

Aim : Implementing page replacement algo.  
LRU optimal

Problem To write java prog to implement LRU  
Statement: and optimal algo for page replacement

Theory :

- ① whenever there is a page reference for which the page is not in memory that event is called page fault. On page fetch on page failure situation.
- In such case we have to make space in memory for this new page by replacing the existing page.
- But we can't replace any page.
- There are two algorithms.

### LRU Page replacement

- The main diff bet<sup>n</sup> FIFO and optimal page replacement is that the FIFO algorithm uses time when the page was brought into memory and the optimal algorithm uses the time when page is to be used.
- If we use LRU recent part as an approximate of the future then we will replace the page that has not been used for the longest period of time.
- This approach called as least recently used algorithm.
- LRU chooses that has not been used for the longest period of time.
- This approach called as least recently used algorithm.

Now consider reference string 7, 0, 1, 2, 0, 3, 4, 2, 3, 0, with three memory frames as block.

- The first three reference case page fault that fill the memo empty frame

1	2	3	4	5	6	7	8	9	10	11	12
7	0	1	2	0	3	0	4	2			

1	2	3	4	5	6	7	8	9	10	11	12
7	0	1	2	0	3	0	4	2	3	0	
7	7	7	2	2	2	2	4	4	4	0	
0	0	0	0	0	0	0	0	0	0	3	3
1	1	1	3	2	3	2	2	2	2	2	2
+	+	+	+	+	+	+	+	+			+

### ③ optimal page replacement algo

- This algo has the lowest page fault rate of all algorithm
- This algorithm states that replace the page which will not be used for largest period of time i.e. future knowledge long. of reference string is req.
- often called as Balady's Hen Boy's idea!
- Impossible to implement

consider 0 2 1 6 4 0 1 0 3 1 2 1

0	2	1	6	4	0	1	0	3	1	2	1
0	2	1	0	2	3	0	1	1	4		



Q) Algorithm for LRU :-

- 1) Start traversing the Page
  - i) If Set holds less Pages than Capacity
    - a) Insert Page into set 1 by 1 until the size of Set reaches capacity.
    - b) simultaneously maintain the recent occurred index of each page in map called indices.
  - ii) else :- if current Page present in set do nothing else :-
    - a) find the Pg in that was least recently used we find it using index array
    - b) Replace the found Page with current
    - c) Implement Pg Fault
    - d) update index wi of current Page
- 2) Return Page Fault

Q) Algorithm for optimal

- 1) Start Process
- 2) Declare the size & get num of Page inserted
- 3) get value
- 4) compare counter label & stack
- 5) select the optimal Page by counter value
- 6) stack them according the selection
- 7) Print Page with fault Pages
- 8) Stop Process

Conclusion :-

Thus we have studied LRU and optimal Page replacement algorithm.