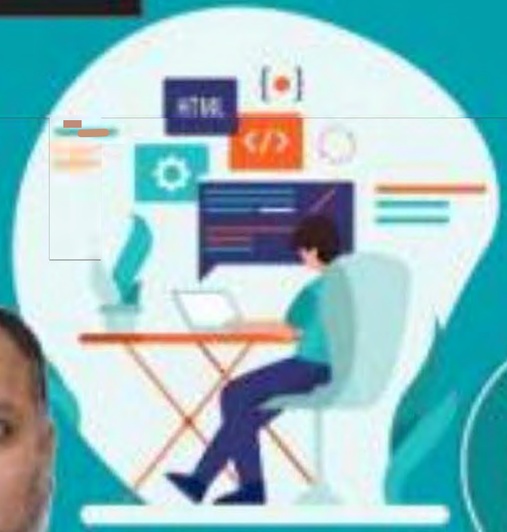
COMPUTER SCIENCE



Memory management

segmentation paging or virtual memory

Lecture No:06

Dr. KHALEEL KHAN SIR

segmentation paging or
virtual memory

Frame Allocation Policies

→ n : processes ($P_1 \ldots P_n$)

→ 's_i' : demand of Process for frames

→ 'D' : Total demand of all Processes ; $D = \sum_{i=1}^{n} s_i$

→ M : Available frames [D >> M]

→ a_i : Frames allocated to Process i ($a_i \leq s_i$)

Ex: n = 5 ; M = 40

50% rule: $a_i = \frac{s_i}{D} \times M$

Pid	s_i	1) Equal Alloc ($\frac{m}{n}$)	2) Proportionate (a_i)
P_1	10	8	$\frac{10}{80} \times 40 = 5$
P_2	5	8	$\frac{5}{80} \times 40 = 3$
P_3	30	8	$\frac{30}{80} \times 40 = 15$
P_4	17	8	
P_5	18	8	
	D = 80		

c) Page Replacement Techniques/Algorithms

Ref. String I : < 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 >

$l = 20$

$n = \{0,1,2,3,4,7\} = 6$

(Pure Demand Paging)

K = 6

'6' =

a	7	2	2	2	4	4	4	0	0	0	7	7	7	
b	0	0	3	3	3	2	2	2	1	1	1	0	0	
c	1	1	1	0	0	0	3	3	3	2	2	2	1	

7 Frames

P	f	V/I	ToL
0	b	1	
1	c	1	
2	–	◎	
7	a	1	

P.T

NO BOA

1) FIFO:

Criteria: ToL

3 Frames: 15 (P: 15/20 = 3/4 = 75%) ✓

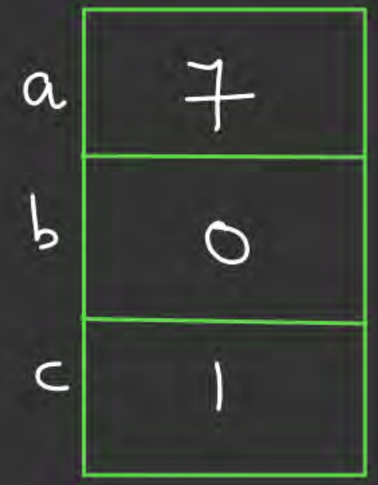
4 Frames: 10 (P: 10/20 = 1/2 = 50%) ↓

PDP < 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 >

2) Optimal Replacement: <In the event of P.F, victimize that Page which will not be used for the longest duration of Time in future>

3F : 9

4F : 8



Limitation of optimal Replacement : is practically Non Implementable

used only as a Benchmark

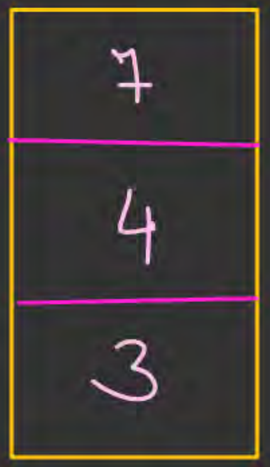
3) Least Recently used (LRU) < victimize the page which has not been referred for the longest duration of time in the past

Ref. string : 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

3F : 12

4F : 8

Criteria : (TOR)



4) Most Recently used (MRU) :

Sel. criteria: TOR

3F : 16

4F : 12 ↓

5) Counting Algorithms

a) L.F.U [Least Frequently used]
b) M.F.U [Most Frequently used]

How many times Page is referred while in Memory.

Conflict : FIFO

Criteria : C.O.R

PDP

3F :

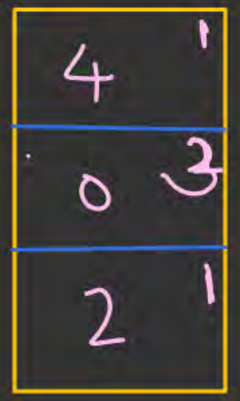
Ref. String: 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

L = 20

n = 6

Man : 20 : L

□ z=1



Implement

1) LRU
2) LRU Approximations
   - NC

1) FIFO
2) LIFO
3) Optimal ~ LRU
4) LRU
5) MRU
6) LFU
7) MFU

Ref. String II: <1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5>

1) FIFO

P.D.P

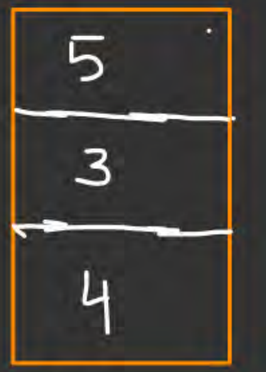
3F → 9

4F → 10

Anamoly (Belady's):

With the Increase in No. of frames, Page fault rate also Sometimes increases;

only FIFO & FIFO Based Algo's

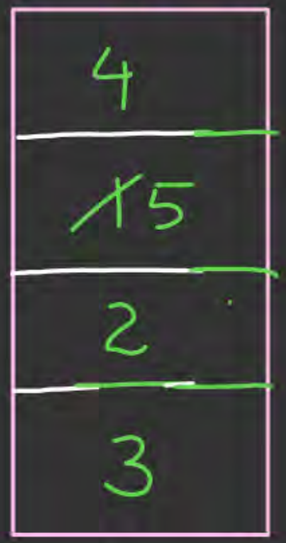


3 Frames

3) LRU: (H/W)

3F:-

4F:-



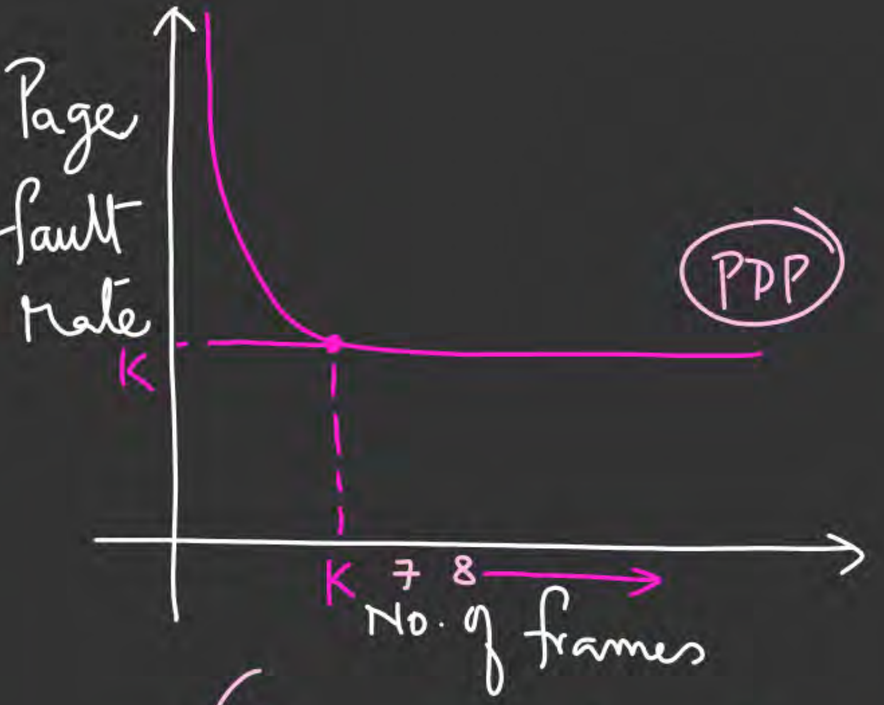
4 frames

2) Optimal Replacement:

NO BA

3F: 7

4F: 6 ↓



→ (In general, with the increase in No. of frames to the Process, Page fault decreases)

Q.8 Which of the following page replacement algorithms suffers from Belady's anomaly?

A. FIFO ✓

B. LRU

C. LIFO

D. Second Chance

Q.9

PYQ
2M

Consider a Process in Demand Page environment having a reference string of length 'L' in which 'K' unique Pages occur. Calculate the lower bound and upper bound of the number of Page Fault for this Process. Assume Process is allocated 'Z' frames.

Ref. String : 'L'

n : 'K'

Frames Alloc : 'Z'

Min : 'K' [Z ≥ K]

Max : 'L' [Z = 1]

Q.10

A Process refers 'n' unique Pages numbered 1 to n is the order and then it refers them back in the reverse order. Process is allocated 'k' Frames. Calculate the number of Page Faults using FIFO replacement with Pure Demand Paging.

'K': Algo

(1 . . . n . . . . 1)

n=10

1,2,3,4,5,6,7,8,9,10,9,8,7,6,5,4,3,2,1

x x x x

(2n)

(2n-K) ✓

Q.11

Consider a System using Demand Paging with Page size of 100 records. Process is allocated one frame and uses pure Demand Paging. The Address sequence generated by the Process is:
0100, 0200, 0430, 0499, 0510, 0530, 0560, 0120, 0220, 0240, 0260, 0320, 0370

(1) What is the Length of Reference String? : 7
(2) Calculate the number of Page Faults? : 7

Ref. String: < 1, 2, 4, 5, 1, 2, 3 >

THANK YOU!