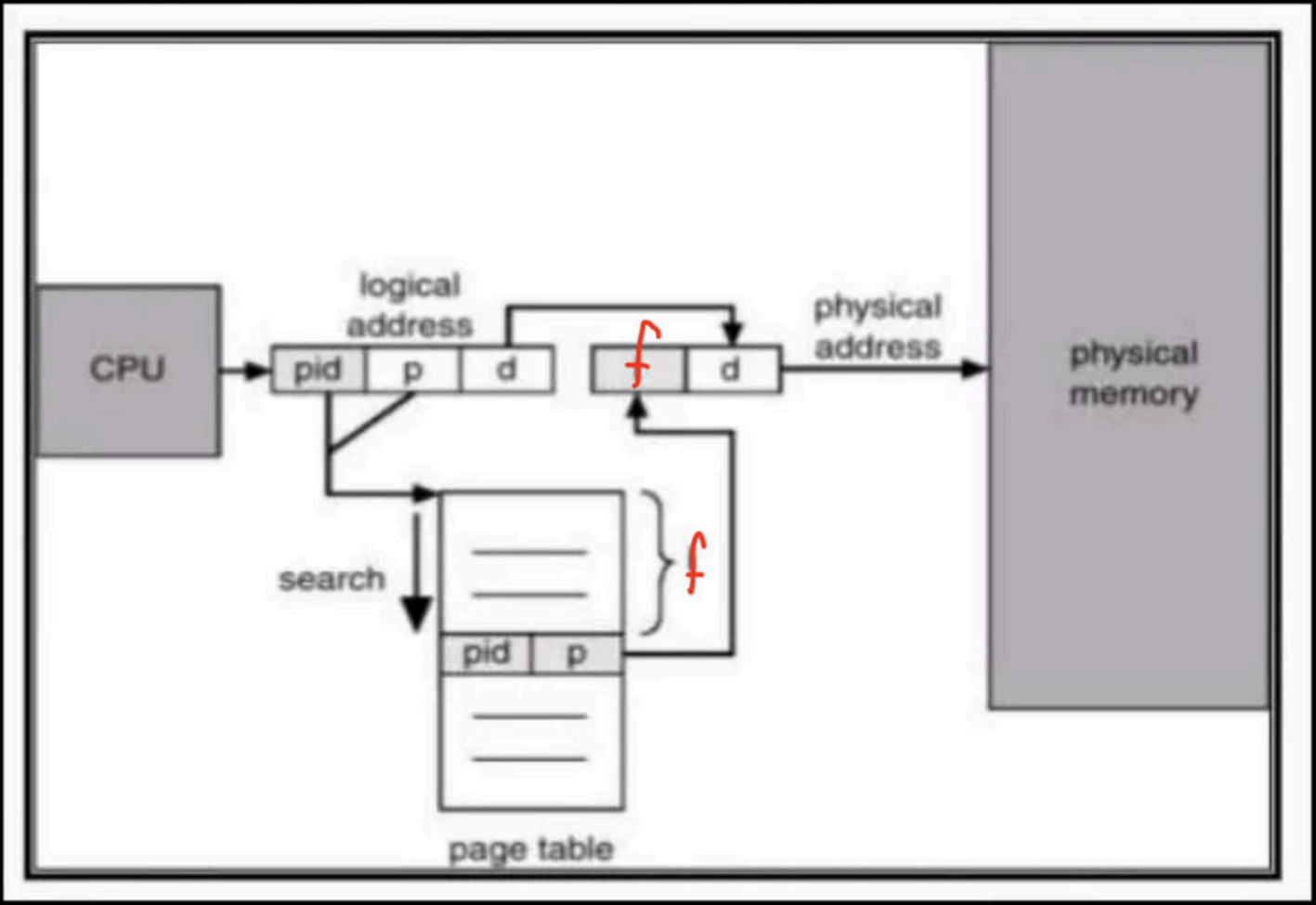

Inverted page
0
1
8
18
P1
page 2
P3
page 2
invalid
mm
0
Page 2 of process P1
8
Page 2 of process P3
18
empty frame

Inverted Page Table

Each entry in the page table contains the following fields:

1. Page number
2. Process id
3. Control bits
4. [illegible] pointer

Inverted Page Table



Inverted Page Table

Advantages & Disadvantages:

1. Reduced memory space
2. Longer lookup time (searching-time) $\Rightarrow$ O(no. of frames in mm)
3. Difficult shared memory implementation

Ques L.A. = 46 bits

P.A. = 32 bit

Page size = 2 kbytes

Process-id = 8 bits

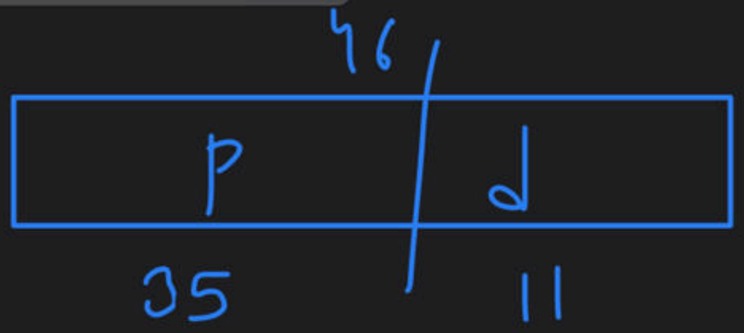
Page table size = ?

Inverted page table size = ?

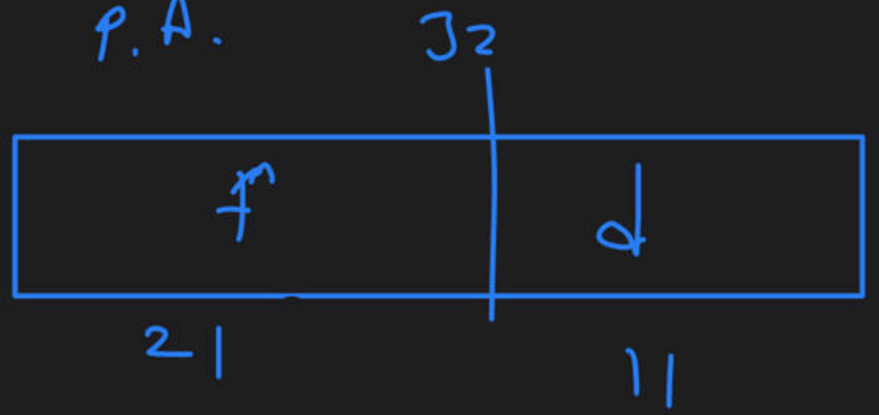
Normal P.T. entry :- frame + 4 extra bit

Inverted P.T. entry => pid + Page no. + 2 extra bit

L.A.



P.A.



normal P.T. size $= 2^{35} * (21 + 4)$ - bits $= 2^{35} * 25$ bits $= 800$ Gbit

$= 100$ Gbytes

Inverted P.T. size $= 2^{21} * (8 + 35 + 2)$ bits

$= 2^{21} * 45$ bits

$= 90$ Mbits

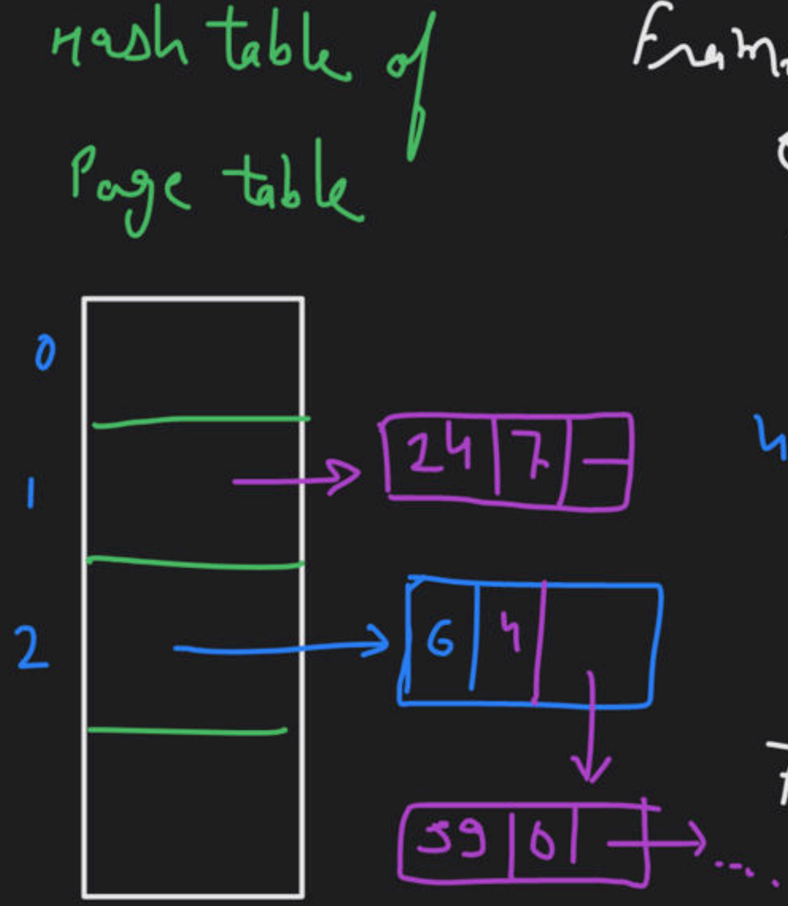
Hashed Page Table

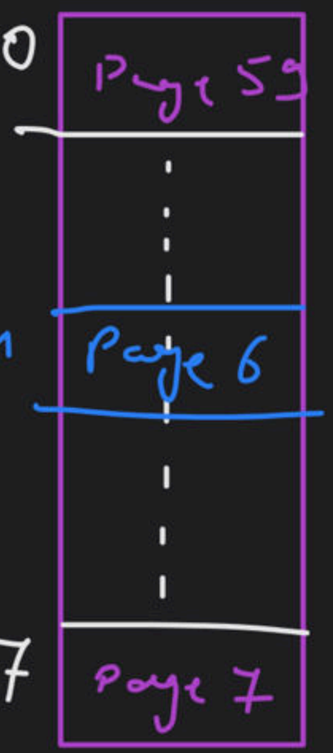
In this virtual page, the number is hashed into a page table

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

location where P.T. entry is stored = H (Page no.)

example:





assume

2 = H (Page no. = 6)

1 = H (Page no = 24).

2 = H (Page no. = 59)

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.

File

A file is a named collection of related information that is recorded on secondary storage.

File Attributes

1. Name
2. Extension
3. Size
4. Date
5. Author
6. Created, Modified, Accessed
7. Attributes: Read-only, hidden
8. Default Program
9. Security Details

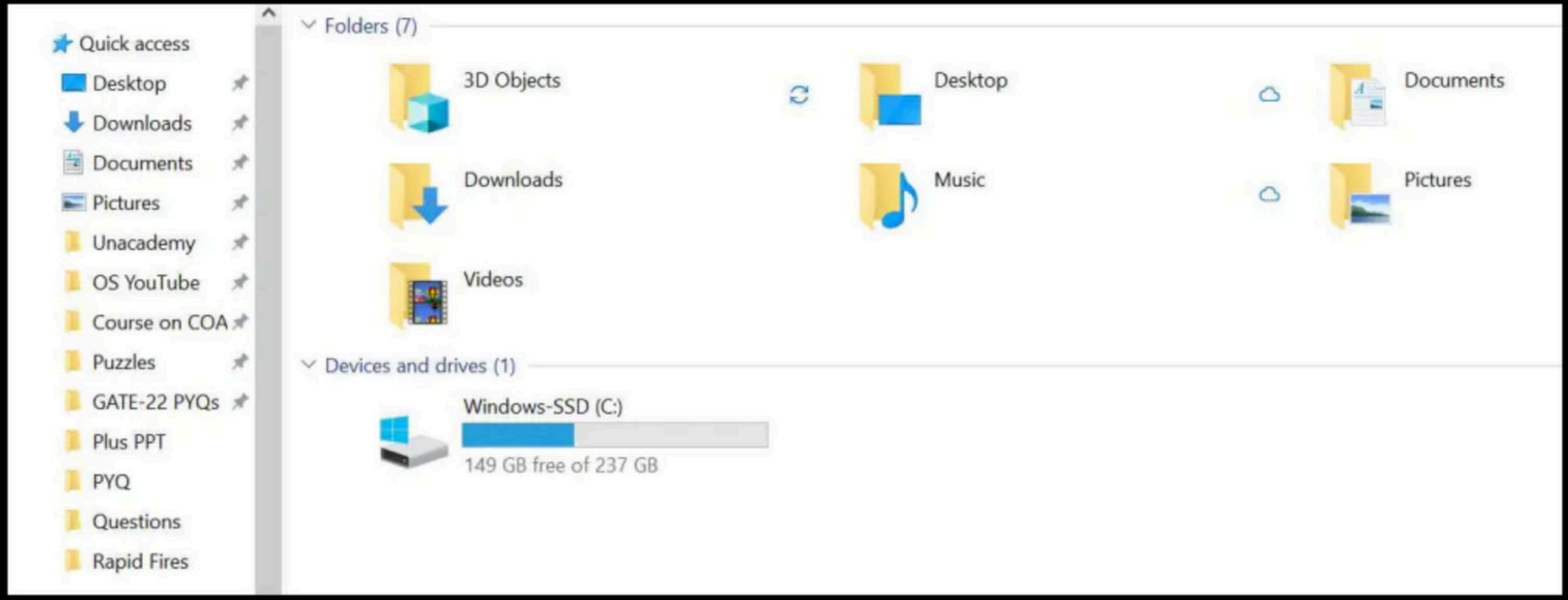
File Directory

Collection of files

File System

Module of OS which manages, controls and organizes files and related structures

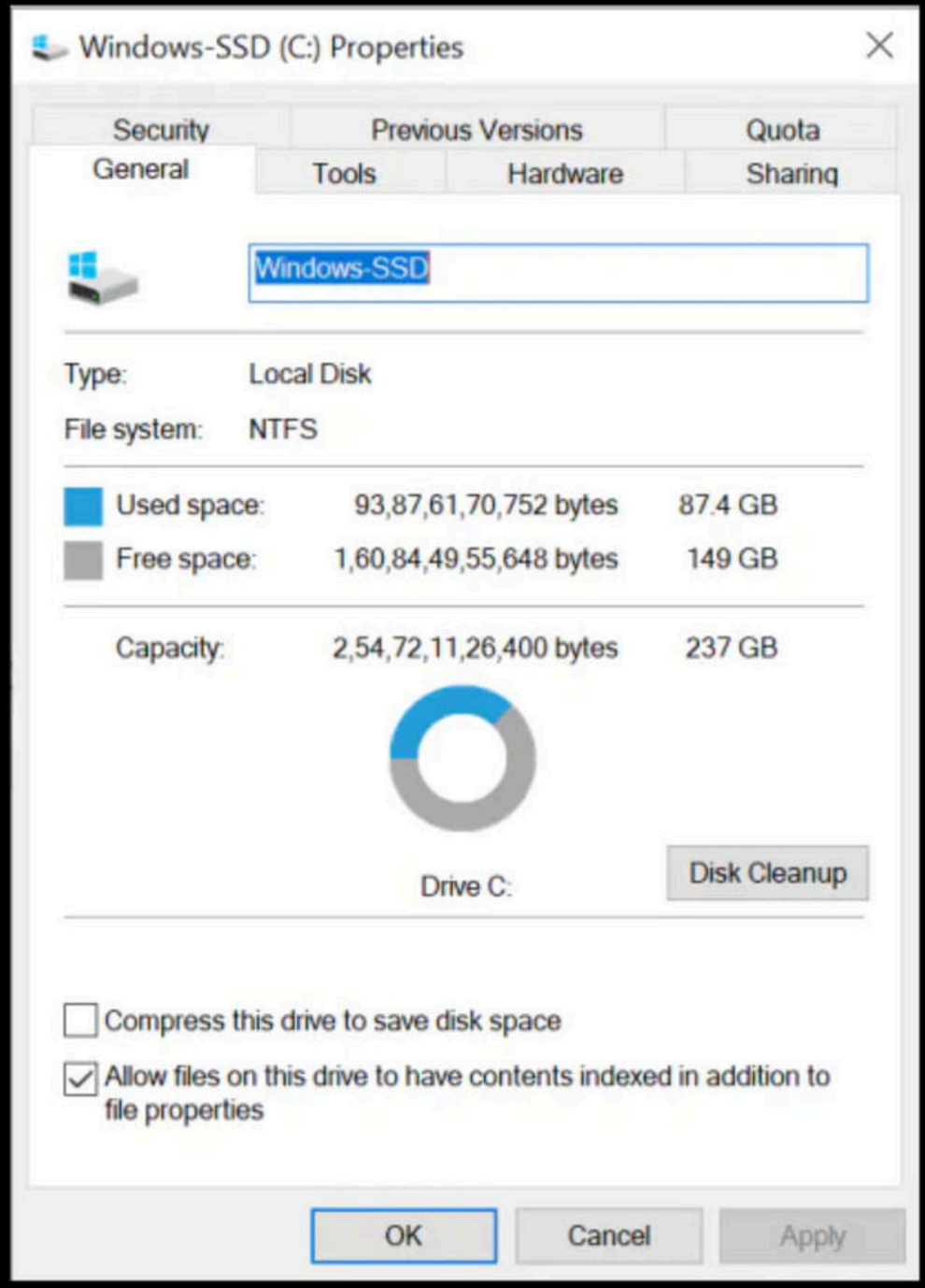
File System


Quick access
Desktop
Downloads
Documents
Pictures
Unacademy
OS YouTube
Course on COA
Puzzles
GATE-22 PYQs
Plus PPT
PYQ
Questions
Rapid Fires
Folders (7)
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Devices and drives (1)
Windows-SSD (C:)
149 GB free of 237 GB

Types of File Systems

1. FAT32
2. NTFS
3. HFS+
4. Ext2 / Ext3 / Ext4
5. Swap

File System


Windows-SSD (C:) Properties
Security
Previous Versions
Quota
General
Tools
Hardware
Sharing
Windows-SSD
Type: Local Disk
File system: NTFS
Used space: 93,87,61,70,752 bytes 87.4 GB
Free space: 1,60,84,49,55,648 bytes 149 GB
Capacity: 2,54,72,11,26,400 bytes 237 GB
Drive C:
Disk Cleanup
Compress this drive to save disk space
Allow files on this drive to have contents indexed in addition to file properties
OK
Cancel
Apply

File Directory Structure

1. Single-Level Directory

File Directory Structure

2. Two-Level Directory

File Directory Structure

3. Tree Structure Directory

unacademy

Happy Learning.!