**National University of Computer & Emerging Sciences (FAST-NU)**

**“Page Replacement Algorithms”**

**Submitted to:**

Sir Ghufran Ahmed (OS Instructor)   
Muhammad Mubassher Ansari (TA of OS)

**By:**

**K190326 Om Kumar  
K191320 Vinay Kumar   
K191309 Darpan Kumar**

**Submission Date:**

**[16-12-2021]**

**Page Replacement Algorithms**

In this project we will be show how different page replacement algorithm works and will show a general comparison between them. The algorithms which we will be comparing are as under:

* First In First Out. (FIFO)
* Least Recently Used. (LRU)
* Second Chance Algorithm. (SCA)

1. **DESCRIPTION:**

FIRST IN FIRST OUT:The simplest page-replacement algorithm is a FIFO algorithm. The first-in, first-out (FIFO) page replacement algorithm is a low-overhead algorithm that requires little bookkeeping on the part of the operating system. The idea is obvious from the name – the operating system keeps track of all the pages in memory in a queue, with the most recent arrival at the back, and the oldest arrival in front. When a page needs to be replaced, the page at the front of the queue (the oldest page) is selected. While FIFO is cheap and intuitive, it performs poorly in practical application. Thus, it is rarely used in its unmodified form. This algorithm experiences Bélády's anomaly. In simple words, on a page fault, the frame that has been in memory the longest is replaced.

LEAST RECENTLY USED: The least recently used (LRU) page replacement algorithm, though similar in name to NRU, differs in the fact that LRU keeps track of page usage over a short period of time, while NRU just looks at the usage in the last clock interval. LRU works on the idea that pages that have been most heavily used in the past few instructions are most likely to be used heavily in the next few instructions too. While LRU can provide near-optimal performance in theory (almost as good as adaptive replacement cache), it is rather expensive to implement in practice. There are a few implementation methods for this algorithm that try to reduce the cost yet keep as much of the performance as possible.

SECOND CHANCE ALGORITHM: A modified form of the FIFO page replacement algorithm, known as the Second-chance page replacement algorithm, fares relatively better than FIFO at little cost for the improvement. It works by looking at the front of the queue as FIFO does, but instead of immediately paging out that page, it checks to see if its referenced bit is set. If it is not set, the page is swapped out. Otherwise, the referenced bit is cleared, the page is inserted at the back of the queue (as if it were a new page) and this process is repeated. This can also be thought of as a circular queue. If all the pages have their referenced bit set, on the second encounter of the first page in the list, that page will be swapped out, as it now has its referenced bit cleared. If all the pages have their reference bit cleared, then second chance algorithm degenerates into pure FIFO.

1. **HOW WE STARTED:**

We started with the idea of a topic of memory utilization and allocation in an optimized pattern. Then, we learnt about 3 different ways our computer takes memory from hard disk to main memory. So, we decided to try and make it into an algorithm to automate and present the process in a simpler manner which is easy to understand for new students of this course.

1. **PROBLEMS FACED:**

During the making of our project we also wanted to include the optimal page algorithm but in practical use there is no implementation of it. So we decided to go with “Second chance algorithm”.

1. **THE ACTUAL WORKING OF THE PROJECT (Methodology):**

In the three processes we explore in this project, we allocated the data by either manually adding the data into the algorithm or choosing an automatic selection algorithm which selects values for the user and runs it into the algorithms of: FIFO, LRU and SCA.

1. **EXPLANATION OF SOURCE CODE:**

* First, we included all necessaries libraries ARE included.
* There are 3 parts of this project
  1. UI
  2. Backend working data structures

Queue Class

Stack Class

Node Class

BitArray Class

* 1. Main working of all page replacement algorithms.

FIFO

LRU

SCA

1. **PROJECT ASPECTS in CURRENT/FUTURE TECHNOLOGY:**

We use this technique in operating system to managing the main memory efficiently.

1. **Project configurations and Code**

****

1. **Conclusion**

In conclusion we have found that there can be multiple algorithms which can be used in memory allocation to load up memory from hard disk or any other resource to be used in a quick and efficient manner so the user can have the best experience.

**REFERENCES:**

* <https://www.geeksforgeeks.org/page-replacement-algorithms-in-operating-systems/>
* <https://en.wikipedia.org/wiki/Page_replacement_algorithm#:~:text=In%20a%20computer%20operating%20system,memory%20needs%20to%20be%20allocated.>