International Journal of Engineering Research and Advanced Technology (IJERAT)

DOI: 10.31695/IJERAT.2022.8.8.1

E-ISSN : 2454-6135

Volume.8, No. 8
August - 2022

Study and Comparison of Replacement Algorithms

Safana Hyder Abbas[1], Wedad Abdul Khuder Naser[1], & Lamyaa Mohammed Kadhim[2]

[1,2]Computer Science Department, University of Al-Mustansirihya, Baghdad, Iraq

[3]Collage of Dentistry, University of Al-Mustansirihya, Baghdad, Iraq

ABSTRACT

When a page fault occurs, the operating system has to choose a page to remove from memory to make room for the page that has to be brought in. The page replacement is done by swapping the required pages from backup storage to main memory and vice-versa. When a new page needs memory for allocation, page replacement algorithms decide which pages to swap out of the memory. In this paper, a comparison and analysis of five replacement algorithms: First in First out (FIFO), Least Recently Used (LRU), Clock, Most Recently Used algorithm (MRU), and Optimal Page Replacement (OPT) are made.

Keywords: Page Replacement, Page fault, Replacement Algorithm, Clock.

1. INTRODUCTION

Once the cache has been filled, fetching a new block into the cache requires the replacement of the existing blocks. For direct mapping, there is only one possible location for any particular memory block, and no choice is possible. For the fully associative and set-associative techniques, an algorithm for replacement is needed [1].

Because a user could occasionally not need the full process in the memory, virtual memory does not require the entire process to be in the memory before we can run. The user may only be interested in a few pages of the procedure at any one moment rather than the complete process. Bring only the pages from the disk that are needed for execution into memory in this situation. The process's pages are never loaded into the memory if they are not required. Demand paging is the name of this method.
Page faults occur when a process tries to access a memory page that hasn't been brought in from the disk. We will utilize the valid/invalid bit method to determine the page fault. If the bit for one of the pages is set to be valid, then that page is present in the memory. If not, the page is invalid and is now stored on the disk rather than in memory. The page fault may happen when the required page is not in memory or when the memory is full but the wanted page is currently on the disk and ready to be brought into memory. Bring the requested page from the disk into memory at this point, or use one of the page replacement techniques to replace part of the pages that are already in memory, to enable the execution of the desired page. The most effective page replacement method with the lowest page fault rate is optimum page replacement [2].

There are several methods for performing multitasking, one of which is the paging approach, which divides physical memory into identically sized blocks known as frames and also known as pages. The results must be saved in the memory that the OS uses while an operation is being processed. The pages are transferred to the main memory during this process, and the main memory may handle many operations at once thanks to virtual memory methods. Page replacement techniques are used in this operation to transfer the necessary pages [3].

The purpose of page replacement is to choose the best page to remove from main memory in the event of a page fault and replace it with a fresh page from disk that contains the desired datum or instruction because main memory has a finite size and is significantly smaller than permanent storage. When a page of memory has to be allocated, the operating system uses page replacement algorithms to select which memory pages to swap out and write to disk. Paging takes place if a page fault occurs and a free page cannot be utilized for allocation purposes due to the lack of available pages or the fact that there are fewer free pages than needed. A read-in from disk is necessary for I/O completion when a page that was chosen for replacement and paged out is referenced again. The duration of waiting for page-ins is what determines the algorithm's quality; the shorter the duration, the better. A page replacement algorithm attempts to choose which pages should be replaced to reduce the overall number of page misses while balancing it with the costs of primary storage and the algorithm's own processor time. It does this by examining the limited information about accessing the pages provided by hardware. Page replacement algorithms come in a variety of forms [4].

2. ALGORITHMS DESCRIPTION

In this section five replacement algorithms (FIFO, LRU, OPTIMAL, MRU and clock) are illustrated.

2.1 OPTIMAL

The optimal replacement policy (OPT), which is also called (offline algorithm), replaces the block that will be needed farthest in the future. This replacement policy is not pragmatic, but it is frequently used for comparability functions to ascertain how effective other replacement policies are in the best possible. In other words, the optimal replacement policy is seen only after a program has already been implemented [5-6].

2.2 FIRST IN FIRST OUT (FIFO)

The easiest method of replacing the page is to use the First in First out (FIFO) page replacement algorithm. The goal is to replace the page that has been in main memory for the longest amount of time, or the oldest page among all the pages. All of the pages in main memory can be stored in a FIFO queue. The FIFO page replacement mechanism has the benefit of being simple to implement, but it also has the drawback of being vulnerable to Belady's oddity. A surprising outcome of the FIFO page replacement method is the Belady anomaly. The page fault rate in several of the reference strings increases as memory size grows [7].

2.3 LEAST RECENTLY USED (LRU)

The memory page that hasn't been utilized for the longest amount of time is replaced using the Least Recently Used (LRU) Page Replacement Algorithm. The benefit of LRU page replacement method is that it does not suffer from Belady's anomaly and the drawback is that it demands costly [8].

2.4 CLOCK

All page frames are represented in CLOCK as being ordered in the shape of a clock via a circular list. The oldest page in the buffer is shown by the clock hand. Every time a certain page is referred, its corresponding reference bit is set. When a page is missed, the oldest page, the one indicated by the hand, is scrutinized before the page replacement policy is applied. The hand is advanced to point to the following oldest page if the page's reference bit is set, after which the bit is reset to zero. This procedure is repeated until a page with reference bit zero is discovered. The page that was so discovered is deleted from the buffer and replaced with a new page that has the reference bit set to 0. This CLOCK method has been shown to approximate LRU quite effectively and with little overhead [9].

2.5 Most Recently Used algorithm (MRU)

The **Most Recently Used (MRU)** algorithm is also called **Fetch-and-Discard** replacement algorithm. This algorithm is used to deal with the case such as sequential scanning access pattern, where most of recently accessed pages are not reused in the near future [10].

3. ANALYSIS

Let us consider the CPU request the following sequence of 10 pages reference string 1,4,1,5,3,1,5,4,,2,5 for memory frames of three and four pages , the page trace will be given to the five replacement algorithms and the page faults are estimated:

FIFO

With page frame=3

1	**4**	**1**	**5**	**3**	**1**	**5**	**4**	**2**	**5**
1	1	1	1	3	3	3	3	2	2
	4	4	4	4	1	1	1	1	5
			5	5	5	5	4	4	4
F	F		F	F	F		F	F	F

Number of page faults (F) = 8

With page frame=4

DOI : 10.31695/IJERAT.2022.8.8.1

1	4	1	5	3	1	5	4	2	5
1	1	1	1	1	1	1	1	2	2
	4	4	4	4	4	4	4	4	4
			5	5	5	5	5	5	5
				3	3	3	3	3	3
F	F		F	F				F	

Number of page faults (F) = 5

LRU

With page frame=3

1	4	1	5	3	1	5	4	2	5
1	4	1	5	3	1	5	4	2	5
	1	4	1	5	3	1	5	4	2
			4	1	5	3	1	5	4
F	F		F	F			F	F	

Number of page faults (F) = 6

With page frame=4

1	4	1	5	3	1	5	4	2	5
1	4	1	5	3	1	5	4	2	5
	1	4	4	5	3	1	5	4	2
			1	4	5	3	1	5	4
				1	4	4	3	1	1
F	F		F	F				F	

Number of page faults (F) = 5

CLOCK

With page frame=3

1	4	1	5	3	1	5	4	2	5
1	1	1	1	3	3	3	3	2	2
	4	4	4	4	1	1	1	1	5
			5	5	5	5	4	4	4
F	F		F	F	F		F	F	F

Number of page faults (F) = 8

With page frame=4

1	4	1	5	3	1	5	4	2	5
1	1	1	1	1	1	1	1	2	2
	4	4	4	4	4	4	4	4	4
			5	5	5	5	5	5	5
				3	3	3	3	2	2
F	F		F	F					

Number of page faults (F) = 4

MRU

DOI : 10.31695/IJERAT.2022.8.8.1

With page frame=3

1	4	1	5	3	1	5	4	2	5
1	1	1	1	1	1	5	5	5	5
	4	4	4	4	4	4	4	2	2
			5	3	3	3	3	3	3
F	F		F	F		F		F	

Number of page faults (F) = 6

With page frame=3

1	4	1	5	3	1	5	4	2	5
1	1	1	1	1	1	1	1	1	1
	4	4	4	4	4	4	4	2	2
			5	5	5	5	5	5	5
				3	3	3	3	3	3
F	F		F	F				F	

Number of page faults (F) = 5

OPTIMAL

With page frame=3

1	4	1	5	3	1	5	4	2	5
1	1	1	1	1	1	1	4	4	4
	4	4	4	3	3	3	3	2	2
			5	5	5	5	5	5	5
F	F		F	F			F	F	

Number of page faults (F) = 6

With page frame=4

1	4	1	5	3	1	5	4	2	5
1	1	1	1	1	1	1	1	2	2
	4	4	4	4	4	4	4	4	4
			5	5	5	5	5	5	5
				3	3	3	3	3	3
F	F		F	F				F	

Number of page faults (F) = 5

Table(1) illustrates the amount of page errors with page frame sizes 3 and 4 for each algorithm for the previous suggested sequence of pages as shown below:

Table 1. Comparison of page faults for the five replacement algorithms

algorithm	FIFO	LRU	CLOCK	MRU	OPTIMAL
Page frame=3	8	6	8	6	6
Page frame=4	5	5	4	5	5

DOI : 10.31695/IJERAT.2022.8.8.1

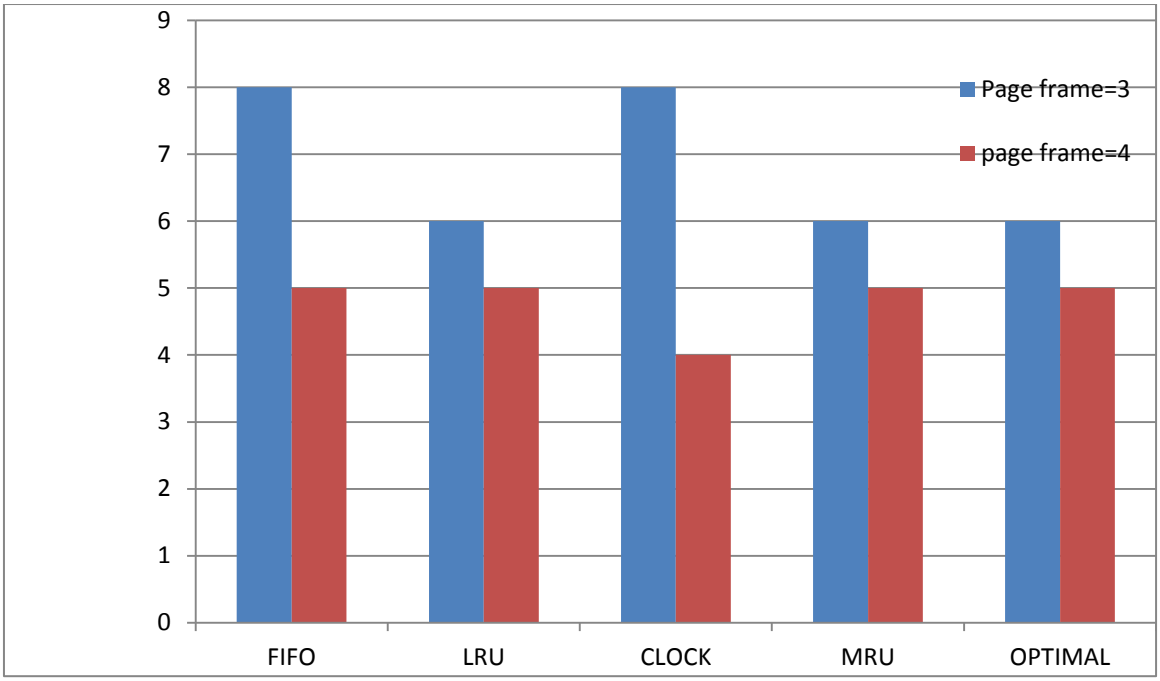

Fig 1: Comparison of page faults for the five replacement algorithms

According to table(1) and figure (1) with 3 page frames, FIFO generates 8 page faults, LRU generates 6 page faults, clock generates 8 page faults , MRU generates 6 page faults and Optimal generates 6 page faults. Similarly with 4 page frames, FIFO generates 5 page faults, LRU generates 5 page faults, clock generates 4 page faults , MRU generates 5 page faults and Optimal generates 5 page faults

4. CONCLUSIONS

1-The page replacement algorithms performance is measured in terms of number of page faults rate.
2- The page fault will decrease as the number of page frames increases.
3-When page frame=3 FIFO and CLOCK have an equal page fault=8 while LRU, MRU and OPTIMAL were = 6. When page frame=4 CLOCK has the lowest page fault=5
4- The best replacement algorithm depends mainly on the order and sequence of the pages that requested by the CPU.

REFERENCES

1- Salam Ayad, "A proposed multilevel replacement algorithm for cache memory", master thesis submitted to the department of computer science/ education collage/ al- mustansiriyah university, 2016.
2- Mahesh Kumar M , Renuka Rajendra ," AN INPUT ENHANCEMENT TECHNIQUE TO MAXIMIZE THE PERFORMANCE OF PAGE REPLACEMENT ALGORITHMS", International Journal of Research in Engineering and Technology, Volume: 04 Issue: 06 | June-2015.
3- Muhammad Waqar, Anas Bilal, Ambreen Malik and Imran Anwar , "COMPARATIVE ANALYSIS OF REPLACEMENT ALGORITHMS TECHNIQUES REGARDING TO TECHNICAL ASPECTS", European Journal of Engineering and Technology, Vol. 4 No. 5, 2016.
4- M.Saktheeswari*1 K.Sridharan*," A STUDY ON PAGE REPLACEMENT ALGORITHMS", International Journal of Technology and Engineering System (IJTES), Vol 3.No.2,2012
5- 11. Prof. D. Pradhan, "Cache Design", design approach, Technical Report, Department of Computer Science, University of Bristol, 2000.
6- 14. D. H. Albonesi, "Selective Cache Ways", Technical Report, On-Demand Cache Resource Allocation, Dept. of Electrical and Computer Engineering, University of Rochester, 2000.
7- 1Sourabh Shastri, 2Anand Sharma, 3Vibhakar Mansotra," Study of Page Replacement Algorithms and their analysis with C#",The International Journal Of Engineering And Science (IJES),|| Volume || 5 || Issue || 1 || Pages || PP -53-57 || 2016 ||
8- Genta Rexha1,Erand Elmazi2,Igli Tafa2," A Comparison of Three Page Replacement Algorithms: FIFO, LRU and Optimal", Academic Journal of Interdisciplinary Studies
MCSER Publishing, Rome-Italy, Vol 4 No 2 S2,August 2015 .

DOI : 10.31695/IJERAT.2022.8.8.1

9- Amit S. Chavan, Kartik R. Nayak, Keval D. Vora, Manish D. Purohit and Pramila M. Chawan," A Comparison of Page Replacement Algorithms", IACSIT International Journal of Engineering and Technology, Vol.3, No.2, April 2011.
10- Yannis Smaragdakis," General Adaptive Replacement Policies", ISMM'04, October 24–25, 2004, Vancouver, British Columbia, Canada.

C. Author: **safanahyder@yahoo.com**

DOI : 10.31695/IJERAT.2022.8.8.1