COMPUTER SCIENCE

Operating System

Virtual Memory

Lecture No:07

Dr. KHALEEL KHAN SIR

Virtual memory

VIRTUAL MEMORY

LRU Approximations : (They approximate to the behaviour of LRU)

a) Reference Bit (R)
b) Additional Ref bits
c) Second chance (clock Algo)
d) Enhanced Second chance (Not Recently Used) (NRU)

Reference Bit (R)

1) Reference bit (R)

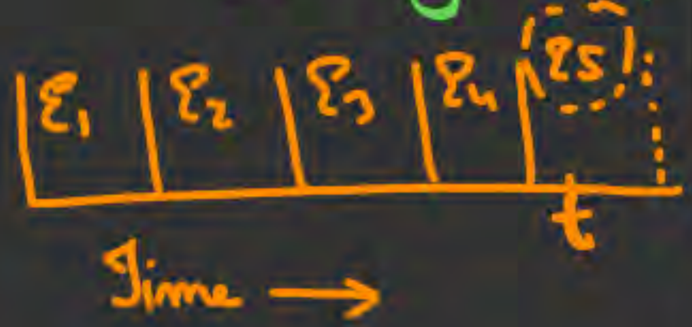
Criteria: R

- 0 → Page has not been Referred during the Present Epoch;
- 1 → Page has been referred atleast once during the present Epoch;

Start Search from first entry of P.T to find a Page whose R value is '0';

P	f	V/I	T.o.L	R	
0	a	1	1	1	
✓1	b	1	3	1	
2	–	0	–	–	
3	k	1	0	1	
4	l	1	2	1	

Page Table



When all Values of 'R' is one ⟹ "Fails"

(ii) Additional Ref. bits

Each page is associated with multiple Ref. bits

– 8 'R' bits.

	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
P_a	1	1	0	1	1	1	1	1
P_b	0	0	1	1	1	1	1	1
P_c	1	0	1	0	1	1	1	1

P_z × → P.F

Current Epoch

P_a ×

P_c ✓

When Curr. Epoch gets over

All 'R' values of Curr. Epoch are cleared to '0' & they are Shifted left

③ Second Chance

⊛ Clock Algorithm

R : 1 ✓

→ P_5

→ P_3 ✓

Criteria : T.O.L + R

→ FIFO : → Belady's Anomaly

P	f	V/I	T.O.L	R	M	
0	a	1	2	1	0	
1	b	1	4	1	0	
2	–	0	–	–	–	
3	c	1	0	0	1	
4	d	1	3	1	0	
5	f	1	1	0	1	

: P_4 ✓

: P_3

4) Enhanced Second Chance

NRU :

Criteria : R + M

Modified bit

R	M	
0	0	– I
0	1	– II
1	0	– III
1	1	– IV

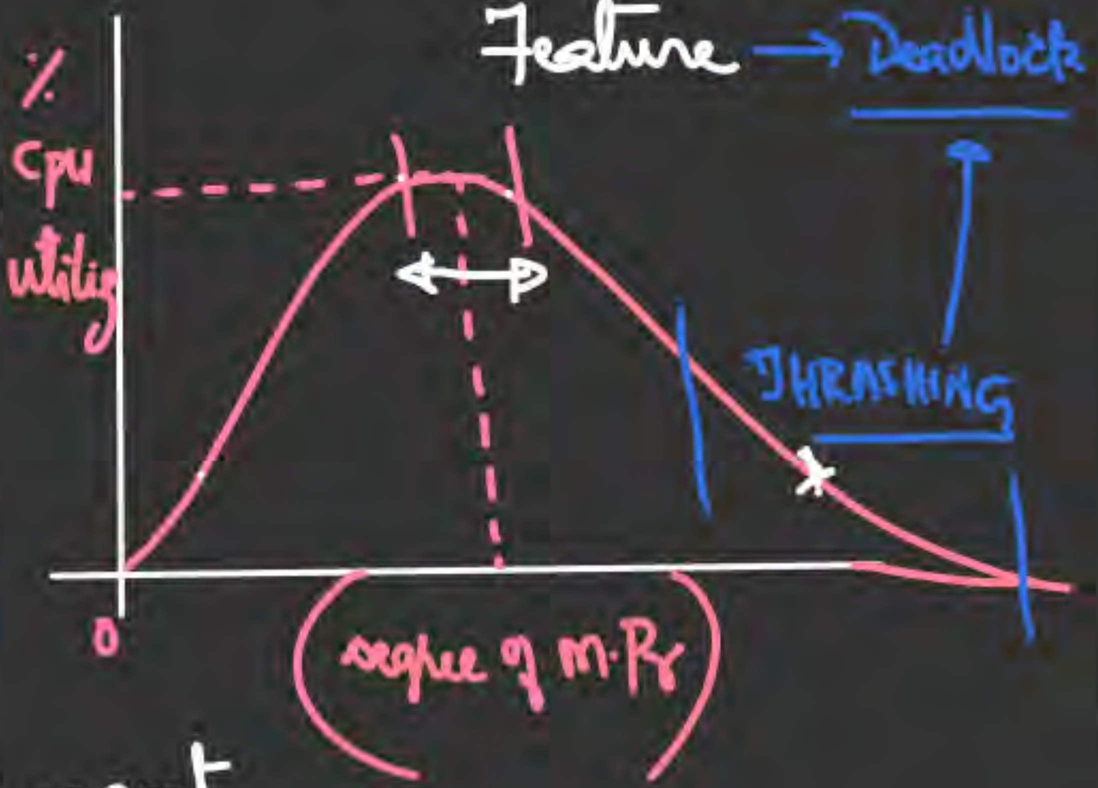
LRU

Thrashing: Excessive/High Paging activity;

<very high Page fault rate of Processes in the System>

↳ Act of servicing the Page fault;

→ is an undesirable Feature → Deadlock



Reasons:

I. Primary:

- ✓ → Lack of Frames (Memory)
- ✓ → High degree of M.Pr

II. Other Reasons:

→ Page replacement Algo.

→ Page-Size: large → (No. of Pages will be less)

Thrashing Control

Strategies

Prevention

↓

Controlling degree of M.Pr

<LTS>

Detection & Recovery

→ Low cpu utiliz.

→ High degree of M.Pr

→ High Paging disk utiliz.

→ Suspension (MTS)

Programming Techniques & Data Structures

① int $A[1..128, 1..128]$; P.S = 128; FIFO; P.D.P; M = 127 frames

128×128

$N_{pages} = 128$

$(128)^2$ Page faults

a) for $i \leftarrow 1$ to 128 :

  for $j \leftarrow 1$ to 128

  $A[j,i] = 1;$ } (128)

CMO

---

b) for $i \leftarrow 1$ to 128 : $(128) \cdot 1$

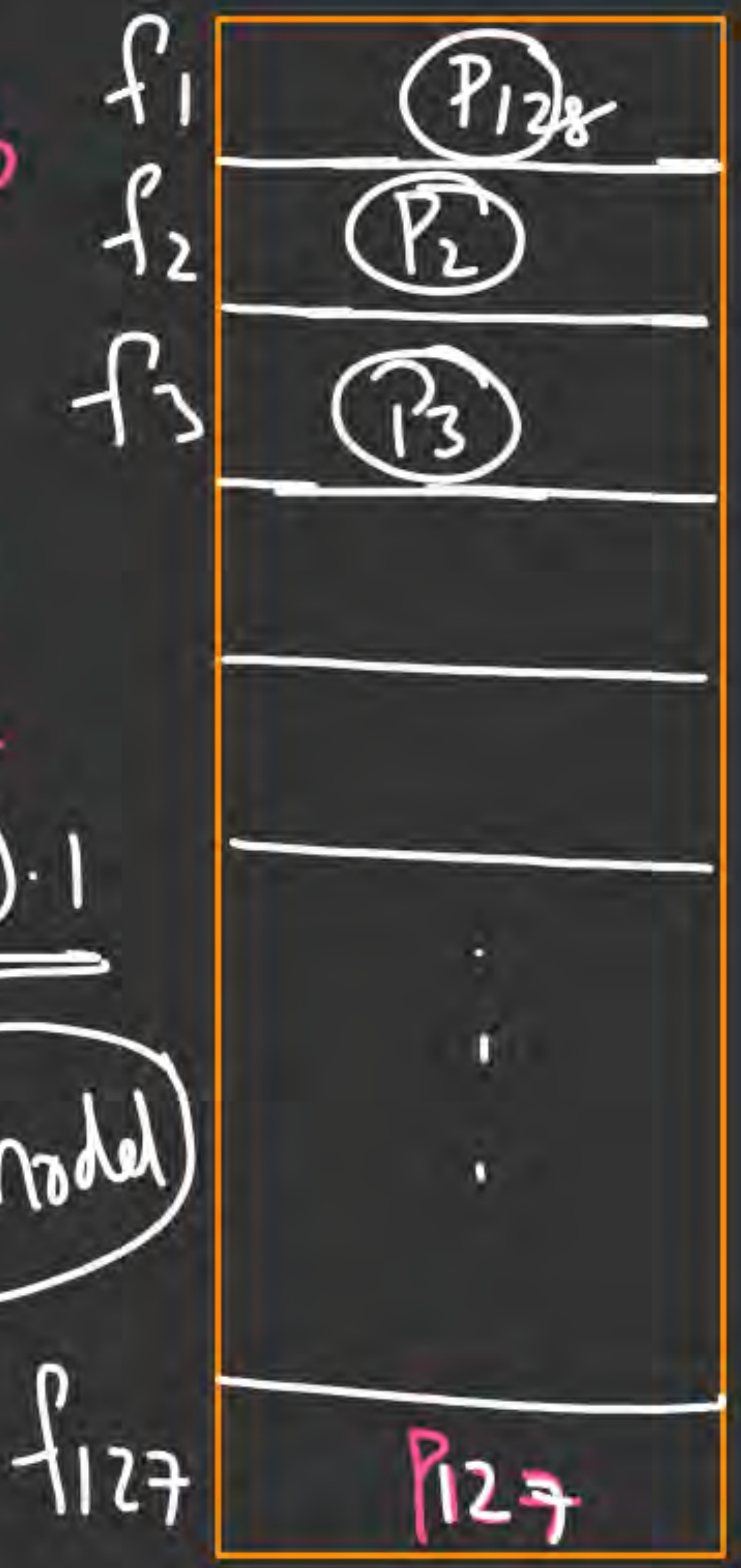
  for $j \leftarrow 1$ to 128

  $A[i,j] = 1;$

Efficient

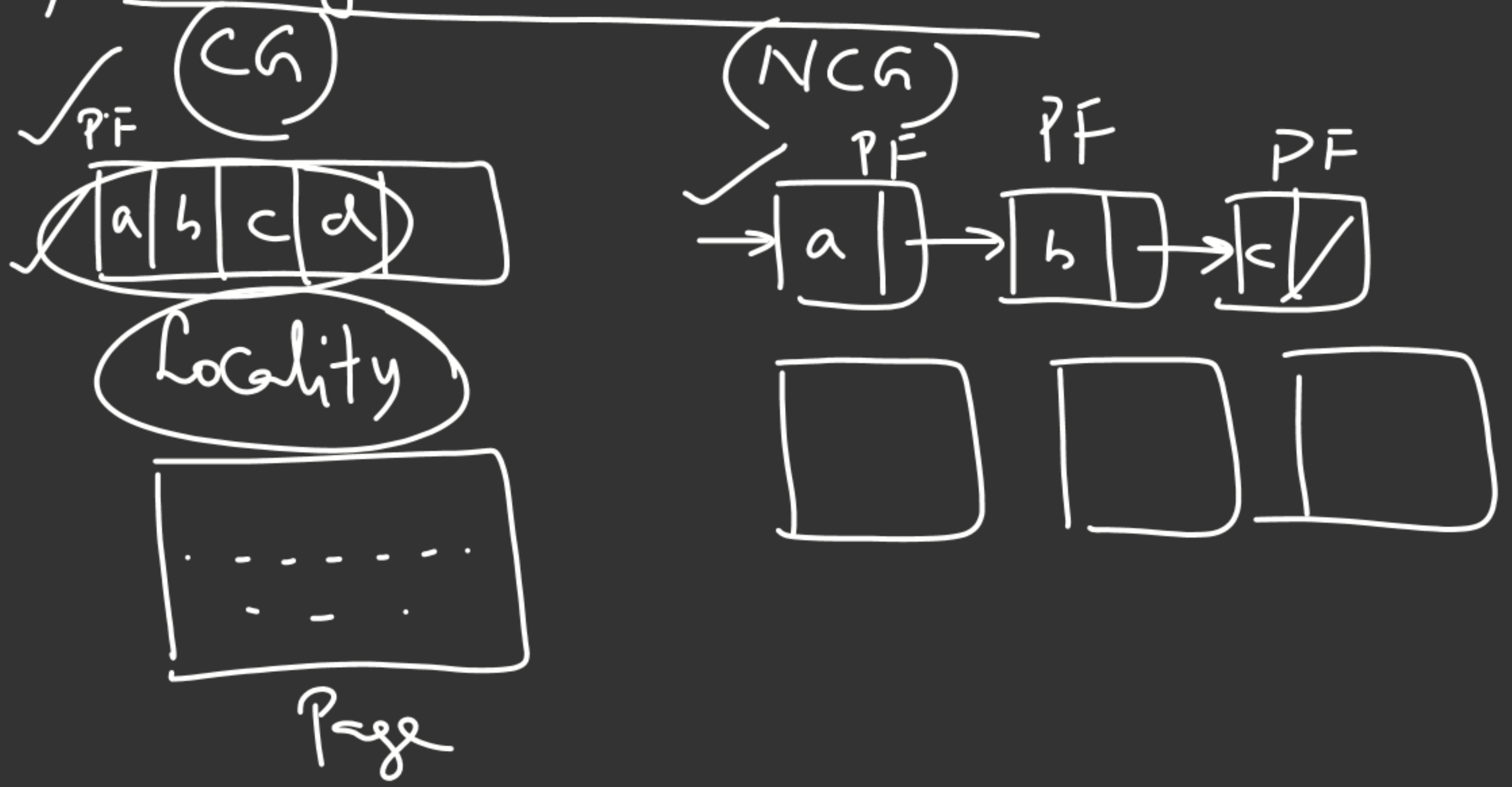
RMO

Locality Model



D.P

II Arrays vs Linked list



WORKING SET MODEL (helps Reduce P.F rate & utilizes Memory Efficiently)

(Locality of Reference)

```
10KB
main( )        | g( ) -20KB
{              | {
  :            |   :  ①
  f( );        |   h( )
  :            |   :
}              | }
f( ) -> 2KB    | h( ) -5KB
{              | {
  :            |   :
  g( );        |   :
}              | }
```

Total Prog. Size : 37 KB

P.S = 1KB; No. of Pages = ⟨37⟩ → 18 frames (Static Alloc.)

Work Set window (WSW) = { Set of unique Pages in the ref String during the Past 'Δ' references }; Δ = integer (given)

Δ=10

R.T:

P_i : 7, 0, 1, 4, 2, 3, 0, 1, 2, 25, 23, 24, 22, 21, 44, 45, 48, 49, 50, 55, 51, 52, 45, 48, 46, ...

Δ=10 ... t

Size of locality = ⟨45, 46, 48, 49, 50, 51, 52, 55⟩

$\boxed{|WSWS_i^{'t'}| = 8 \text{ pages}}$ demand of the Process

S_i

→ n : no. of Processes

→ s_i : demand of Process for frames <WSWS$_i$>

→ D : Total demand

$$D = \sum_{i=1}^{n} s_i$$

→ M : Avail. frames

't'

I. $D \simeq M$

∴ NO THRASHING

(1000 100)

II. $D > M$ (SWAP OUT) Susp.

∴ THRASHING

III. $D < M$

∴ NO THRASHING

<Inc. degree of M.Pr>

Q.1 Consider a virtual memory system with FIFO page replacement policy, for an arbitrary page access pattern, increasing the number of page frames in main memory will

11:45 am

- (A) Always decrease the number of page faults.
- (B) Always increase the number of page faults
- (C) Sometimes increase the number of page faults ✓ (Belady's Anamoly)
- (D) Never affect the number of page faults.

Q.2 The minimum number of page frames that must be allocated to a running process in a virtual memory environment is determined by

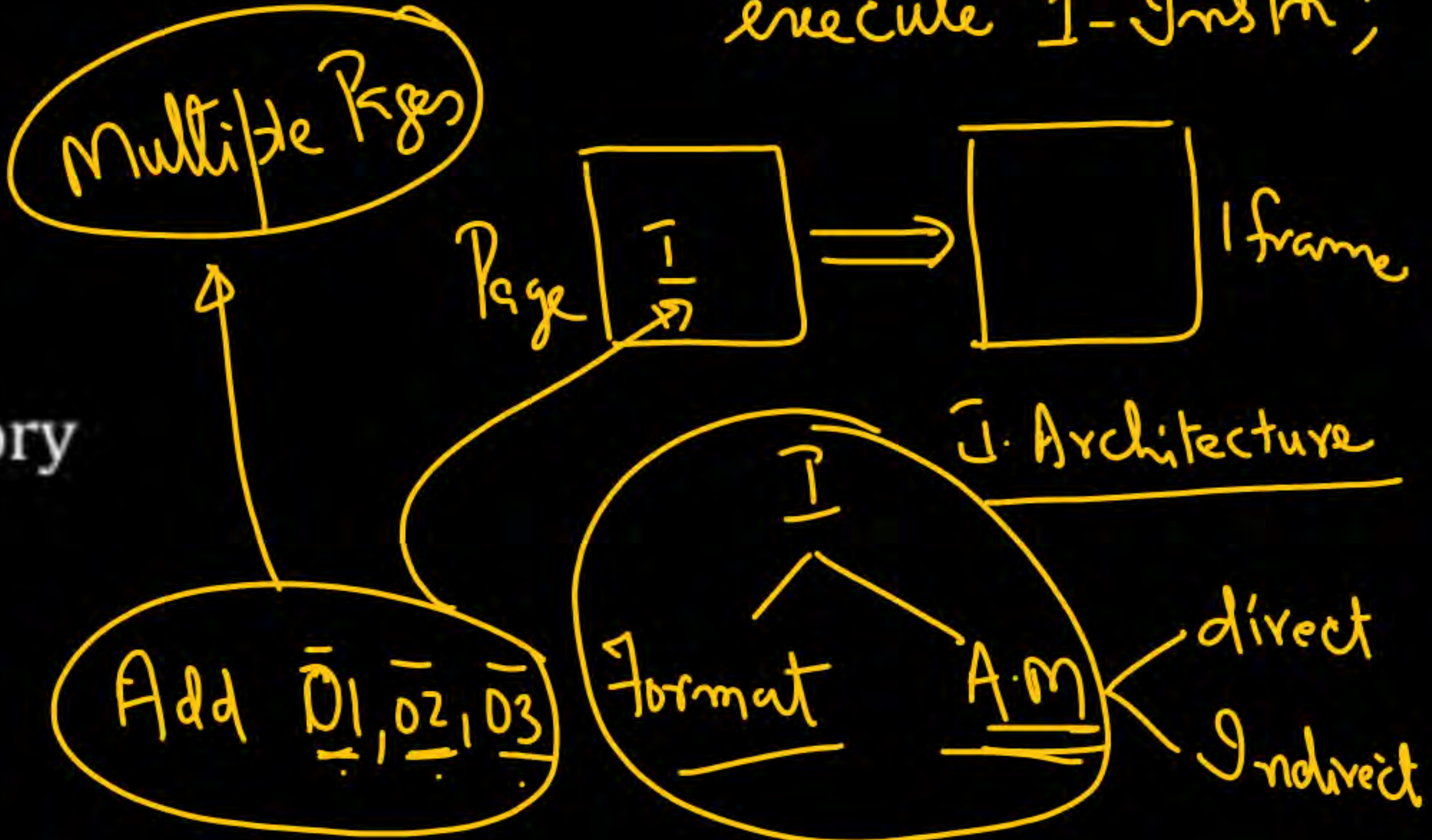
(A) the instruction set architecture ✓

(B) page size

(C) physical memory size

(D) number of processes in memory

Min → Process should be able to succ. execute 1-Instn;



Q.3

Consider a System with V.A.S = P.A.S = 2^{16} Bytes. Page Size is 512 Bytes. The size of Page Table entry is 32 bits. If the Page Table Entry contains besides other information 1 V/I bit, 1 Reference, 1 Modified bit, 3 bits for Page Protection. How many bits can be assigned for storing other attributes of the Page. Also compute Page Table Size in Bytes?

$PAS = VAS = 2^{16}\,B$

$P.S = 512\,B$

$N = M = \frac{2^{16}}{2^{9}} = 2^{7}$

$P = f = 7$ bits

PTE

32 bits

P	f	V/I	R	M	PPr	other
	7	1	1	1	3	x

$13 + x = 32$

$x = 19$ bits

$P.T.S = N * e$

$= 128 \times 4B = 512B$

Q.4

Let the Page Reference and the Delta (▲) be "c c d b c e c e a d" and 4, respectively. The initial Working Set a time t = 0 contains the pages {a, d, e , where 'a' was referenced at time t = 0, 'd' was referenced at time t = -1, and 'e' was referend at time t = -2. Determine the total number of page faults and the average number of page frames used by computing the working set at each reference.

$\Delta = 4$

-2	-1	0	1	2	3	4	5	6	7	8	9	10
e	d	a	c	c	d	b	c	e	c	e	a	d

PF's: 5

t_0: ⟨e d a⟩ : 3 ×

t_1: ⟨e d a c⟩ : 4

t_2: ⟨d a c⟩ : 3

t_3: ⟨a c d⟩ : 3

t_4: ⟨c d b⟩ : 3

t_5: ⟨d b c⟩ : 3

t_6: ⟨d b e c⟩ : 4

t_7: ⟨b e c⟩ : 3

t_8: ⟨c e⟩ = 2

t_9: ⟨c e a⟩ = 3

t_{10}: ⟨c e a d⟩ = 4

Q.5

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:

FIFO

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

Q.6 In the context of operating systems, which of the following statements is/are correct with respect to paging?

MSQ

A. Page size has no impact on internal fragmentation. X

B. Paging helps solve the issue of external fragmentation. ✓

C. Paging incurs memory overheads ✓ (P.T-Size ovhd)

D. Multi-level paging is necessary to support pages of different sizes X

<BC>

THANK YOU!