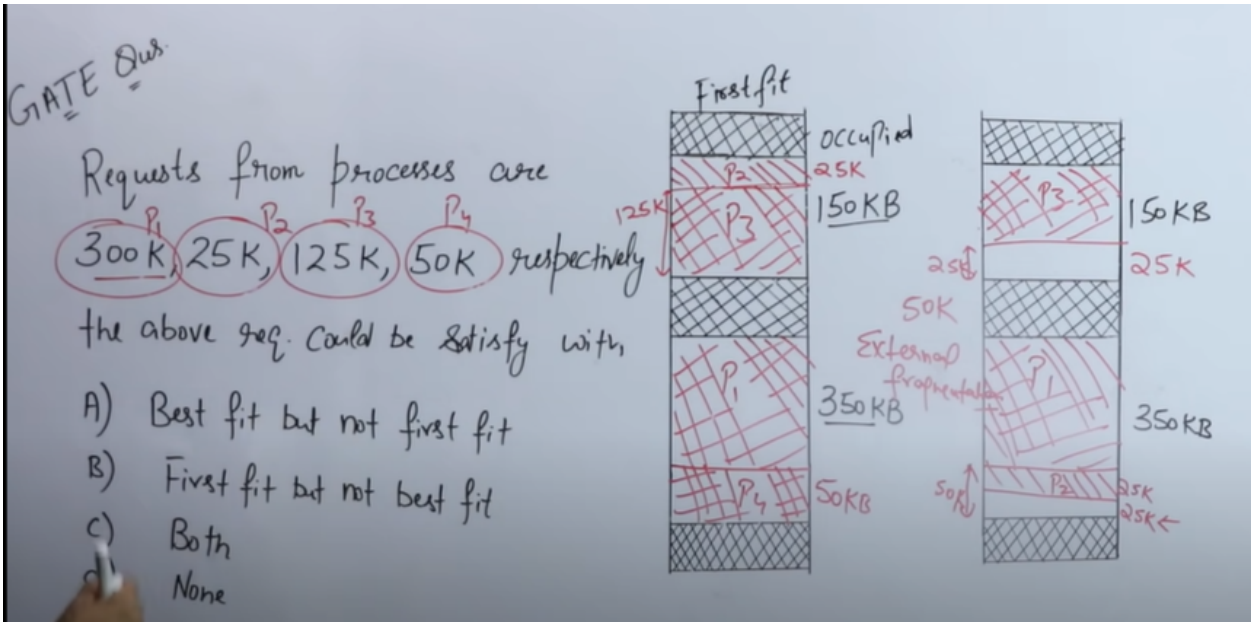
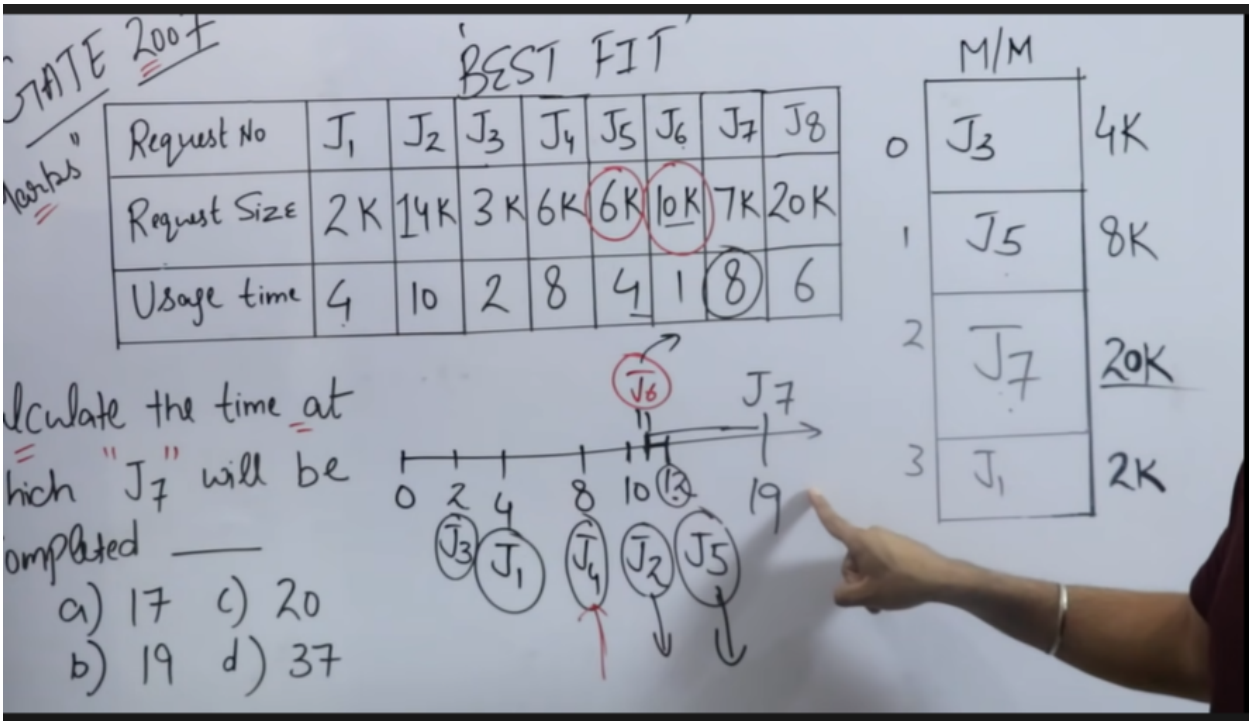
GATE Question Solved on First Fit,Best Fit and Worst fit Memory Allocation



Q2:



Non Contiguos Memory Allocation:

Need of Paging | Memory Management | Operating System

to avoid the problem of external fragmentation this concept is introduced which saved alot of memory wastage:

Time consuming process - to avoid this process is devided into the pages outside the memory and then it is brought into the memory and pages are devvided into the smaller frames:

So here the processes are devided into the pages and the main memory is devided into the frames.

```
page size = frame size
```

What is Paging | Memory management

Multiple page tables:

Mapping:Mapping is done by Memory Management Unit, in this case when CPU ask for the any bit of the process then it should check in the memory and get that bit but i memory it wopulkd be allocated at diff address so MMU converts the logicl address inot the physical address with the help of PAGE TABLE.

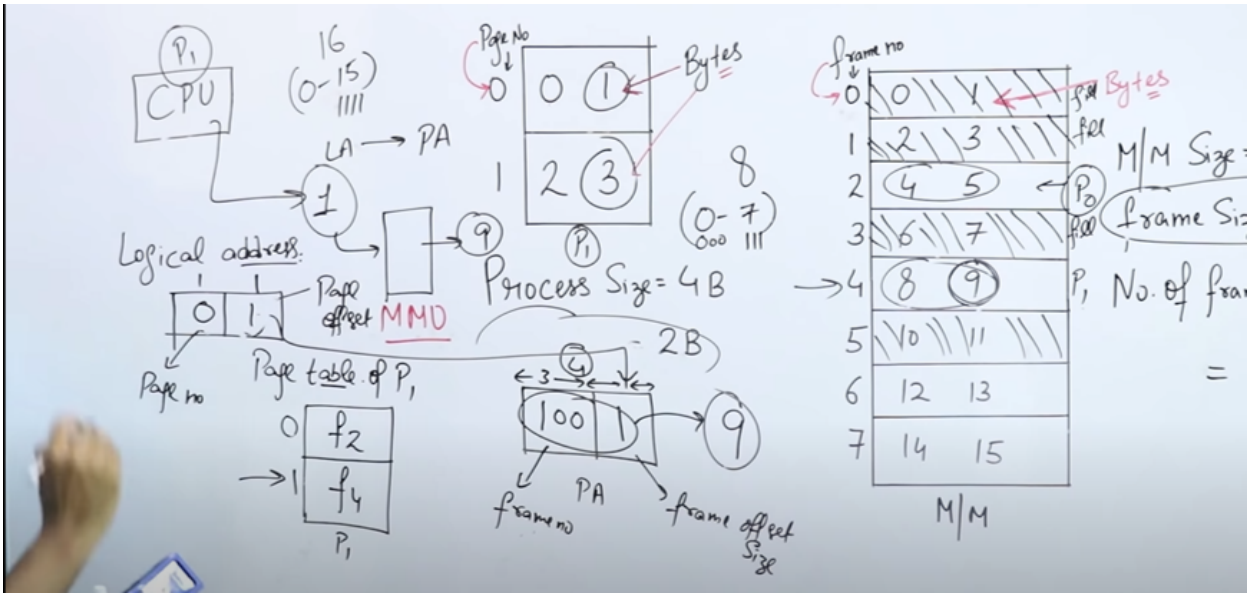
CPU always works on logical address:

Logical Address : `page no` `page offset` // (Virtual address space)

pafe offset also called as page size and convert that into the binary!

Physical address :

Every process will have won page table and page table contains the frame numbers



Lets' say : Page P1 size is 4 B and page size is 2 B so the No of pages for P1 is 2
Main memory size is 16B and frame size is 2B so the no of frames are 8.

for every process page table would be different as pages for particular process wouild be in different frames.
lets say for P1,pages are in frame 2 and frame 4
so page table is ;

0 f2
1 f4

As CPU can access only main memory so when CPU askf for the 3rd byte of the process P1 its' not there at 3rd byte in main memory due to non contiguous allocation, so here logical address of P1 i.e. 3rd byte will converted into the physical i.e. actual address of that byte in the main memory which here in this e.g. is at 9th byte in MM

So the logical address is: `1 1` // here value is 3rd byte of Main Memory
(Logical Address : `page no` `page offset`)
here for 3rd byte the page no is 1st for P1 nad page offset is also 1.

here page offset is nothing but the no of bits required to represent the page size so in this e.g. it is 2B so the range is 0-1 as in binary no starts from 0 and to represent the 1 we need only 1 bit.
if page size would have been 8B then range would be 0-7 and to represent the 7 we need 3 bits i.e.(000-111 i.e. 0-7)
if page size would have been 16B then range would be 0-15 and to represent the 15 we need 4 bits i.e. (0000-1111 i.e. 0-15)

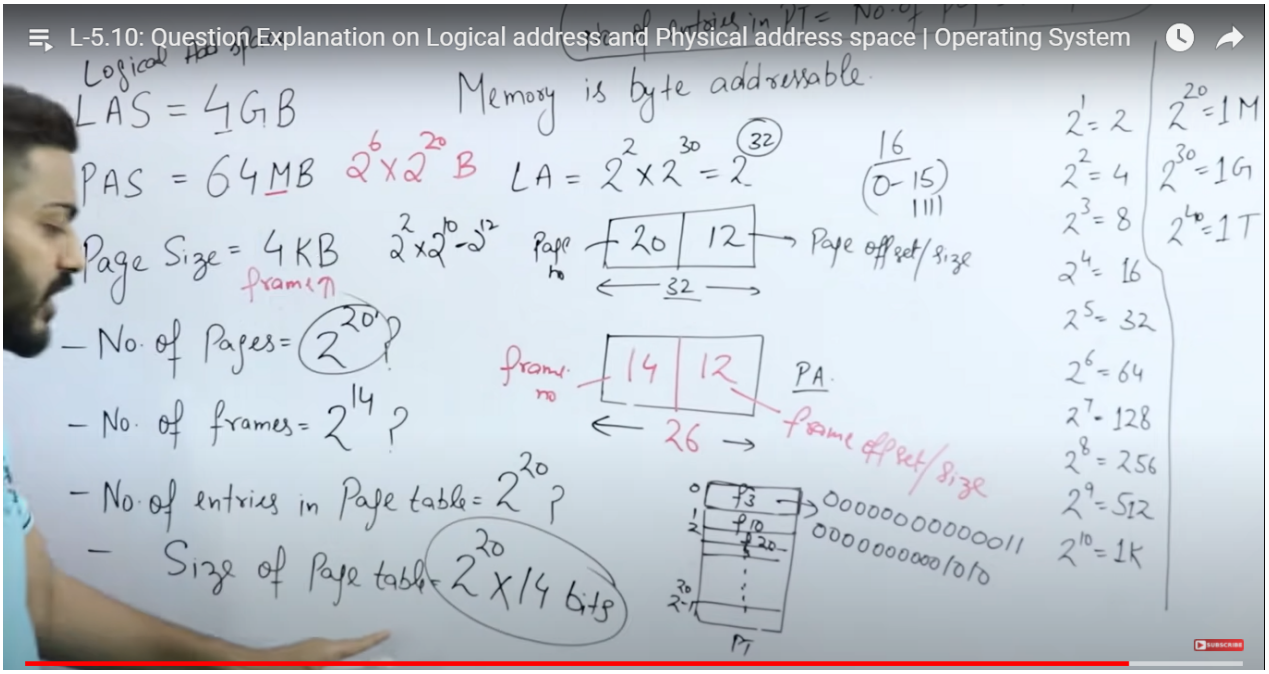
Physical address : It depends on the size of the main memory
In this eg size of MM is 16Bytes so to represent the range 0-15 we must need 4 bits so from this 4 bits also few allocated to the frame no and rest to the frame offset

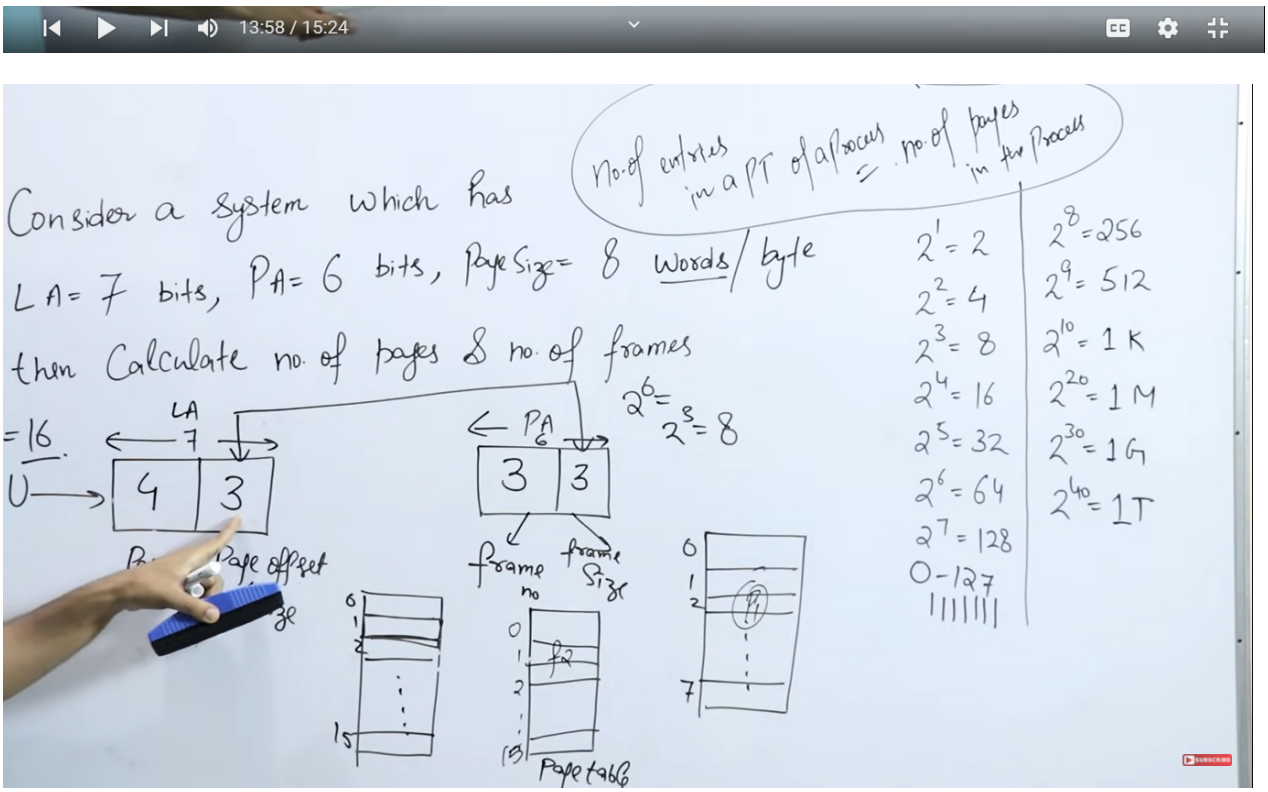
so physical address size is 4 bits and 3 for frame no is 3 bits and frame offset is 1 bite so the PA is 1001

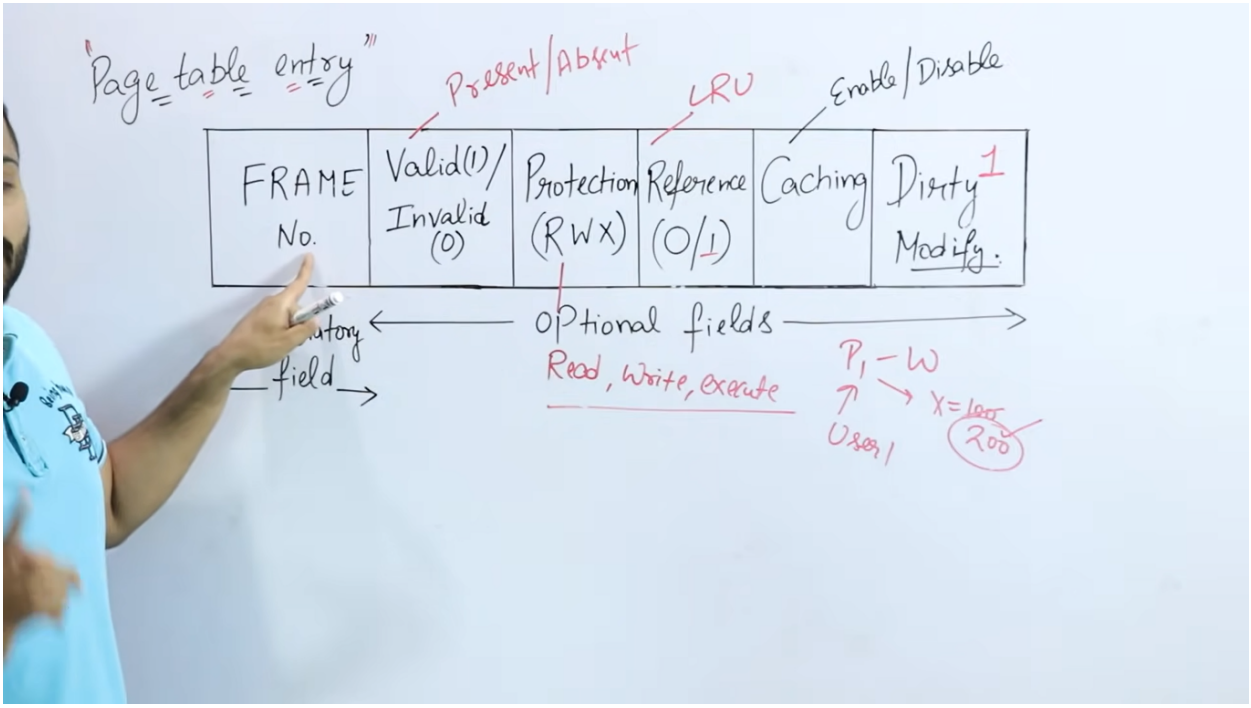
LAS : Logical address space always represents the size of the process
PAS: Physical address space always represents the size of the Main memory

MEMORY IS BYTE ADDRESSABLE

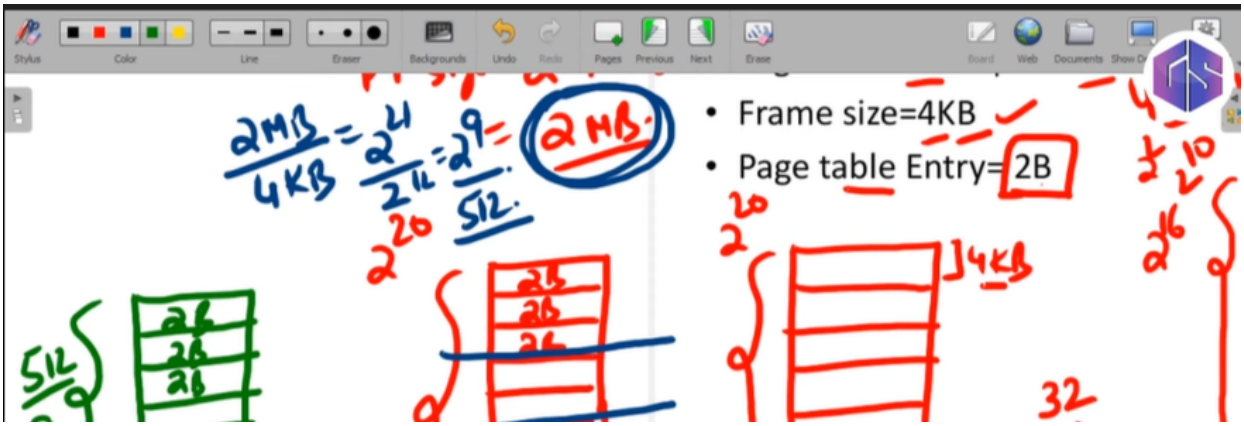
MSB bits for page no and LSB bits for page offsets

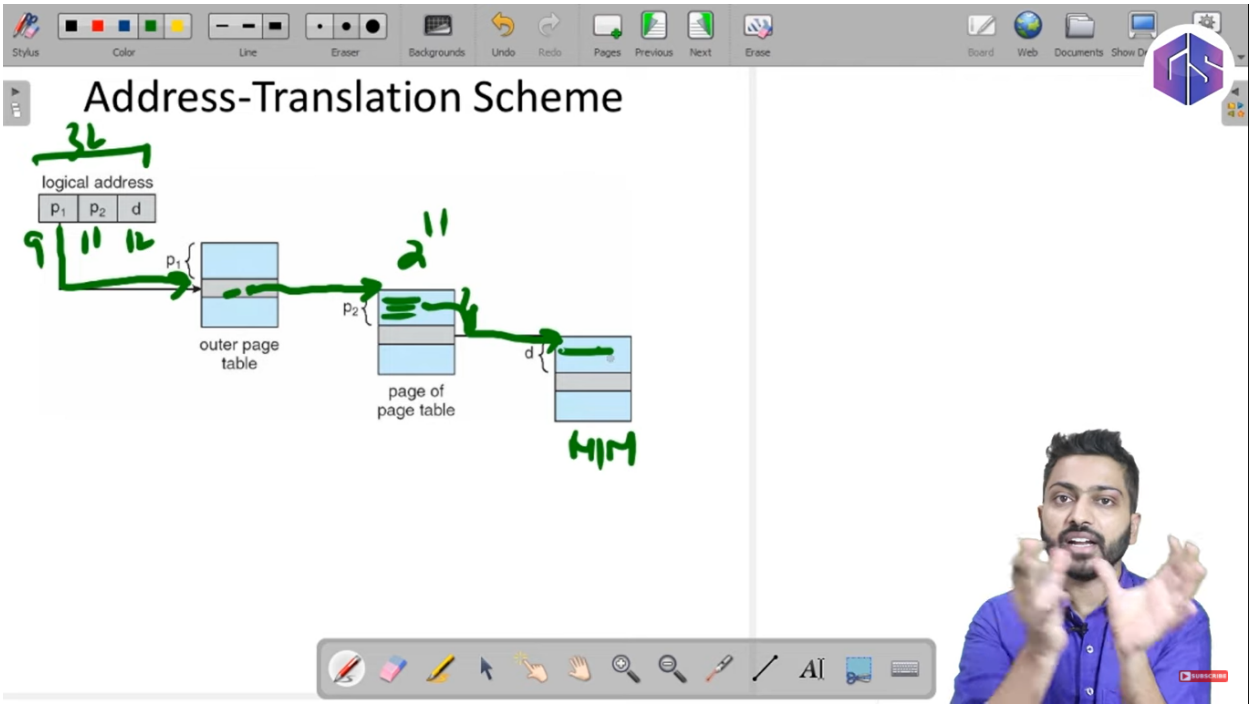






Level Paging in Operating System | Multilevel Paging





Inverted paging | Memory Management | Operating System

Reason to bring this concept:

Basiclaly for every process page table will be created in normal paging and during its execution that page needs to called and stores in the main memory from the secondary memory so if the no of processes increaseses then the no of page will in the main memory also gets increased due to which proper utilisation is not happening for main memory so to avoid this wastage inverted paging concept is introduced.

In this global page table will be created
frame no || page no || process ID

Searching time of the pages for the particular process is the biggest challenege in inverted paging:

Virtual address space

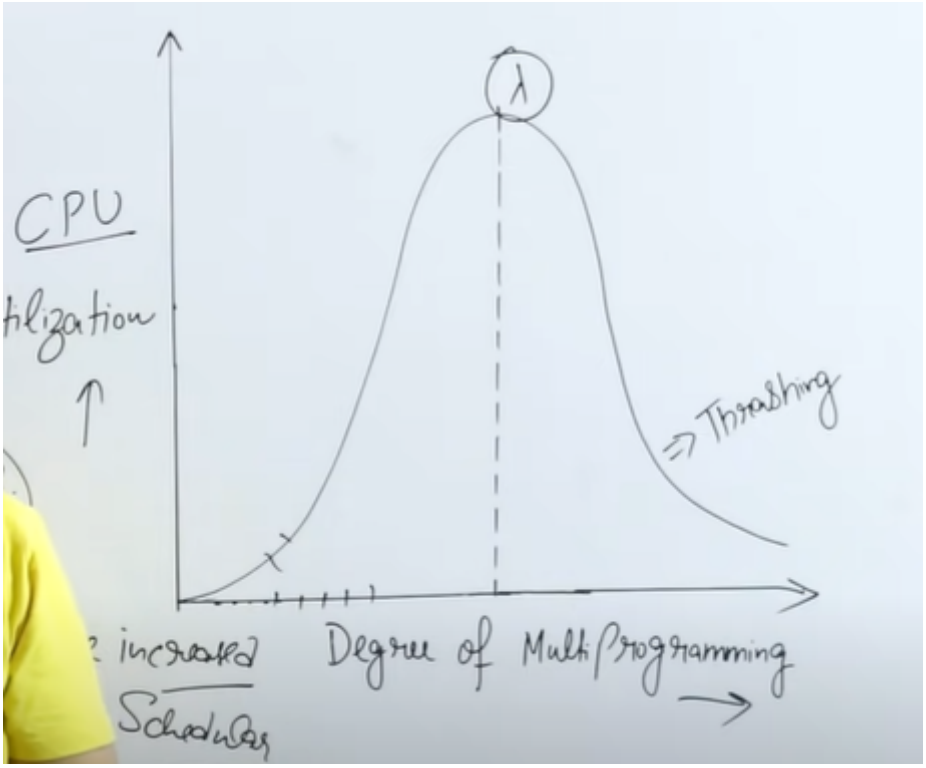
What is Thrashing | Operating System

linked with the DoMP: Degree of Multi Programming

Page fault / PF service time

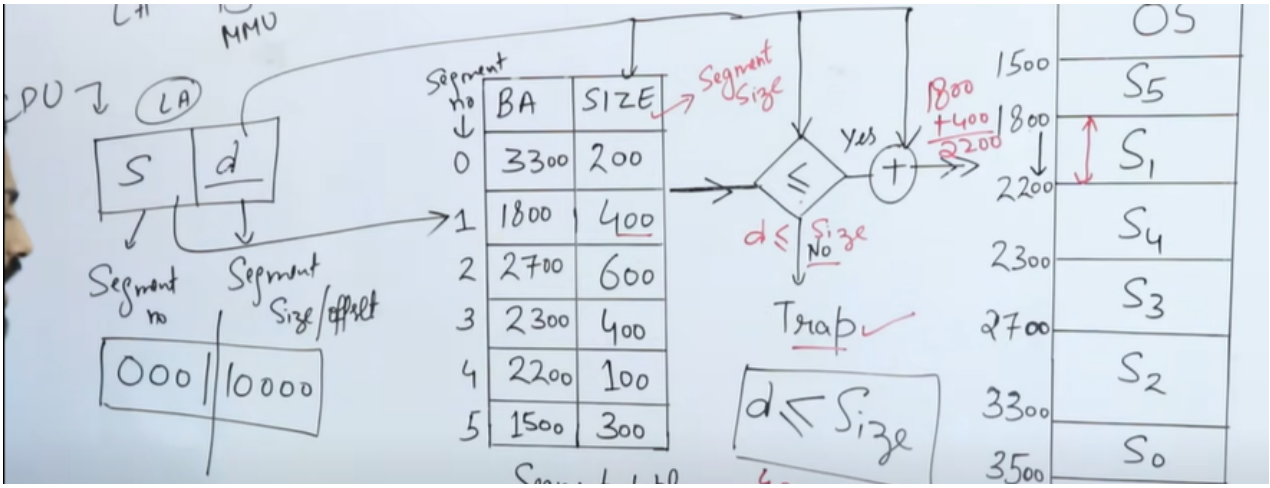
Due to rise in the page fault, system will get busy in bringing the missing pages of the processes from the secondary memory!

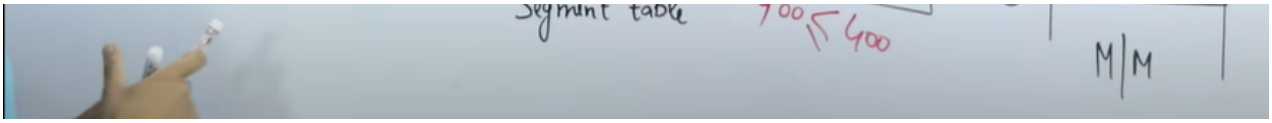
this can be reduced by making LTS bringing the less processes pages into the ready queue!
In belwo fig lamba is the point where LTS has brought first page of all the proceses into the RAM and due to which inspite of having good CPU utilisation after lambda CPU utilisation will be decreased abruptly and that results in the thrashing:



Segmentation Vs Paging | Segmentation Working

Process will be devided into the no of segments of module but in case of paging we devided the process into the pages without knowing whats inside that page but in segmentation we know ehre we have broke the process into the page





Base address:
Limit address:
Size:

SIZE OF EACH SEGMENT IS DIFFERENT:

Overlay:
It's the mthod with which we can bring the process with the size greater than the main memory into the MM.
Widely used in embeded systems:

Virtual Memory | Page fault | Significance of virtual memory | Operating System