IMPLEMENT FISRT IN FIRST OUT PAGE REPLACEMENT ALGORITHM?

**Introduction**

In the memory management system, the replacement of the pages is a main concept. when the kernel creates process with the system pages are required to be in the main memory. These pages placed in the main memory, we will face with a situation that is called page fault. When a page fault occurs, the process needs to find the page that caused the page fault in disk. If there is a free space in main memory it will put in this area. If there isn’t free space it is necessary to make a page replacement with the algorithm that is implemented. After that the page that is removed will be written in disk. Then the process can continue the job, because the required page is in the main memory and the process can read it.

There are many page replacement algorithms such as: the optimal, the not recently used (NRU), The not frequently used (NFU), The first in first out(FIFO) The Second Chance, The Clock, The Least Recently Used (LRU),

**The First in first out Algorithm(FIFO)**

What is first in first out(FIFO)?

This is a method of managing data were the first element to be added to the queue is the first element to be removed from the queue. This is similar to a real-world queue, like waiting in line for registration in a university set up were the first person to join the will be the first person to be attended to in memory concept first in first out is a page replacement algorithm that manages the pages in memory by maintaining the order in which they were loaded. The first page that was loaded into memory is the first one to be replaced when a new page needs to be loaded.

IMPLEMENTATION

FIFO can be implemented using a queue data structure, when a page is added it is put in a queue in the main memory and when a new page is requested for the page in front of the queue is dequeued. The following are the steps:

1. **Initialization**

Define the capacity of the page frame.

Create an empty list to hold the pages currently in memory.

Initialize a counter for page faults.

1. **Page Reference Handling**

For each page reference string;

Check if the page is already in memory.

If the page is not in memory

If the memory is full, remove the oldest page (the first page that was added).

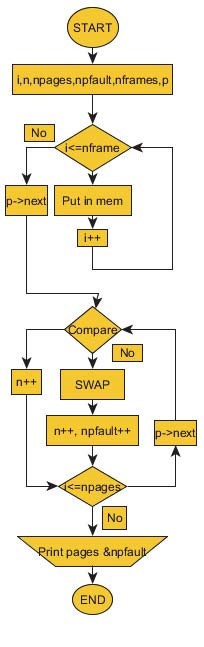
Add the new page to memory.

Increment the page fault counter.

3.**Output Results**

After processing the reference string, output the total number of page faults and the current pages in memory

Below is a FIFO flow chart

s

**What the symbols in the flow chart mean.**

 **npages** – length of the page reference string (how many page requests)

 **nframes** – number of physical frames available.

 **i** – how many frames are currently filled.

 **n** – index of the current request in the reference string.

 **npfault** – number of page faults counted so far.

 **p** – a pointer to the current page request (moves to the next request with p -> next).

References and citations

G. Rexha, E. Elmazi, I. Tafa, “A comparison of three page replacement algorithms: FIFO, LRU and Optimal,” **Academic Journal of Interdisciplinary Studies**, 2015.

A. Silberschatz, P. B. Galvin, G. Gagne, **Operating System Concepts** (various editions). (Standard textbook source for paging, page faults, and FIFO behavior; widely cited on Scholar.)