CS & IT ENGINEERING

Operating Systems

Memory Management:

Lecture No. 12

By- Dr. Khaleel Khan Sir

- Page Replacement Techniques
- Problem Solving
- Thrashing

L.R.U Approximations

(i) Reference Bit (R):

(ii) Additional Ref. bits:

(iii) Second chance/clock Algo:

: (T.O.L + R)

FIFO (when all R-values are 1's)

< Belady's Anomaly >

(iv) Enhanced Second chance

N.R.U (Not Recently used):

Criteria: R+M (modified bit / dirty bit)

	R M	
I:	0 0	→ Not Referred & Not Modified
II:	0 1	→ Not Ref'd but Modified
III:	1 0	→ Referred & clean
IV:	1 1	→ Referred & Modified

PTE Structure

P	f	VI	T.o.L	R	M	
0	c	1	4	1	1	
1	a	1	3	0	0	
2	—	0	—	—	—	
3	b	1	0	1	1	
4	d	1	2	1	0	
5	e	1	1	0	1	

Page-Table

(i) FIFO : P_3

(ii) R : P_1

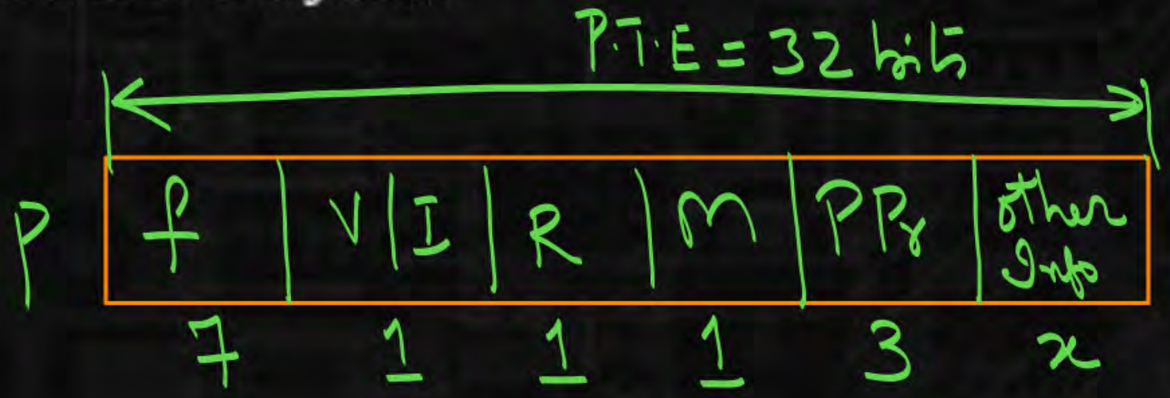
(iii) Sec. chance : P_5

(iv) Enhanced Second chance : P_1

Q.

*

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?



$M_{frames} = \frac{2^{16}}{2^9} = 2^7$

$\therefore x = 19$ bits ✓

P.T.S = N * e

$= 2^7 \times 2^2 B = 2^9 B = 512B$ ✓

Q. 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

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 Anomaly)

D. Never affect the number of page faults.

Q. A memory page containing a heavily used variable that was initialized very early and is in constant use is removed when

(brought first)

A. LRU page replacement algorithm is used

B. FIFO page replacement algorithm is used

C. LIFO page replacement algorithm is used

D. None of the above

Q.

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:

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

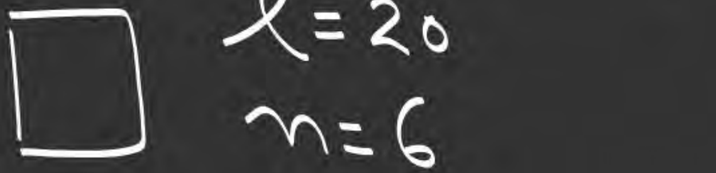
B. S1 is true, S2 is false

C. S1 is false, S2 is true

D. S1 is false, S2 is false

Q) Consider a Process having ref string of length 'L' in which n-unique Pages occur; 'Z' Frames are allocated to the Process; Calculate the Lower Bound (Min) & Upper Bound (Max) of the no. of Page faults;

Ref String: 7, 0, 1, 2, 0, 3, - - - -



$L = 20$

$n = 6$

a) Max() : 'l' [Z = 1]
U.B

b) Min() : 'n' [Z ≥ n] (Pure D.P)
L.B
: 0 (Pre-fetch D.P)

* THRASHING : Excessive/High Paging activity (High Page fault rate)

↓

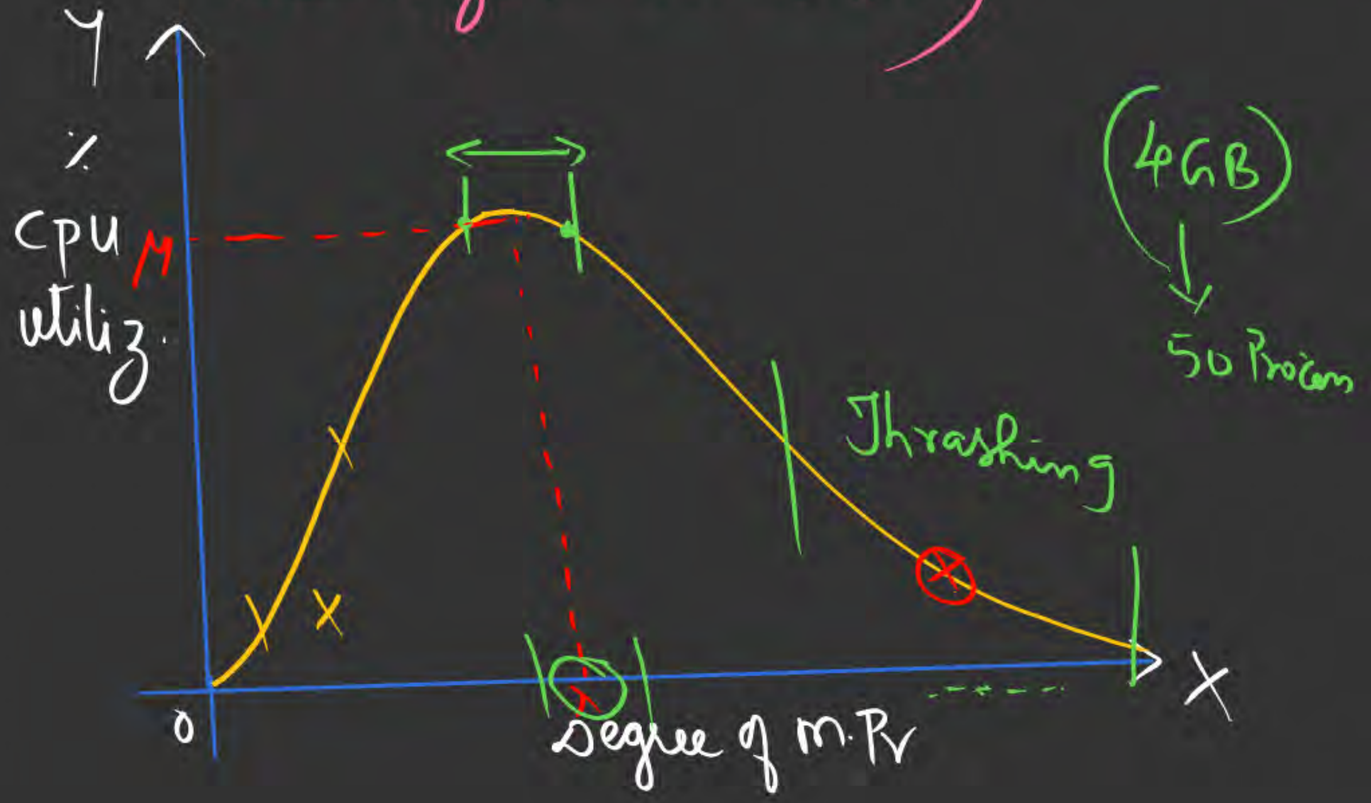
Like deadlock is also an undesirable feature of O.S

(↓ The act of causing a Page-fault leading to Loading & Saving the Pages on disk)

Reasons for Thrashing:

✓ 1. High degree of M.Pr

✓ 2. frame alloc declines <Lack of Memory>



Other Reasons:

1) Page Replacement Policy

2) Page-Size:

large: (Less pages) → Less P. faults

(*) 3) (Programming Techniques & Data Structures) **

Thrashing Control Strategies

- Prevention
  - <Never occurs>
  - (Controlling degree of M.Pr)
  - ↓
  - (Long Term Scheduler)
- ∞ etection & Recovery
  - Process Suspension
  - ⇒ MTS
  - <occurs>
  - (Low cpu utiliz
  - → Max # of Processes getting Blocked
  - → High degree of M.Pr
  - → High disk utiliz.)

Case 1: integer A[1..128][1..128]; (PS = 128 W); R.M.O (i) PS = 256 W (ii) PS = 64

*

$(128) * (128) = 2^{14} = 16K$

Pure D.P

FIFO

1. for i ← 1 to 128
   for j ← 1 to 128
   A[j, i] = 1;

C.M.O

Locality of Reference

2. for i ← 1 to 128 (128 * 1)
   for j ← 1 to 128
   A[i, j] = 1;

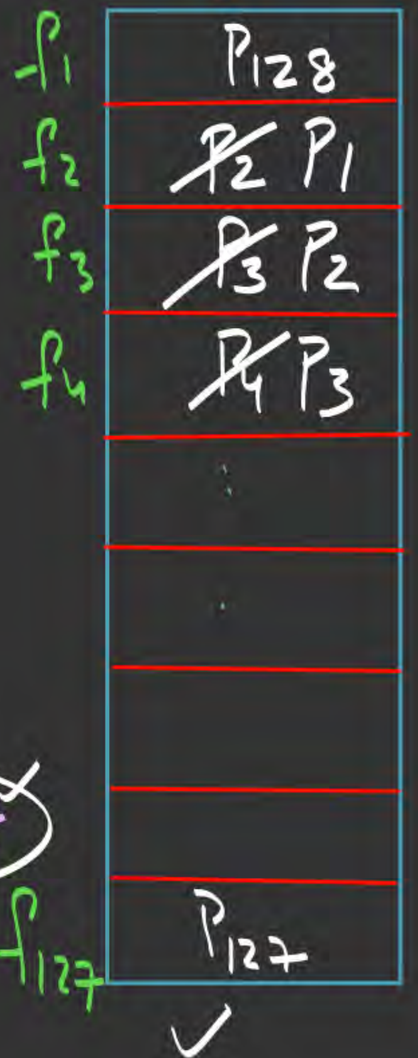
R.M.O

A) 128, 1

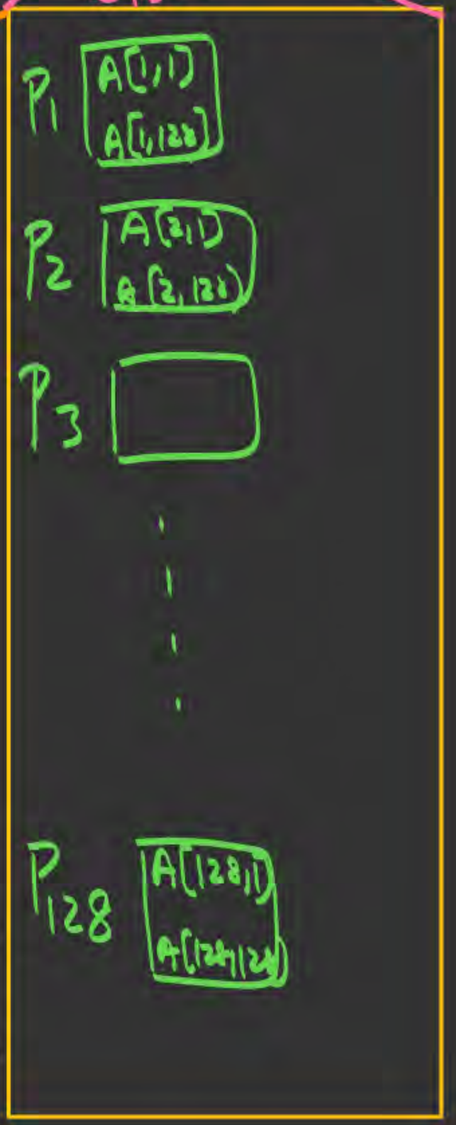
B) 1, 128

C) 128 * 128, 128

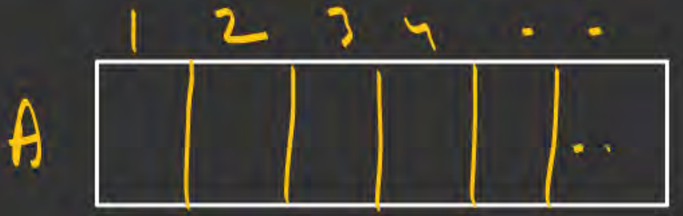
D) 128, 128



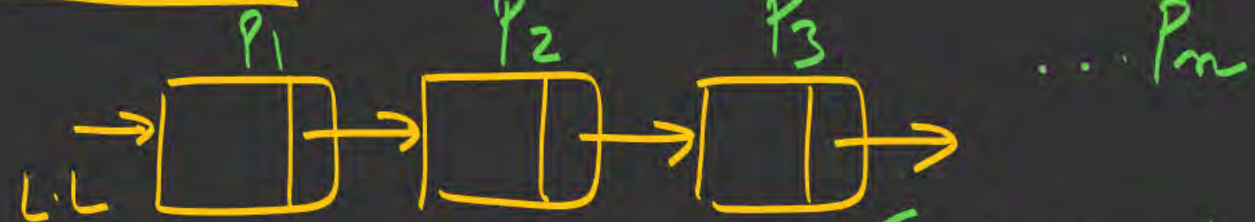
1 frame



Case II: Arrays vs Linked list



Contiguous



(n' - P.F.s)

Which is better for O.P. Environment

(Arrays) → Less P.F's

Page

Case III: Linear Search vs Binary Search

(May gen. Less P.F.s) O(n)

(Locality)

O(Log n)

(A. Programming Tech. of a O.S is said to be good in Demand Page environment, iff it satisfies Locality Model)

Working Set Strategy (Model): (To Minimize Page fault rate & also utilize Memory effectively)

(Principle of Locality of Reference)

10 Pages
10KB
main()
{
f();
}

2 Pages
2KB
f()
{
g();
}

2
28KB
g()
{
h();
}

15KB
h()
{
Scanf();
10KB
}

Prog-Size = 55KB

Page Size = 1KB

Npages = 55

(Static)

50% rule

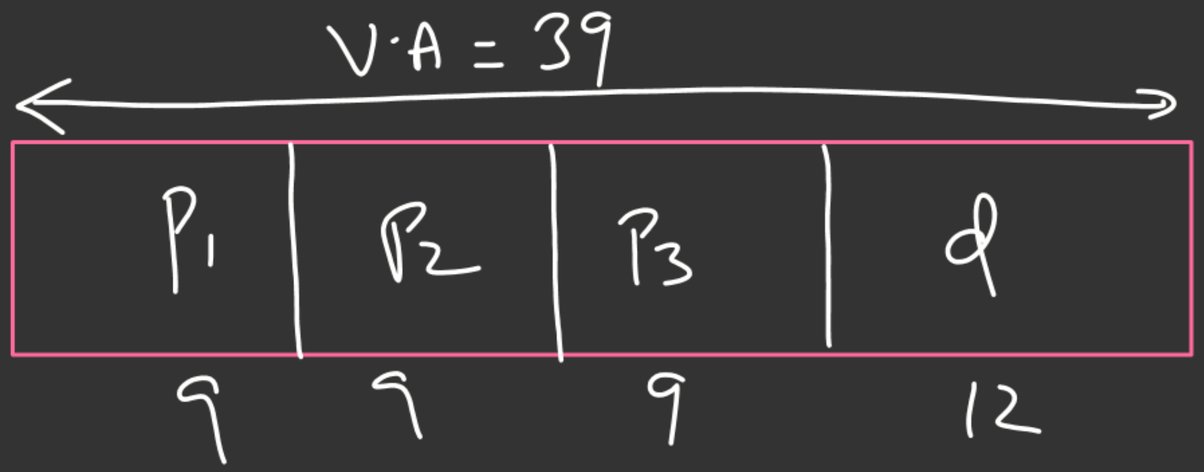
26 wastage

The basic Idea of working Set model is to estimate the Size of Locality in which the program is executing and ask for only those many frames;

W.S.S follows dynamic type of Mem. Alloc. (Alloc. of frames)

(Implementation)

< 1 opt + 2 pages of Next level P.T
Chunks
+ 1024 Chunks of 1st level) * 4KB



V.A.S = 2GB

PAS = 2GB

$\frac{2GB}{P.S} = N$

THANK YOU!