TITLE : Implementation of Page replacement algorithms.

PROBLEM STATEMENT:
      Basic functionalities of OS.
      Memory management in OS.

OBJECTIVES:

Understand virtual memory management

Analyze the need of page replacement algorithms

Compare various page replacement algorithms

OUTCOMES:

The students will be able to

- Implement various page replacement algorithms like FIFO, LRU and Optimal
- Compare the page replacement algorithms based on hit ratio

THEORY:
    Whenever there is a page reference for which the page needed in not present memory, that event is called page fault or page fetch or page failure situation. In such case we have to make space in memory for this new page by replacing any existing page. But we cannot replace any page. We have to replace a page which is not used currently. There are some algorithms based on them. We can select appropriate page replacement policy. Designing appropriate algorithms to solve this problem is an important task because disk I/O is expensive.
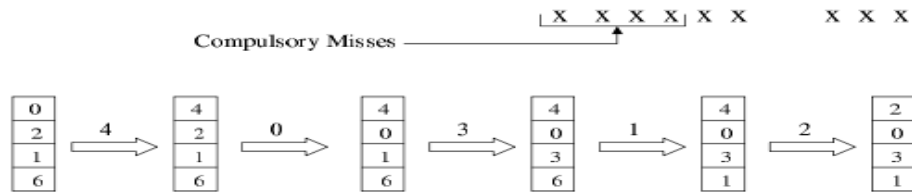
First in First out (FIFO)

The oldest page in the physical memory is the one selected for replacement. Keep a list On a page fault, the page at the head is removed and the new page added to the tail of the list
Example -

FIFO

•Consider the following reference string: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1
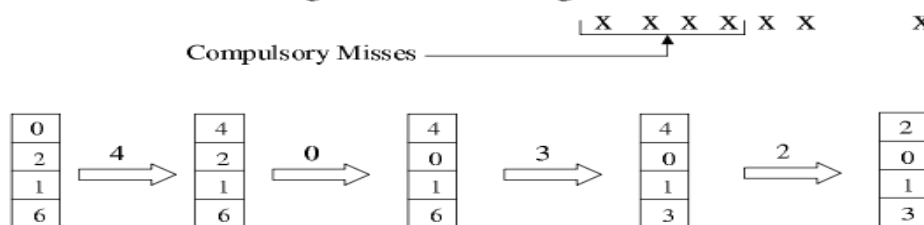
Compulsory Misses ⟶ | X   X   X   X | X   X   X   X   X

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

•Fault Rate = 9 / 12 = 0.75

LRU(Least Recently Used): In this algorithm, the page that has not been used for longest period of time is selected for replacement. Although LRU is theoretically realizable, it is not cheap. To fully implement LRU, it is necessary to maintain a linked list of all pages in memory, with the most recently used page at the front and the least recently used page at the rear. The difficulty is that the list must be updated on every memory reference. Finding a page in the list, deleting it, and then moving it to the front is a very time consuming operation, even in hardware (assuming that such hardware could be built).

LRU

•Consider the following reference string: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Compulsory Misses ⟶ | X   X   X   X | X   X   X   X

| 0 | | 4 | | 4 | | 4 | | 2 |
2	4 ⟹	2	0 ⟹	0	3 ⟹	0	2 ⟹	0
1		1		1		1		1
6		6		6		3		3

•Fault Rate = 8 / 12 = 0.67

The Optimal Page Replacement Algorithm:
The algorithm has lowest page fault rate of all algorithm. This algorithm state that: Replace the page which will not be used for longest period of time i.e future knowledge of reference string is required. Often called Balady's Min Basic idea: Replace the page that will not be referenced for the longest time.

Optimal Page Replacement

•Consider the following reference string: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1

Compulsory Misses ⎯⎯⎯⎯⎯⎯⎯⎯→ ⌐X X X X⌐ X   X

0
2
1
6

4 ⟹

0
2
1
4

3 ⟹

3
2
1
4
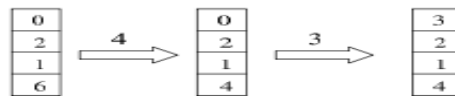
•Fault Rate = 6 / 12 = 0.50
•With the above reference string, this is the best we can hope to do

Steps to do /algorithm:

1. Create s menu to select various page replacement algorithms
2. Take no of page frames and pages along with reference strings.
3. Calculate the number of page faults.
4. Perform a comparative assessment of best policy for given reference string.

CONCLUSION : We have successfully implemented page replacement algorithms.