

# Operating System

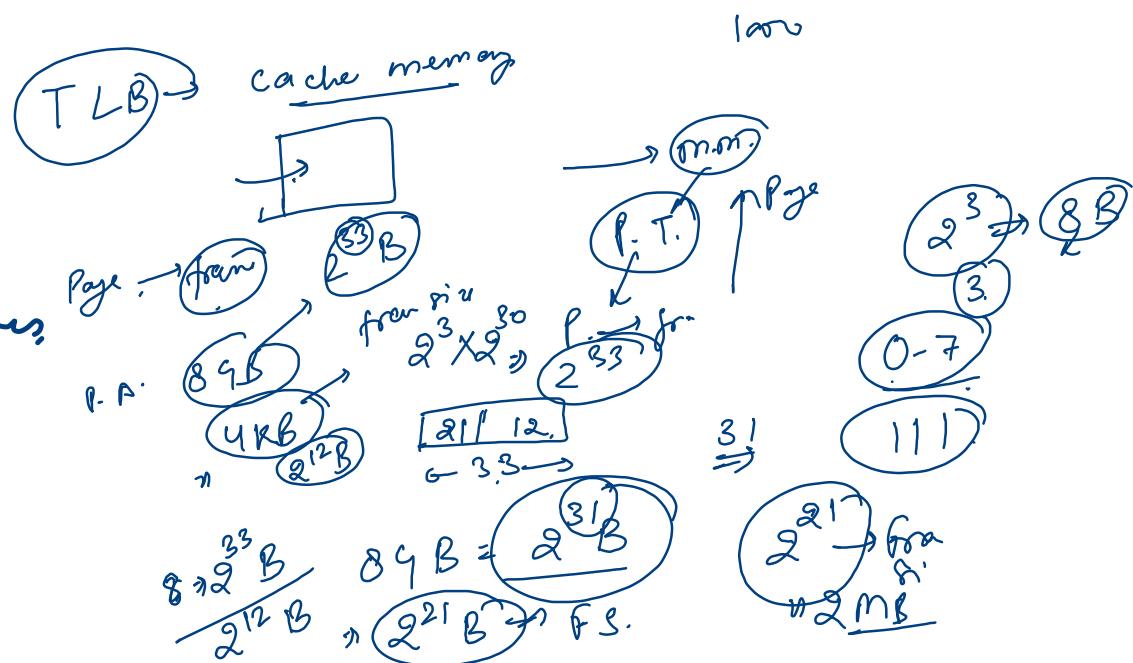
(L-8)

- 1) Inverted paging
- 2) Multilevel paging
- 3) Page fault
- 4) Swapping
- 5) Optimal page replacement algorithm
- 6) FIFO
- 7) LRU
- 8) MRU
- 9) Thrashing

$\text{P.A} \rightarrow \text{frame size} \times \text{No. of frames}$

$$G_B = \gamma_{KB} \times \text{No of frames}$$

$$\frac{8KB}{4KB} = \frac{2^{33}B}{2^{12}B} \Rightarrow 2^{21} \Rightarrow \text{No. of frames}$$



# Operating System

(L-8)

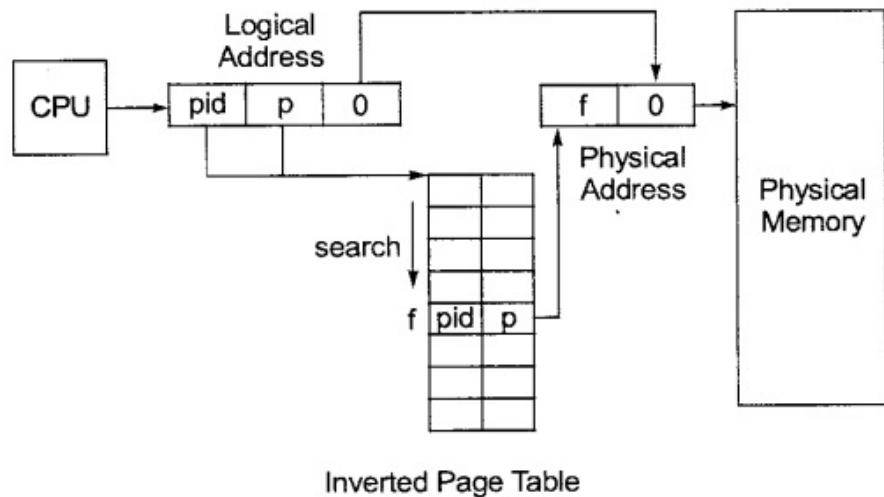
- 1) Inverted paging
- 2) Multilevel paging
- 3) Page fault
- 4) Swapping
- 5) Optimal page replacement algorithm
- 6) FIFO
- 7) LRU
- 8) MRU
- 9) Thrashing



# Inverted Paging

more processes , more page tables

→ overhead of maintaining page table for every process.



- Only one inverted page table  $\Rightarrow$  global page table
- Entries in inverted page table must include process id.
- Maps physical frames to virtual Pages

## Advantages

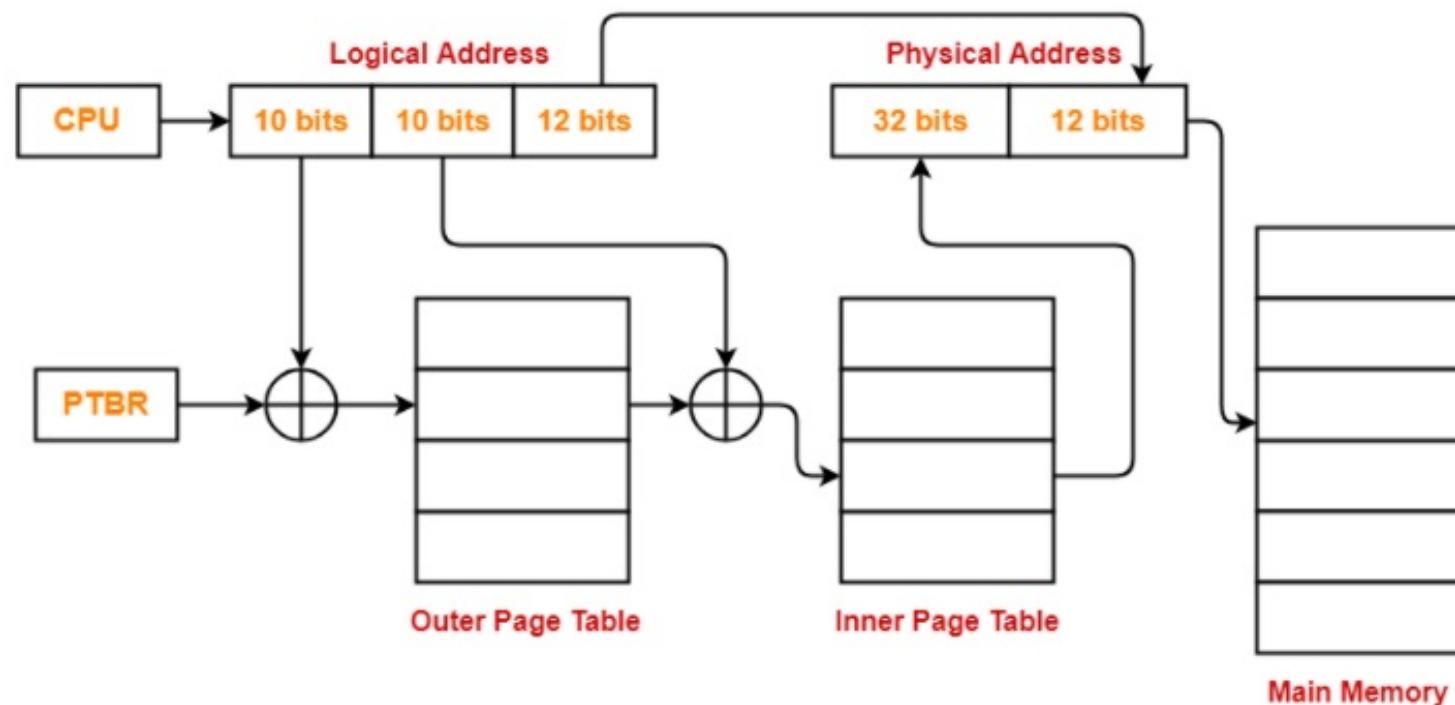
- i) Only one table for many processes

## Disadvantages

- i) Look up time is increased

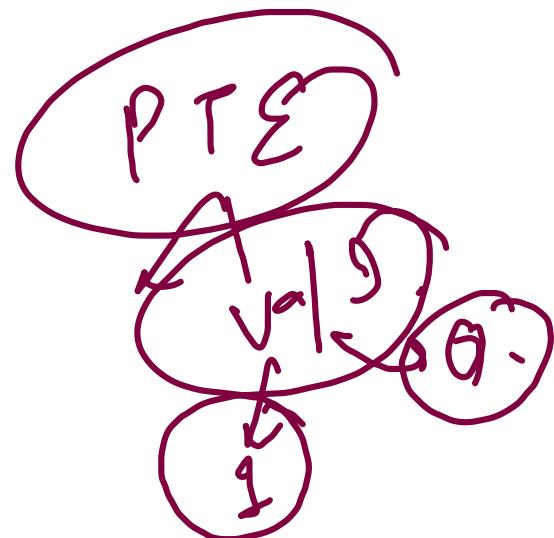
# Multi-level Paging

If size of page table is more than the frame size



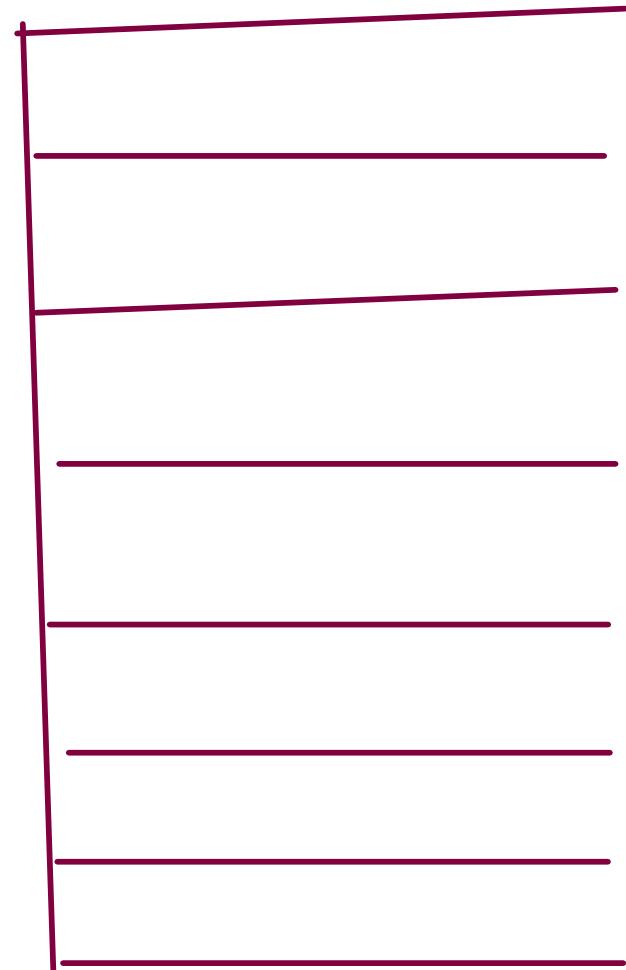
# Page Fault

1. Valid/Invalid bit in PTE  
if 0, then page is not present in frame.  
Page fault occurs -



Process  
Pages

Process  $\rightarrow$  9-  
0  
1  
2  
3



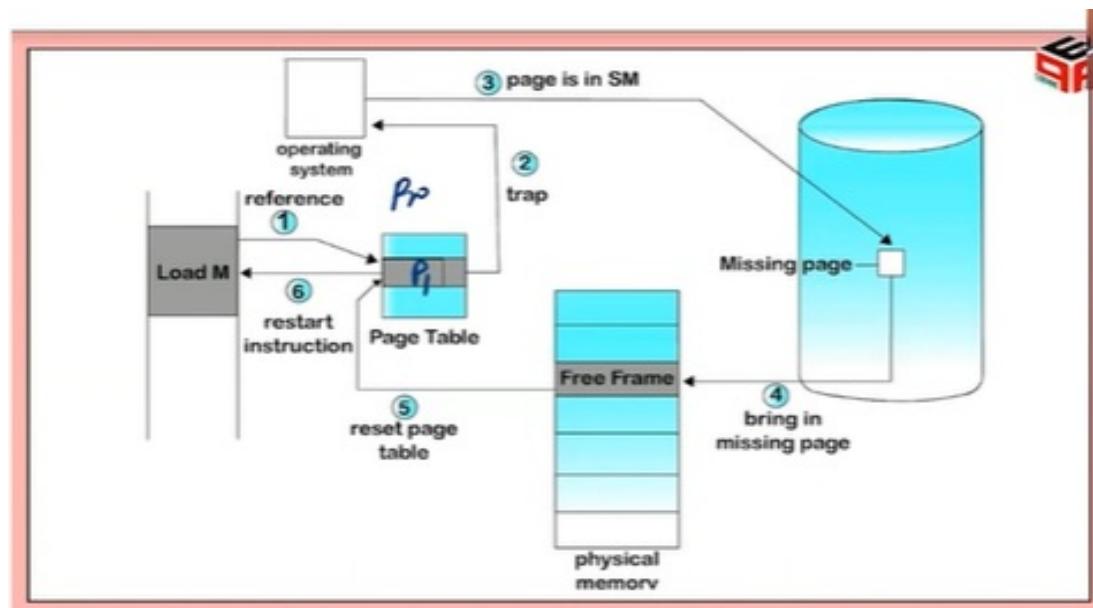
M.M  $\rightarrow$   
frames

$P_1, P_2, P_3$

Page fault

# Handling a Page Fault

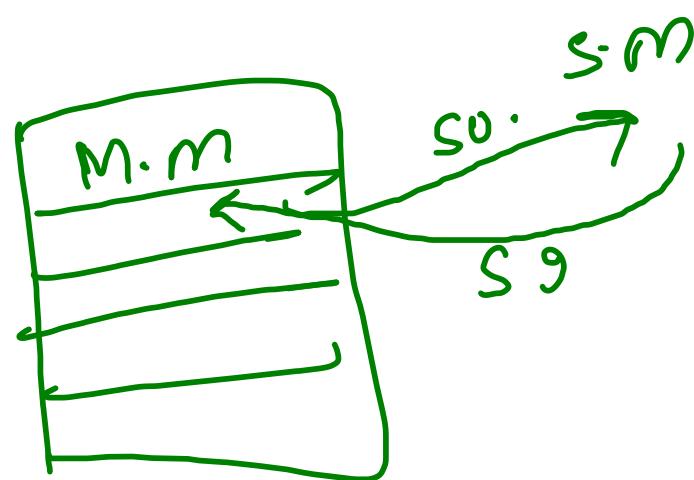
- 1) Check for the page's location in Page Table
- 2) If valid bit is 0, page fault occurs  
Trap the OS.
- 3) Check if authentic user is asking for that page or not.
- 4) If yes, load the page from Sec. memory to M.M.
- 5) Reset page table
- 6) Restart instruction



# Swapping | Page replacement Algo.

free frame not available , so existing pages are swapped out

swap out , swap in.



- 1) optimal Page replacement -
- 2) FIFO
- 3) LRU
- 4) MRU

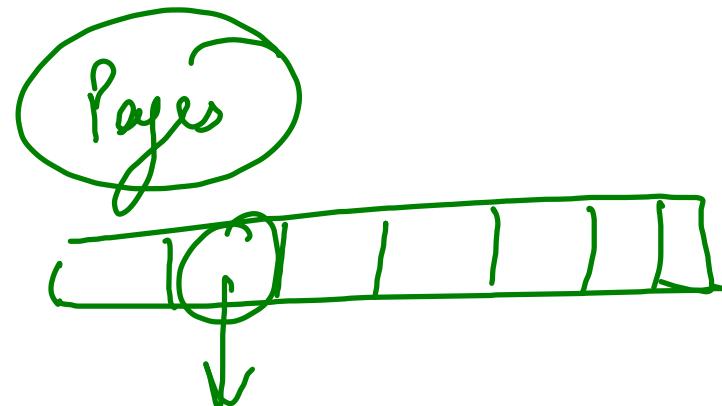
- 1) optimal page replacement
- 2) FIFO
- 3) LRU
- 4) MRU

S.M



# Optimal Page Replacement

Principle:- to be swapped out with a page which is not required in future for the longest time gap, or not required in the future at all.



future  
Page X.

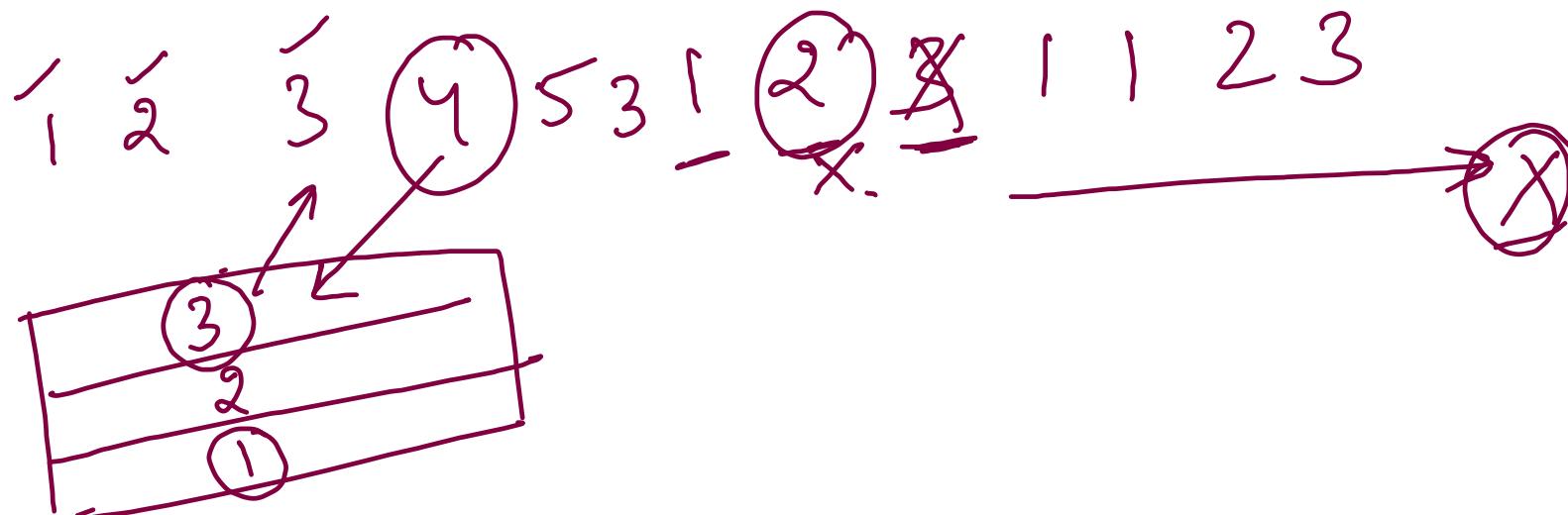
Optimal Page Replacement Algo														
m m m m h h m h h m m h														
1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5,														
f1	1 1 1 1 1 1 1 1 1 3 4 4													
f2	2 2 2 2 2 2 2 2 2 2 2 2 2													
f3	3 4 4 9 5 5 5 5 5 2 5 5													
(12) →	5 → h 7 → m													

RPV → Page Tab  
1 → FR  
tmp → OS

Principle → replace the req. page with a page that won't be used for the largest time gap, won't be used ever now.

Valid 11  
4 hit

→  
miss  
fault



m m m m m m h h m m h  
 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5,

	3	3	3	2	2	2	2	4
	2	2	2	1	1	1	3	3
+	1	1	4	4	5	5	5	3

$P_i = m \cdot 9$   
 $P_h = 3$

Queue  
 X  
 Z  
 Y  
 X  
 X  
 2 5 3 4

No. of frame  $T$ , no. of page  $n+P$   
page miss  $\downarrow$

Belady's anomaly  $\rightarrow$

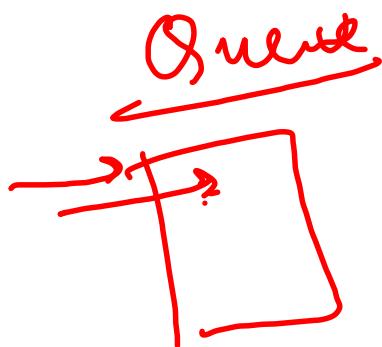
Q

FIFO - First in First Out

'Jo sabse pehle aaya, sabse pehle  
bahar bhi jaega'

- the page which has been present in memory for the longest time is replaced.
- Independents of the locality of the process.

Swap out



**1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5,**

**1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5,**

## Belady's anomaly.

- Inc. in no. of frames do not guarantee dec. page miss
- Page miss or can inc. with inc. in no. of frames

# Belady's anomaly

FIFO  
 3 frame  
 $P.h = 3$   
 $P.m = 9$

Sequence: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5, ...

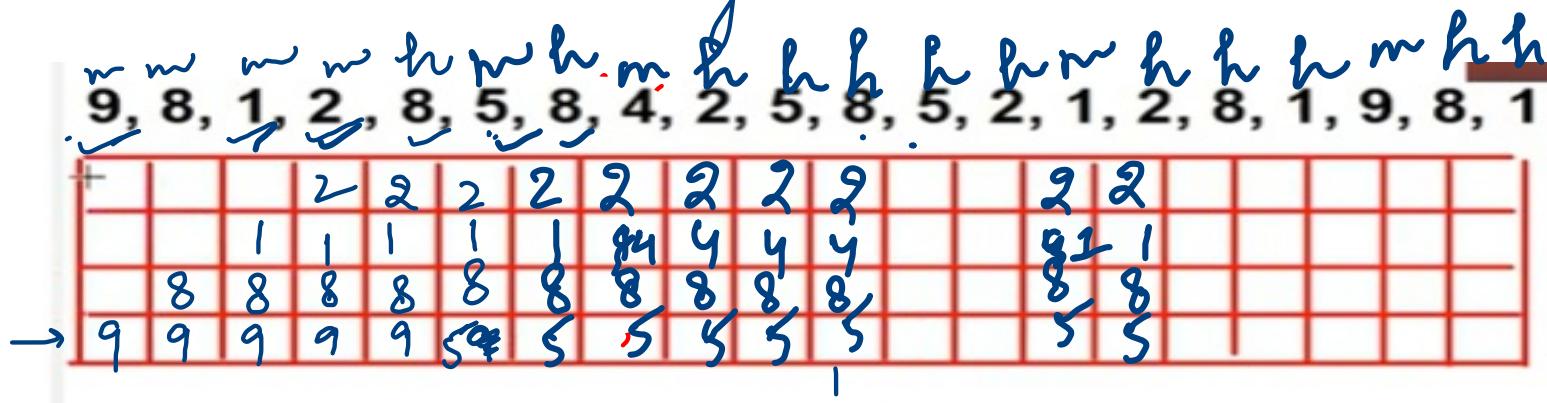
$P.h = 2$   
 $P.m = 10$

Buene  
 X  
 2  
 3  
 4  
 5  
 1  
 2  
 3  
 4

Least Recently used :-

- Jo itne time se use nahi hua, ab kya hogा.  
WTF  $\rightarrow$  Sabr door

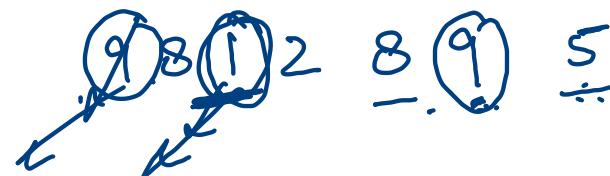
left → Sabre door



FIFO  
LRU

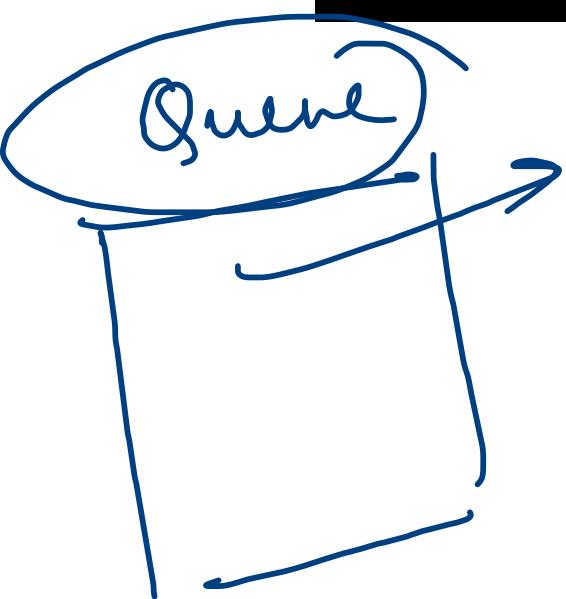
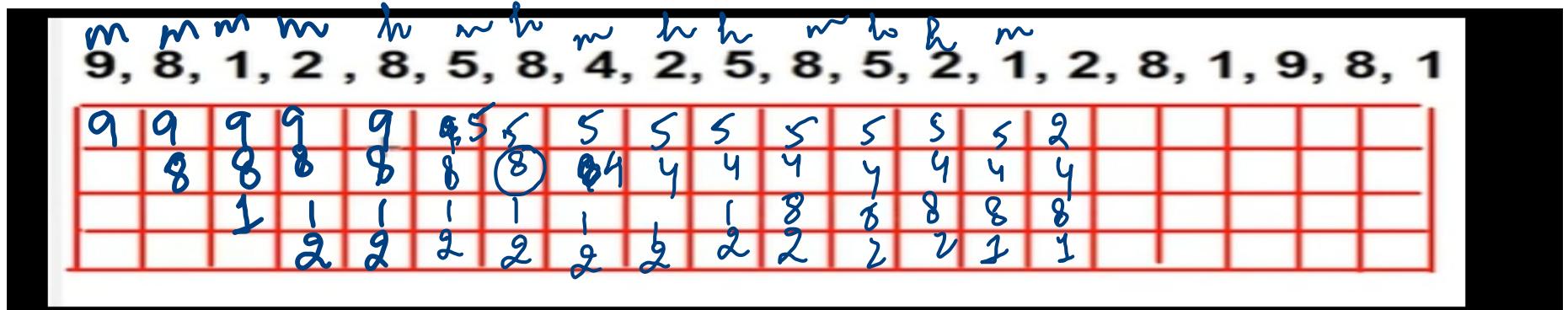
f(x)  
Quelle  
g.

LRJ

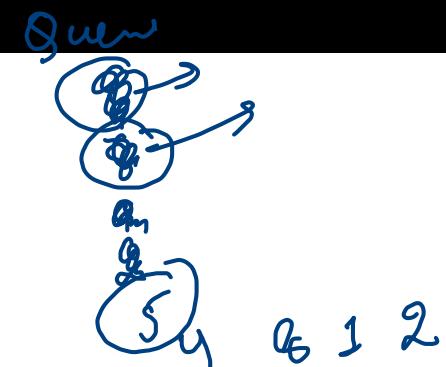


P.H. ) P.M. = ?

# Most Recently Used

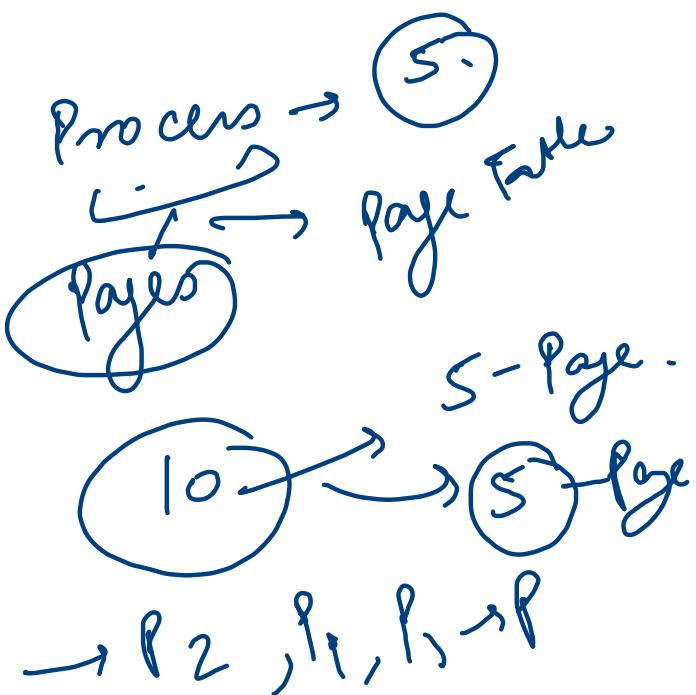
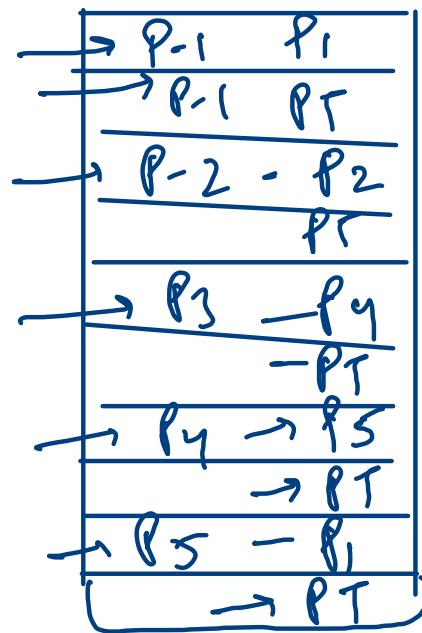
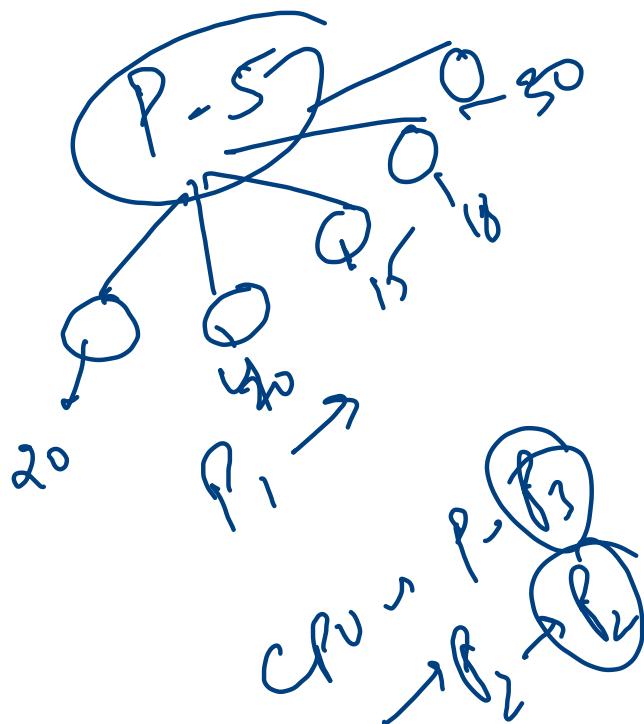
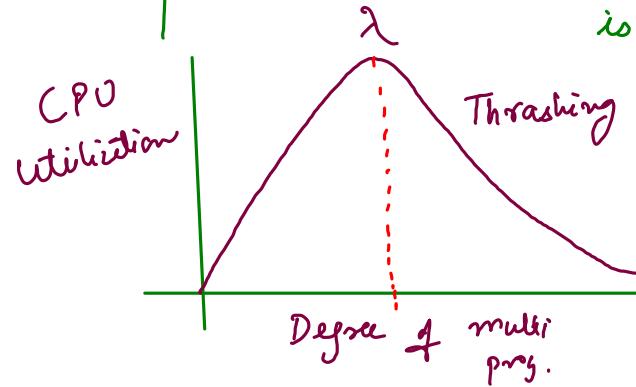


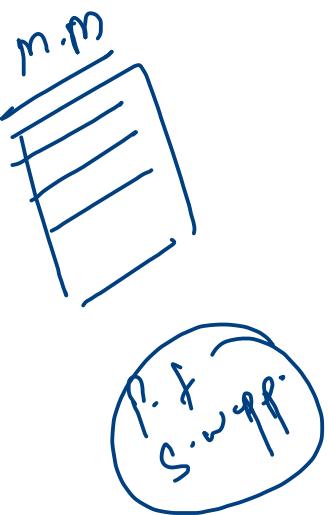
FIFO



# Thrashing

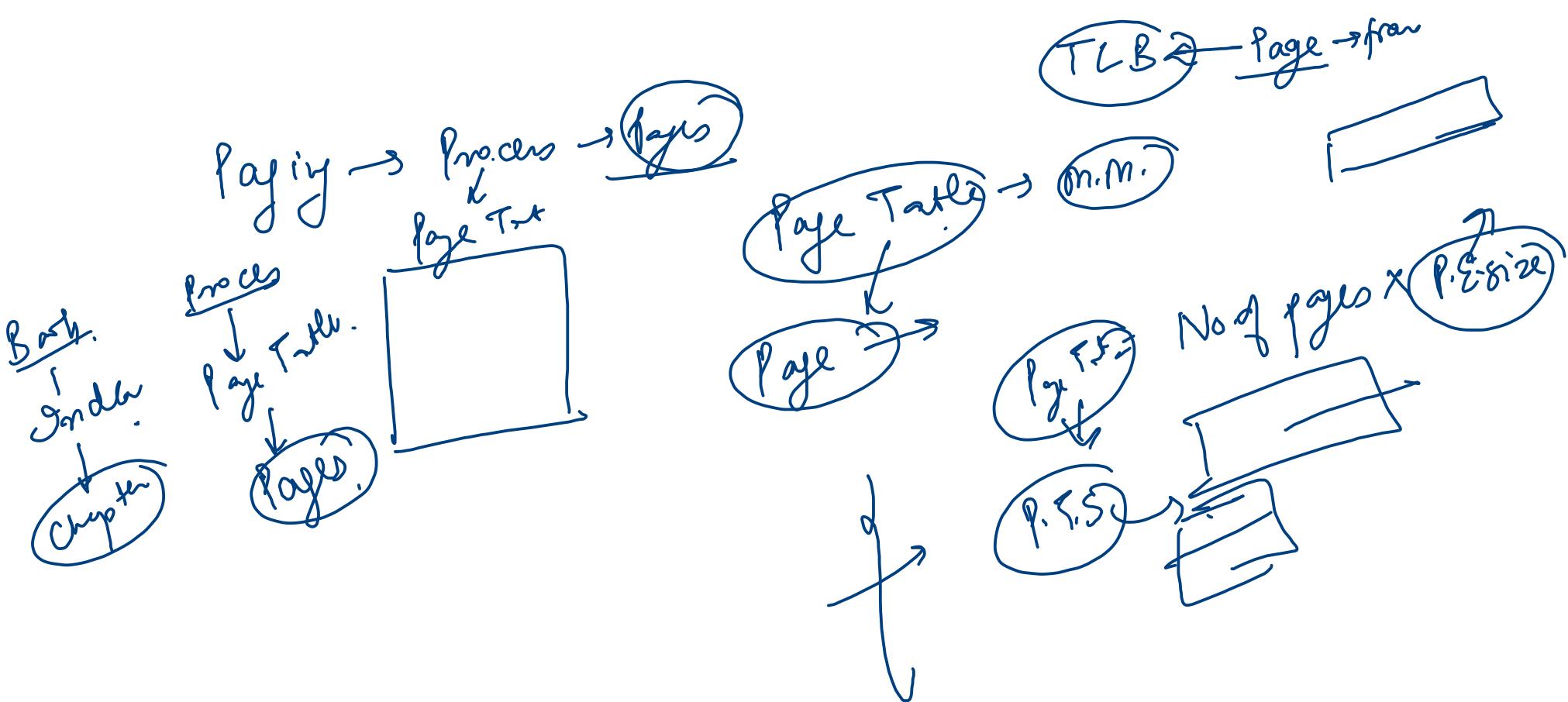
To inc. CPU utilization  $\propto$  multiprogramming  
is done.

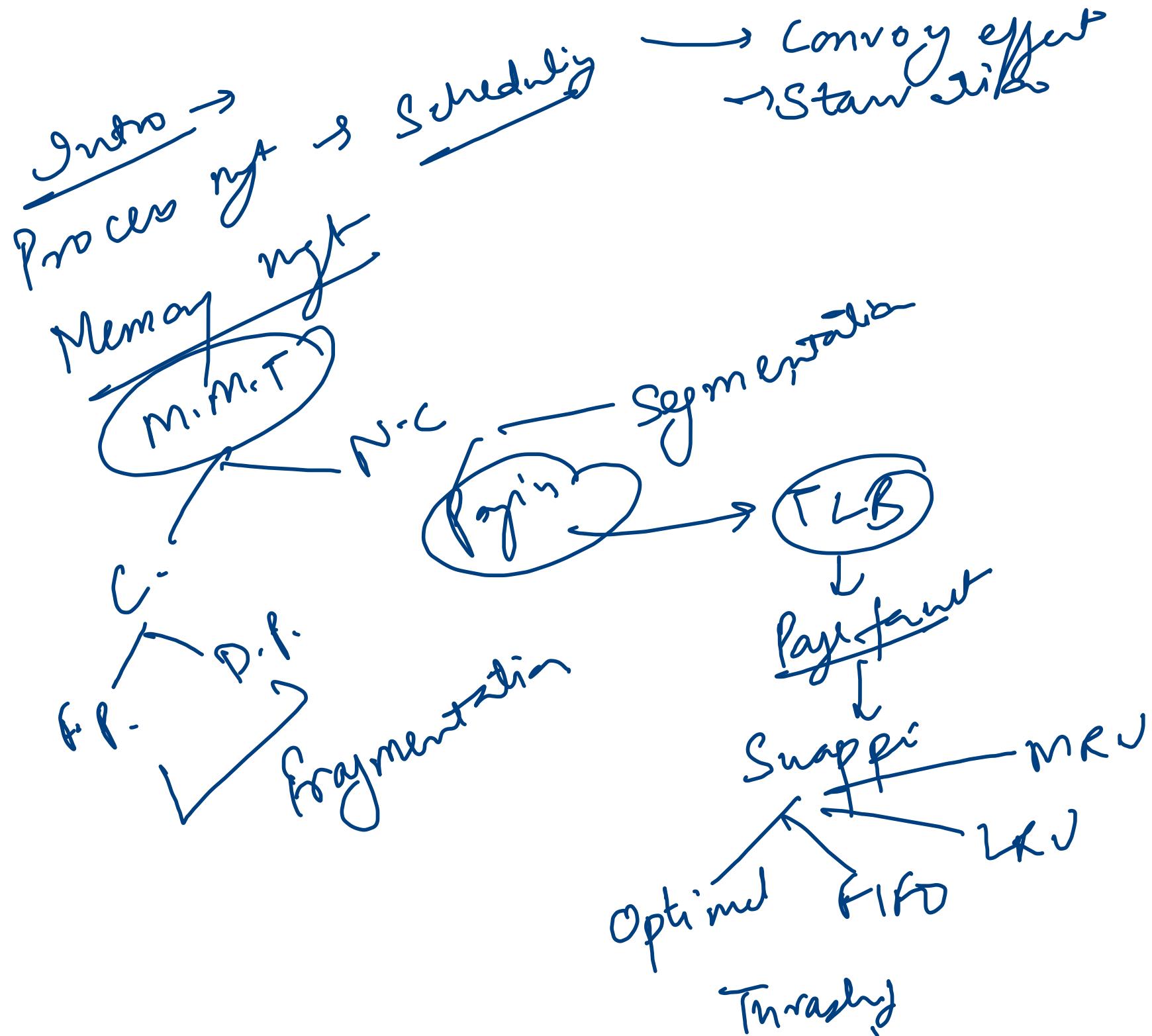




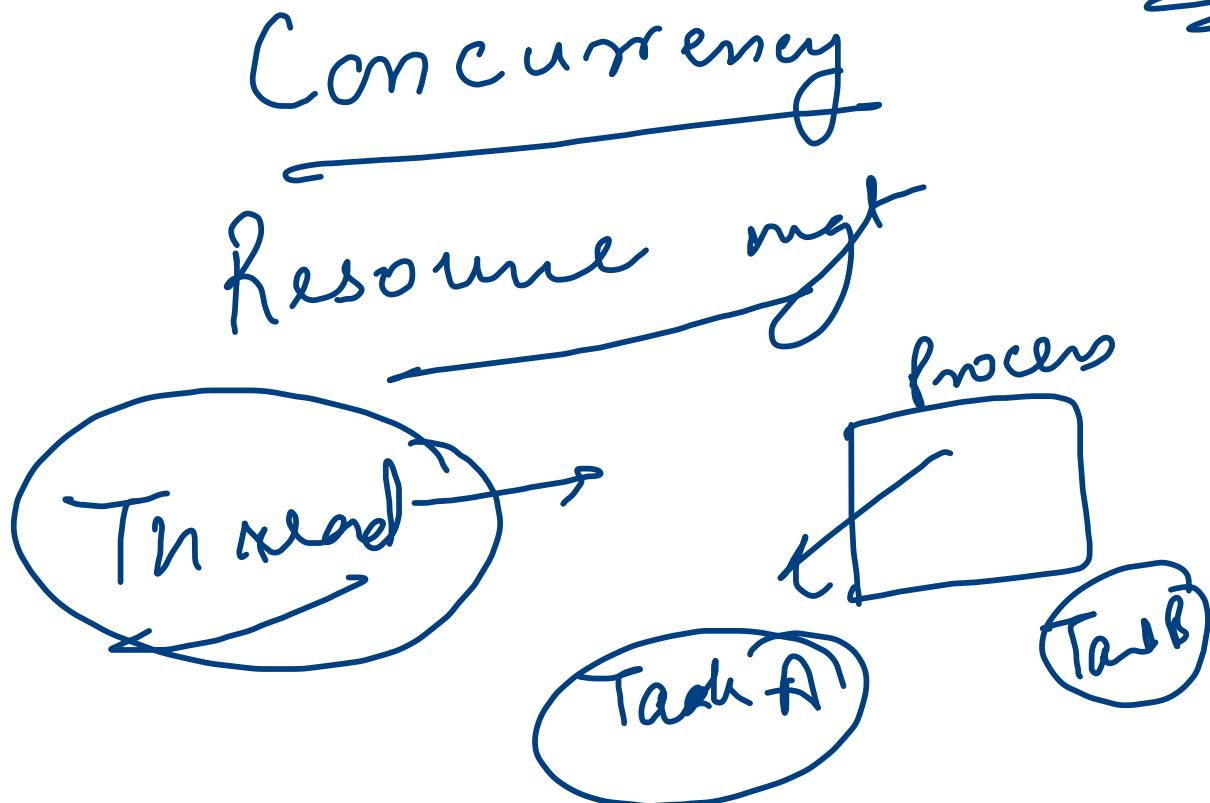
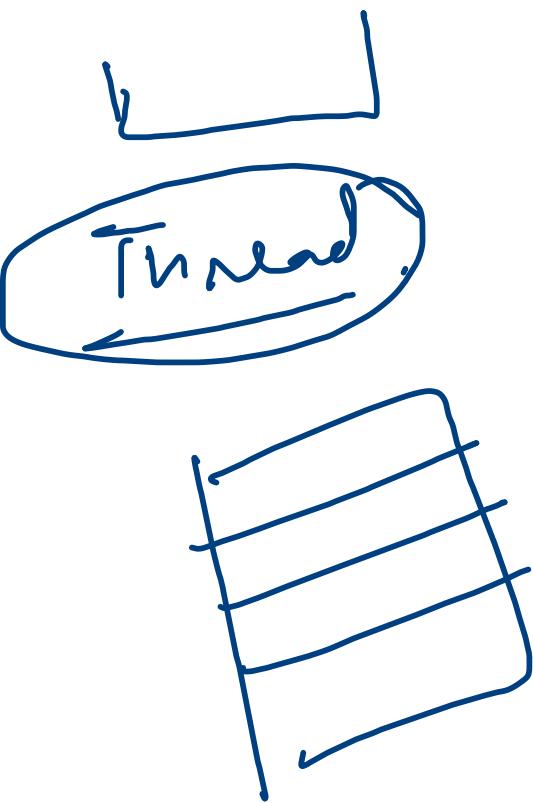
Page table → No. of pages

Pagesize = frame size





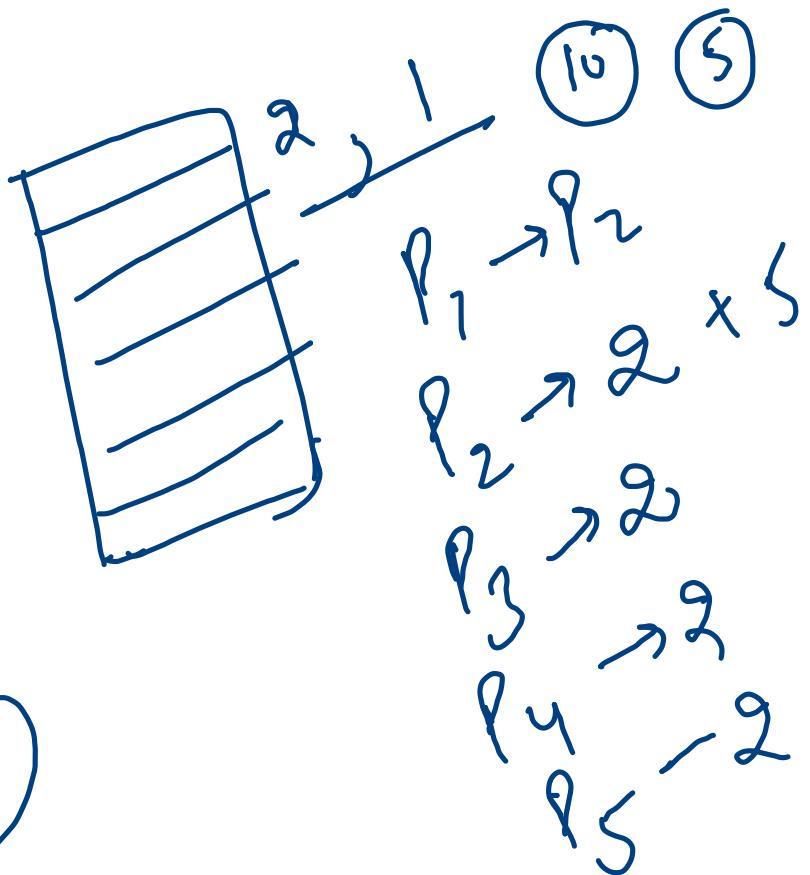
OS - Test



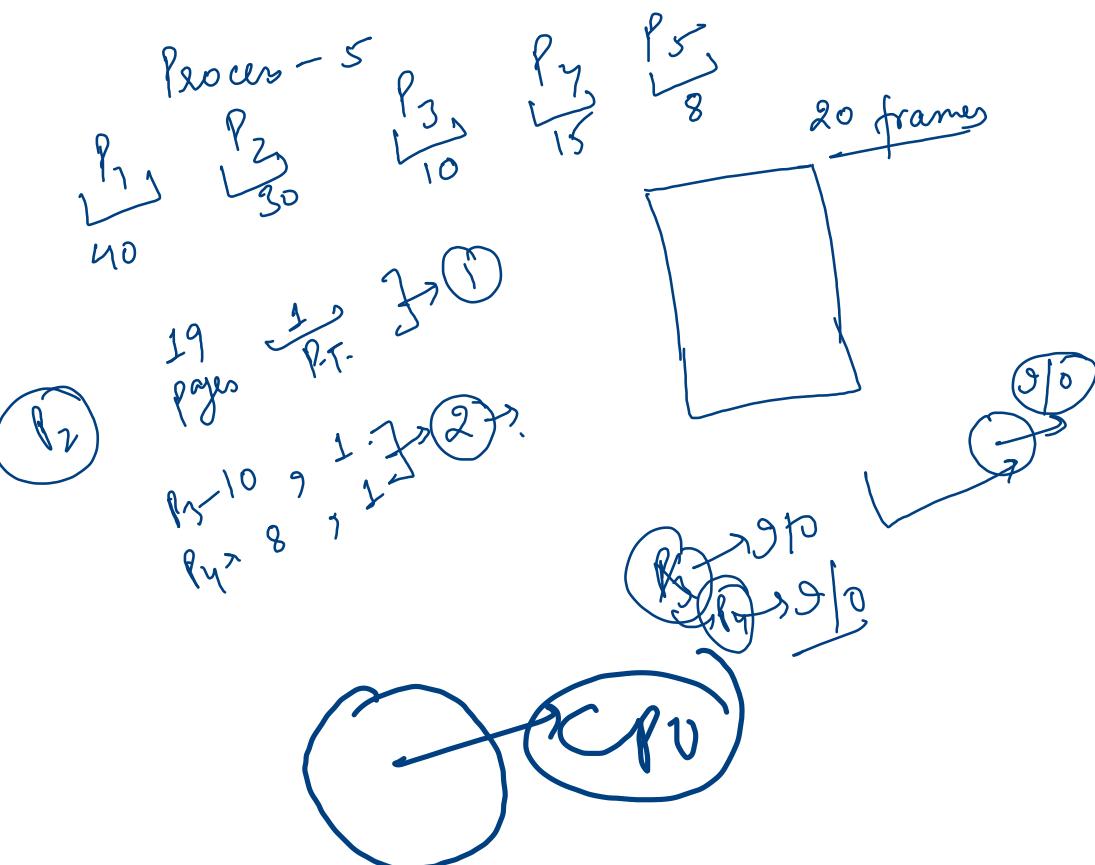
Optimal







5





i) Page size = 4 kB

Page table entry takes = 4 B

How many levels of page tables will be required to map 32 bit address space if every page fits into a single page.