ASSIGNMENT D1

TITLE :Implementation of Page replacement algorithms.

Problem Statement:

Write a Java Program (Using OOP features) to implement paging simulation using

1) FIFO
2) Least Recently Used (LRU)
3) Optimal Algorithms

Objectives:

- Understand virtual memory management
- Analyze the need of page replacement algorithms
- Compare various page replacement algorithms

Outcomes:

I will be able to

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

Software and Hardware Requirements:

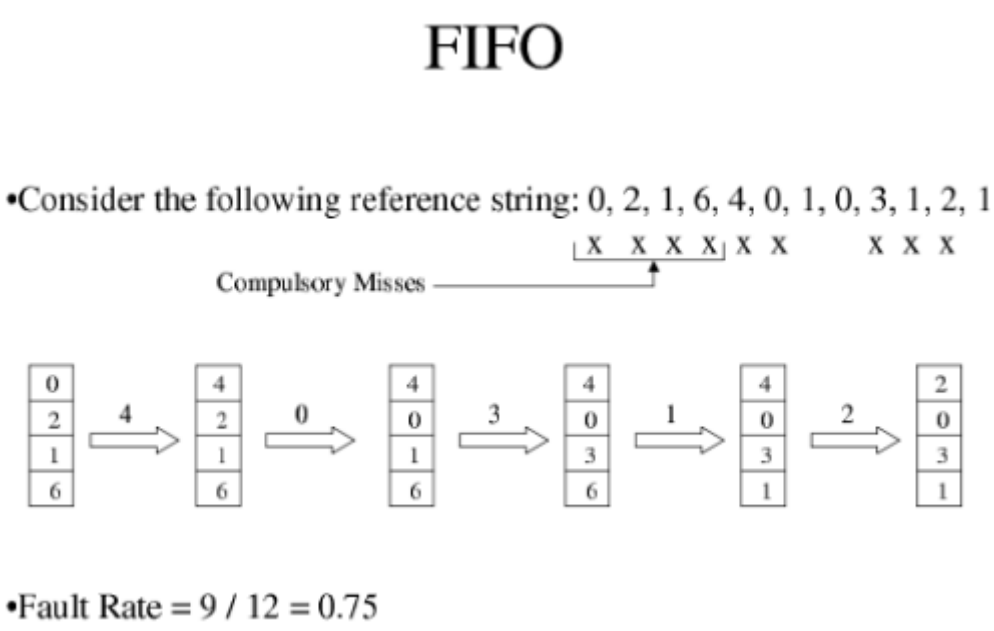
- Working PC.
- 64 bit Fedora OS
- Eclipse IDE and JAVA
- I3 processor

Theory

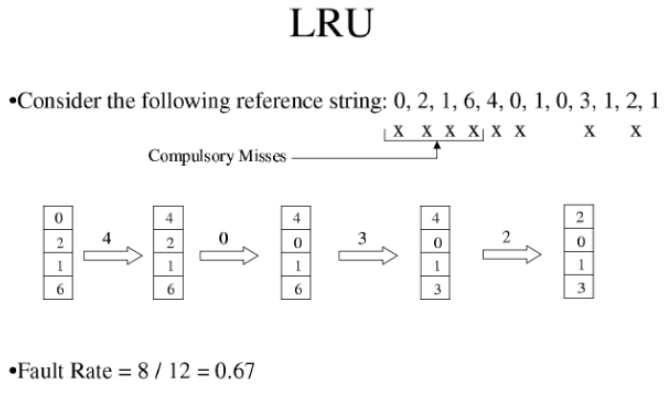
Whenever there is a page reference for which the page needed in not present memory, that event is called page fault or page fetch or page failure situation. In such case we have to make space in memory for this new page by replacing any existing page. But we cannot replace any page. We have to replace 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

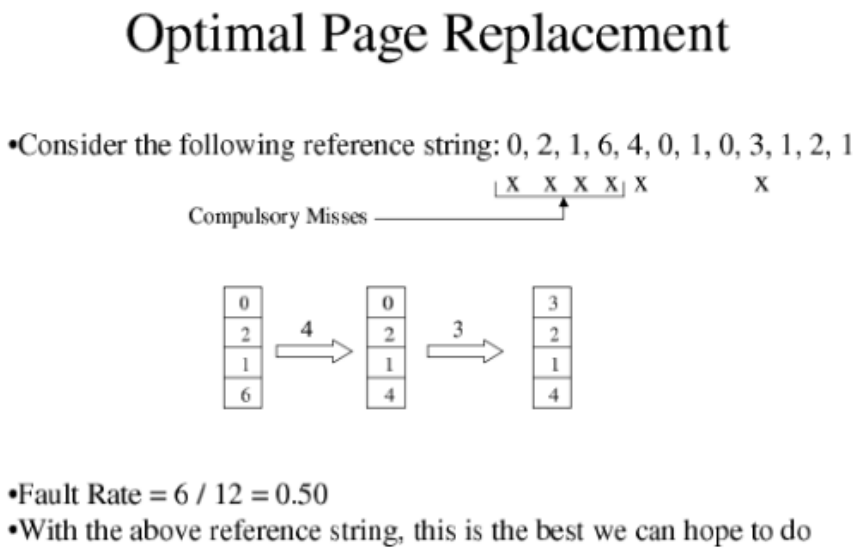


LRU(Least Recently Used): In this algorithm, the page that has not been used for longest period of time is selected for replacement. Although LRU is theoretically realizable, it is not cheap. To fully implement LRU, it is necessary to maintain a linked list of all pages in memory, with the most recently 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).



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.



TEST CASES:

Reference string : 0 2 1 6 4 0 10 3 1 2 1

Number of frames: 4

Algorithm	Page Fault	Page Fault Ratio
LRU	8	0.67
Optimal	6	0.50

Reference string :3 2 1 0 3 2 4 3 2 1 0 4

Number of frames: 3

Algorithm	Page Fault	Page Fault Ratio
LRU	10	0.83
Optimal	7	0.58

ANALYSIS:

Page Faults are in the following order:

FIFO $>$ LRU $>$ Optimal

Optimal uses the future reference of a page and then replaces it.

LRU replaces pages with last referred ones.

Conclusion:

The page replacement algorithms like LRU and Optimal have been successfully implemented and compared with one another.