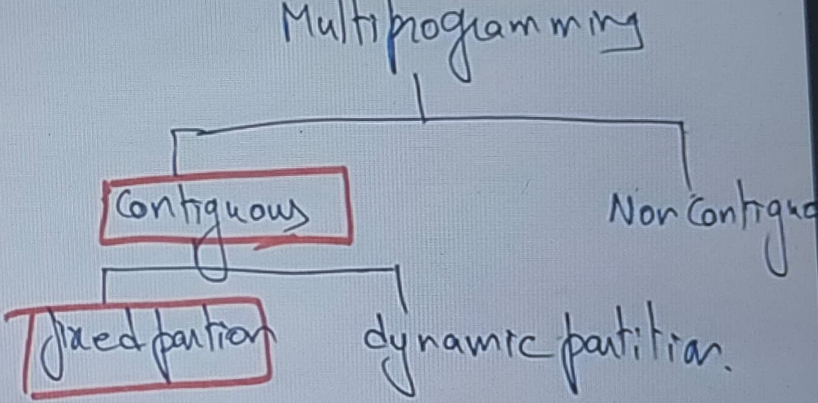
Fixed Partitioning

⇒ M.M is divided into no: of static partitions at sys generation time.

- Equal size partition
- Un Equal size partion.

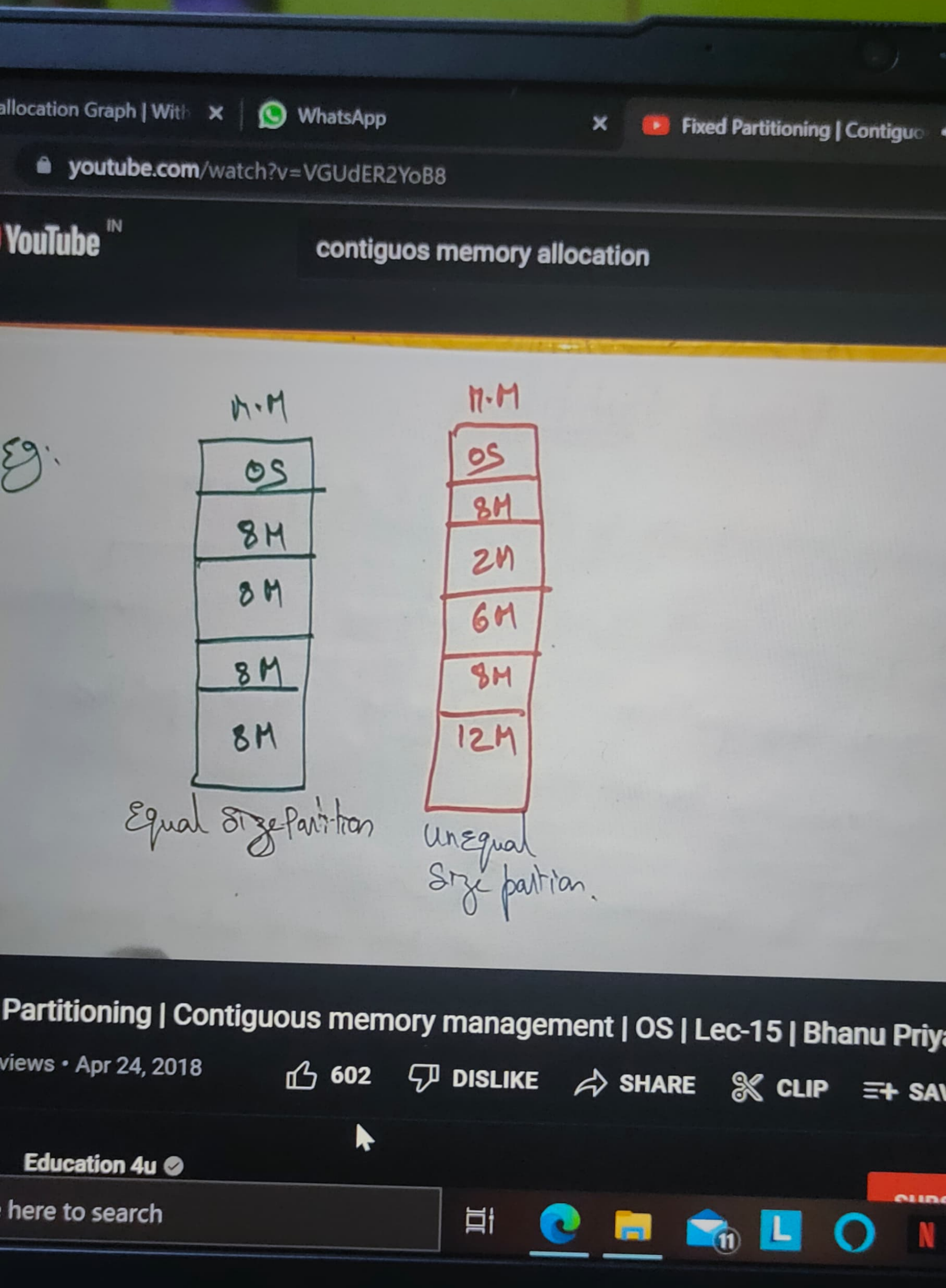


Pros:-

- simple to implement
- little os overhead.

Cons:-

- In efficient
- Max no of active process



allocation Graph | With ✕ WhatsApp ✕ Fixed Partitioning | Contiguo

youtube.com/watch?v=VGUdER2YoB8

YouTube IN contiguos memory allocation

Eg:

M.M
OS
8M
8M
8M
8M

Equal Size Partition

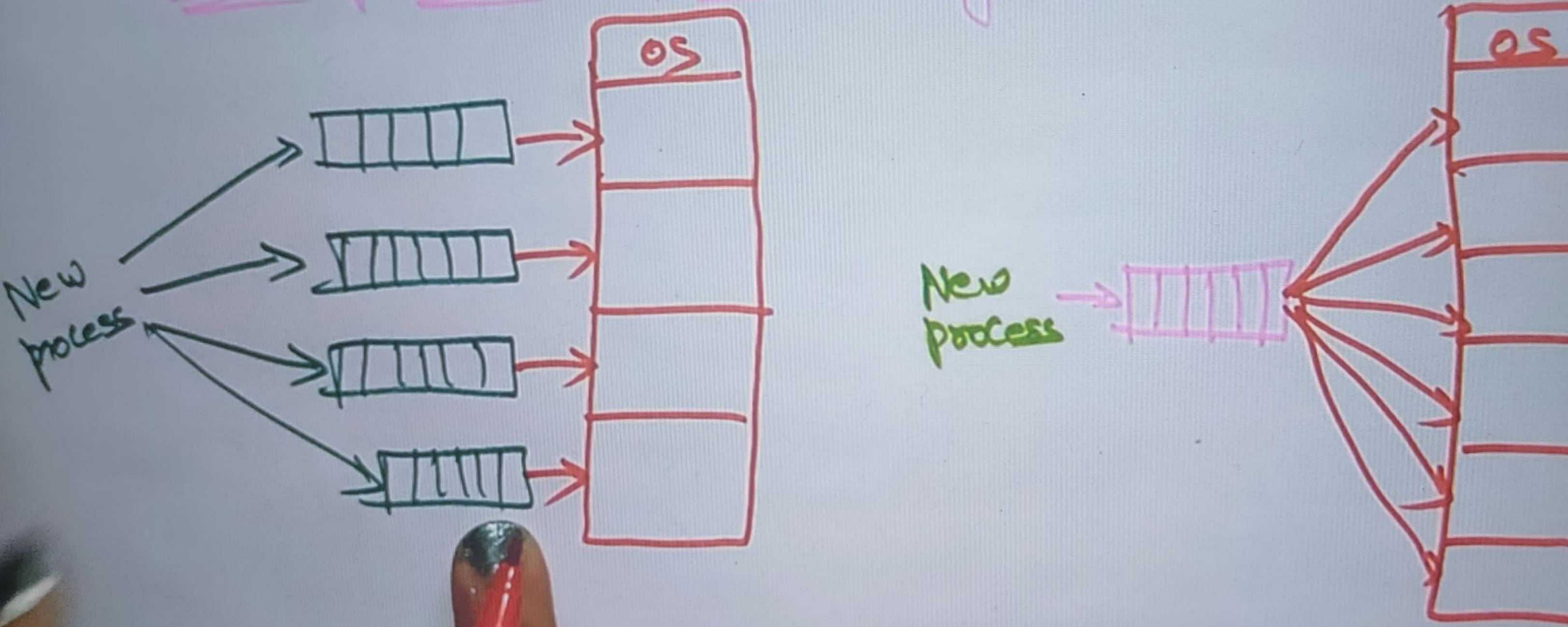
M.M
OS
8M
2M
6M
8M
12M

unequal Size partion.

Partitioning | Contiguous memory management | OS | Lec-15 | Bhanu Priya

views • Apr 24, 2018

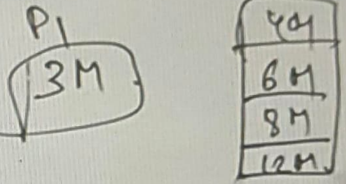
Fixed partitioning placement Algorithm



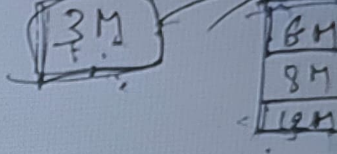
Dynamic Partitioning

⇓

Partitions are created dynamically.



Each process is loaded into a partition of exactly the ~~size~~ same size of that process



⇓

Partitions are created dynamically.

Each process is loaded into a partition of exactly the ~~size~~ same size of that process

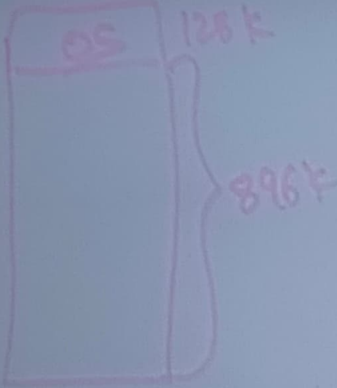
Placement Algorithm:

First fit: first hole that is big enough is allocated to prog.

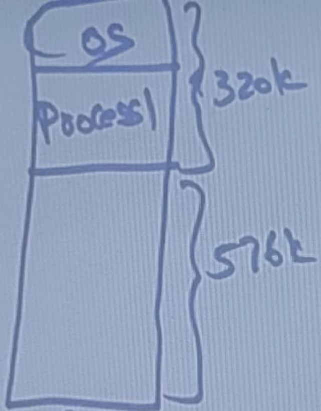
Best fit: Smallest hole that is big enough is allocated to prog

Worst fit: largest hole that is big enough is alloc

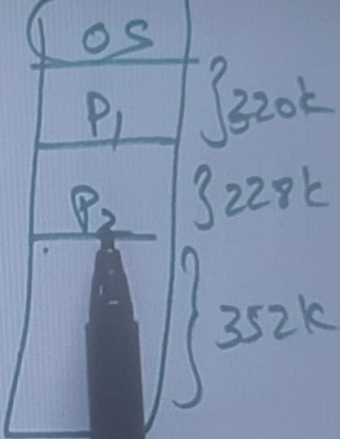
Eg:


OS
128k
896k

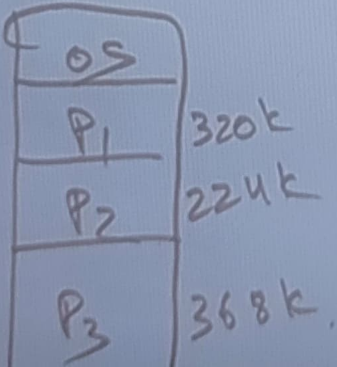
(a)


OS
Process1
320k
576k

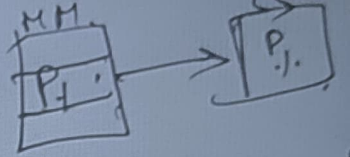
(b)


OS
P1
320k
P2
228k
352k

(c


OS
P1
320k
P2
224k
P3
368k.

Fragmentation

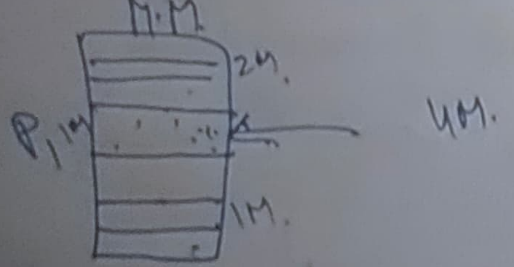


A processes are loaded & removed from memory the free memory space is broken into little pieces.

⇒ After some times that processes cannot be allocated to memory blks of small size & memory blk remains unused.

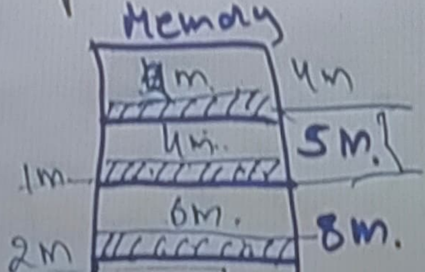
External fragmentation

Total memory space is enough to satisfy a request or reside process in it, but it is not ntiguous, so it cannot be used



Internal fragmentation

⇒ Memory block assigned to process is bigger. Some portion of memory is left unused, as it cannot be used by another proces.

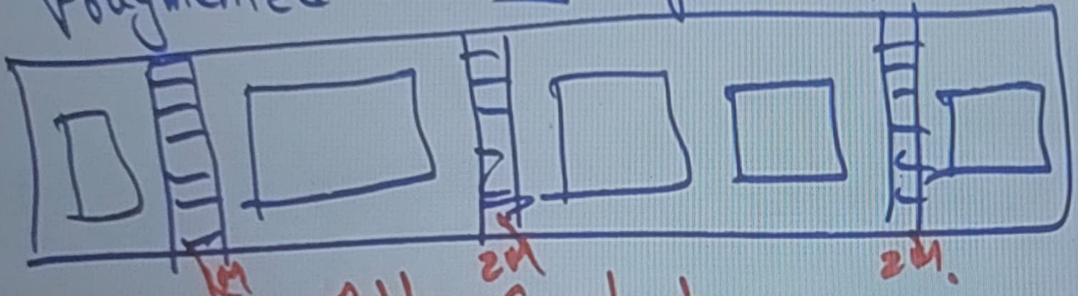


$P_1 = 4M$

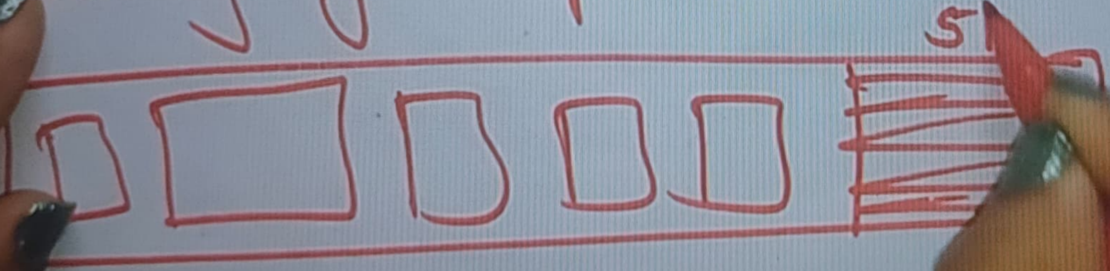
$P_2 = 6M.$

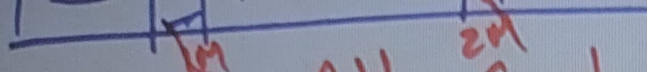
$P_3 = 3m.$

Fragmented Memory before compaction

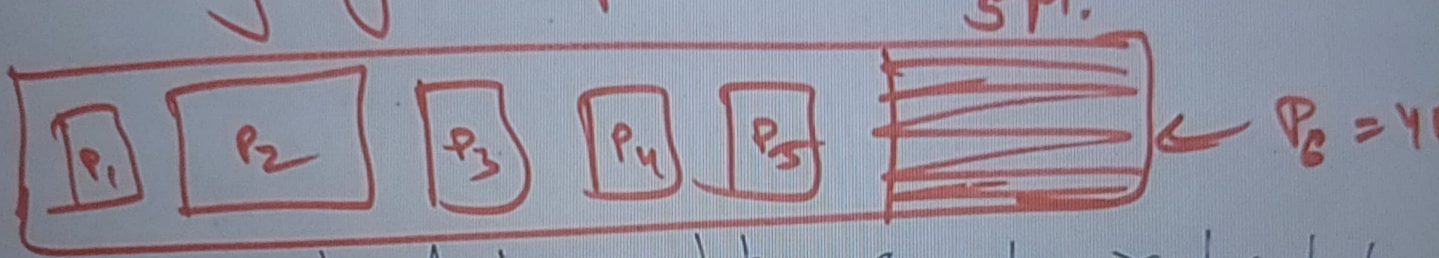


Memory After Compaction.





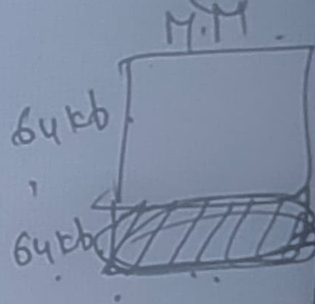
Memory After Compaction.



⇒ External fragmentation can be reduced by compaction or shuffle free memory together in one largeblock

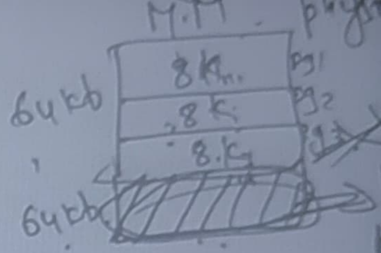
⇒ Internal fragmentation can be reduced by effectively assigning the smallest partition but large enough for the process contiguos memory allocation

* A comp can add more memory then ~~the~~ the amoun physically installed on sys this extra memory called virtual memory.



physically ins

Called virtual Memory.

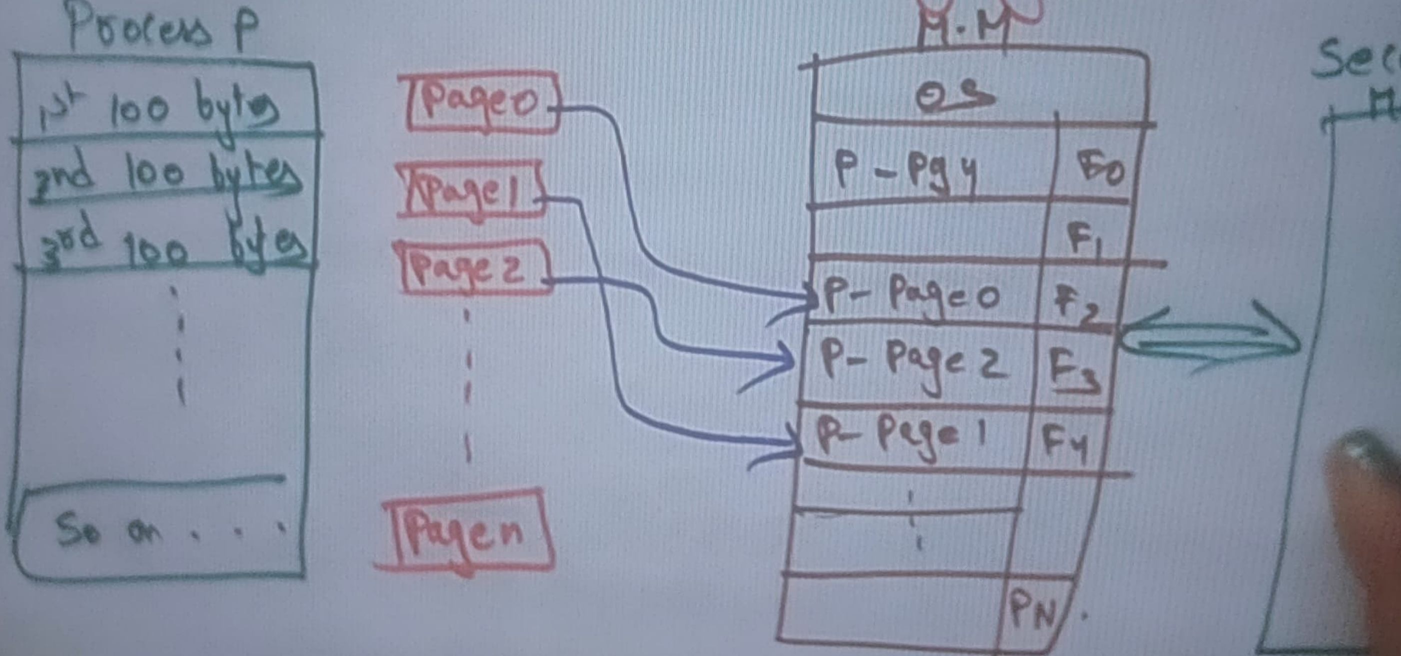


- It is a memory management technique in which process add space is broken into blocks of same size called paging. (power of 2)

- The size of process can be measured in no: of pages. ||ly M.M is divided into small fixed blocks (physicall of memory called frames.

Size of frame == Size of page

We can avoid External fragmentation.



Addration translation

Page add is logical add & represented by.

logical add = Pg no + page offset

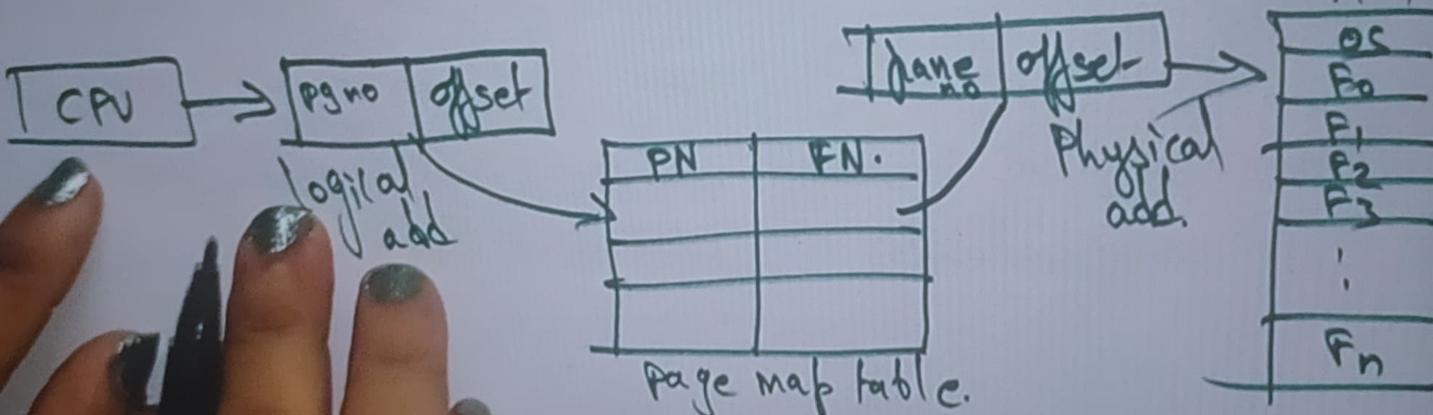
frame add is called physical add. & represen

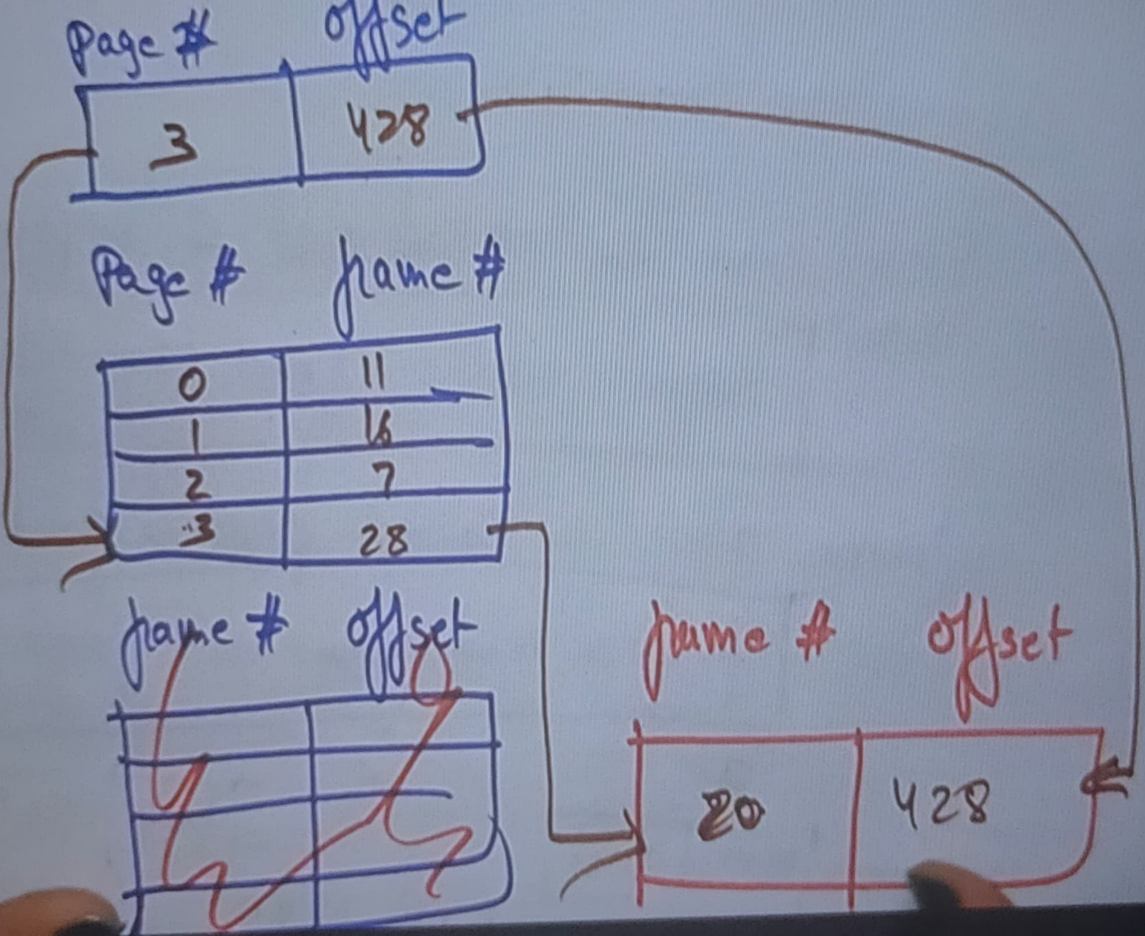
physical add = frame no + page offset

logical add = pg no + pg off

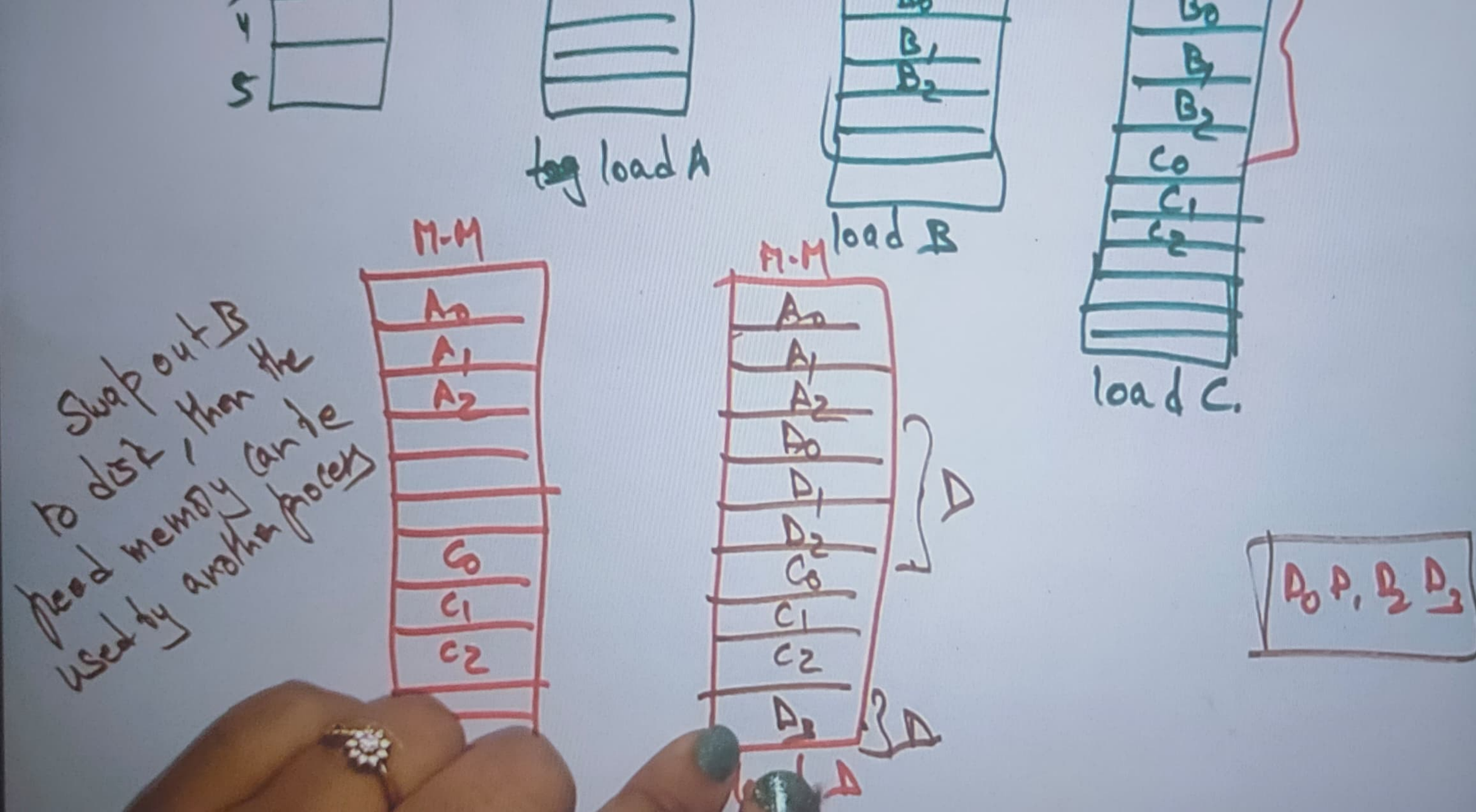
frame add is called physical add. & represent

physical add = frame no + page offset





Page #	frame #
0	11
1	16
2	7
3	28



Swap out B to disk, then the freed memory can be used by another process

(1) FIFO

(2) LRU

(3) optimal

(1) FIFO page replacement Algorithm :

* replace the pages that has been in memory the longest time.

Reference string :- 7 0 1 2 0 3 0 4 2 3 0 3 1 2 0

P₁ ⇒ Pages

Page fault → *

Page hit → Hit

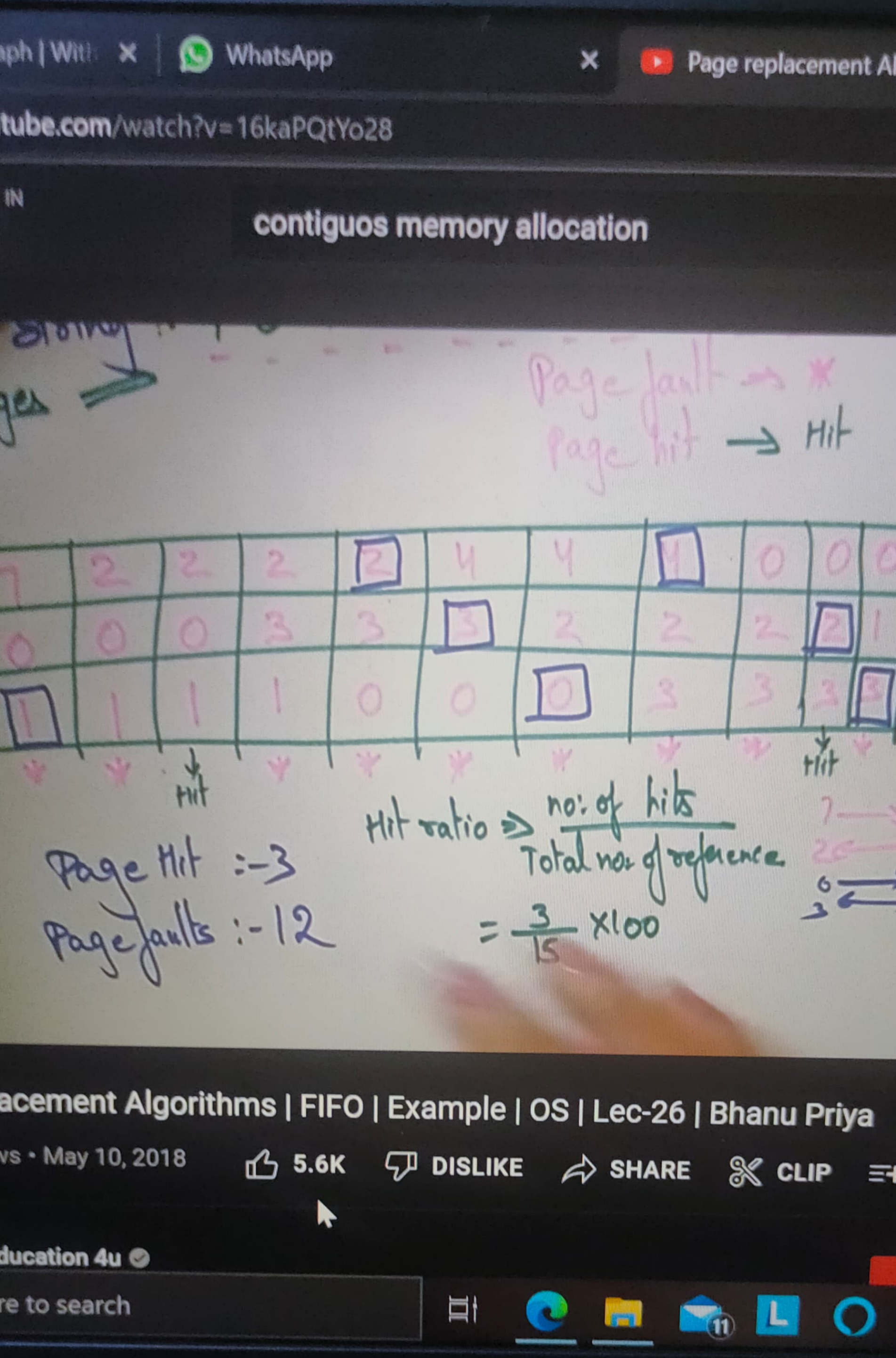
7	7	7	2	2	2	2	4	4	4	0	0	
	0	0	0	0	3	3	3	2	2	2	2	
		1	1	1	1	0	0	0	3	3	3	
*	*	*	*	Hit	*	*	*	*	*	*	Hit	

7 → disk

2 ← 2

0 → 3

3 ← 3



ges →

Page fault → *
Page hit → Hit

7	2	2	2	$\boxed{2}$	4	4	$\boxed{4}$	0	0	0
0	0	0	3	3	$\boxed{3}$	2	2	2	$\boxed{2}$	1
$\boxed{1}$	1	1	1	0	0	$\boxed{0}$	3	3	3	$\boxed{3}$
*	*	Hit	*	*	*	*	*	*	Hit	*

Page Hit :- 3

Page faults :- 12

Hit ratio ⇒ $\frac{\text{no. of hits}}{\text{Total no. of reference}}$

$= \frac{3}{15} \times 100$

acement Algorithms | FIFO | Example | OS | Lec-26 | Bhanu Priya

vs • May 10, 2018

LRU (Least Recently Used)

Reference string :

7	0	1	2	0	3	0	4	2	3	0	$\boxed{3}$	1	2	0
7	7	$\boxed{7}$	2	2	2	$\boxed{2}$	4	4	$\boxed{4}$	0	0	$\boxed{0}$	2	2
	0	0	0	0	0	0	0	$\boxed{0}$	3	3	3	3	$\boxed{3}$	0
		1	1	$\boxed{1}$	3	3	$\boxed{3}$	2	2	2	$\boxed{2}$	1	1	1
*	*	*	*	↑ Hit	*	↑ Hit	*	*	*	*	↑ Hit	*	*	*

Page Hit = 3

fault = 12

Optimal Page Replacement Algorithm

ference string:-	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7
	7	7	7	2	2	2	2	2	2	2	2	2	2	2	2	2	2	7
		0	0	0	0	0	0	4	4	4	0	0	0	0	0	0	0	0
			1	1	1	3	3	3	3	3	3	3	3	1	1	1	1	1
	*	*	*	*	↑ Hit	*	↑ Hit	*	↑ Hit	↑ Hit	*	↑ Hit	↑ Hit	*	↑ Hit	↑ Hit	↑ Hit	*