A Comparison of Page Replacement Algorithms: A Survey

Juhi Kumari, Sonam Kumari & Devendra Prasad
Department of Computer Science and Engineering,
Shivalik College of Engineering, Dehradun

Abstract-This paper is about algorithms specific to paging. For outline of general cache algorithms (e.g. processor, disk, database, web),In a computer operating system that uses paging for virtual memory management, page replacement algorithms decide which memory page into page out (swap out, write to disk) when a page of memory needs to be allocated. Paging happens when a page fault occurs and a free page cannot be used to satisfy the allocation, either because there are none, or because the number of free pages is lower than some threshold. When the page that was selected for replacement and paged out is referenced again it has to be paged in (read in from disk), and this involves waiting for I/O completion. This determines the quality of the page replacement algorithm: the less time waiting for page-ins, the better the algorithm. A page replacement algorithm looks at the limited information about accesses to the pages provided by hardware, and tries to guess which pages should be replaced to minimize the total number of page misses, while balancing this with the costs (primary storage and processor time) of the algorithm itself.

Keyword-Paging, Page fault, page replacement algorithm,

1. INTRODUCTION

In operating system, that uses paging for virtual memory management, page replacement algorithms decide which memory pages to page out (swap out, write to disk) when a page of memory needs to be allocated. Paging happens when a page fault occurs and a free page cannot be used to satisfy the allocation, either because there are none, or because the number of free pages is lower than some threshold.

When the page that was selected for replacement and paged out is referenced again it has to be paged in (read in from disk), and this involves waiting for I/O completion.

A page fault typically occurs when a process references to a page that is not marked present in main memory and needs to be brought from secondary memory. In such a case an existing page needs to be discarded. The selection of such a page is performed by page replacement algorithms which try to minimize the page fault rate at the least overhead. This paper outlines the major advanced page replacement algorithms. We start with basic algorithms such as optimal page replacement, LRU, FIFO.

A frame refers to a storage frame or central storage frame. In terms of physical memory, it is a fixed sized block in physical memory space, or a block of central storage. In computer architecture, frames are analogous to logical address space pages. Hit ratio = Total number of Hit Counts / Total number of Reference Counts

2. THEORETICAL BACKGROUND

Manisha Koranga et al.[1] reviewing and comparing the basic page replacement algorithms and analyze the effect of increasing number of frames on the page fault and to determine hit ratio in each case. The algorithms are implemented in C++ language and the results have been simulated using MATLAB.

Anvita Saxena [2] proposed a newer memory access patterns were explored, research mainly focused on formulating newer approaches to page replacement which could adapt to changing workloads. They also attempts to summarize major page replacement algorithms. they look at the traditional algorithms such as Optimal replacement, LRU, FIFO and also study the recent approaches such as Aging, ARC, CAR.

Amit S et al. [3] attempts to summarize major page replacement algorithms proposed till date. We look at the traditional algorithms such as LRU and CLOCK, and also study the recent approaches such as LIRS, CLOCK-Pro, ARC, and CAR.

Genta Rexha et al. [4} define the algorithm that realizes the best performance of the system. A good page replacement algorithm can reduce the page faults, when the program is executing, reduce the number of I/O, and then increase the system's efficiency effectively. The time is a critical point for the systems. An improvement in the performance of the system, having less page faults is made. Optimal results the best algorithm .FIFO has the worst performance. It has more page faults (degenerates) when the number of pages is increased. Many tests show, that FIFO, sometimes, makes a wrong decision. It deletes a page from memory and brings it back only after two steps. This takes many times, because it writes a page in disk and brings it back in main memory in two steps. LRU is the better algorithm to implement in these conditions.

3. PAGE REPLACEMENT ALGORITHM:

Page replacement algorithms are the techniques using which an Operating System decides which memory pages to swap out, write to disk when a page of memory needs to be allocated. Paging happens whenever a page fault occurs and a free page cannot be used for allocation purpose

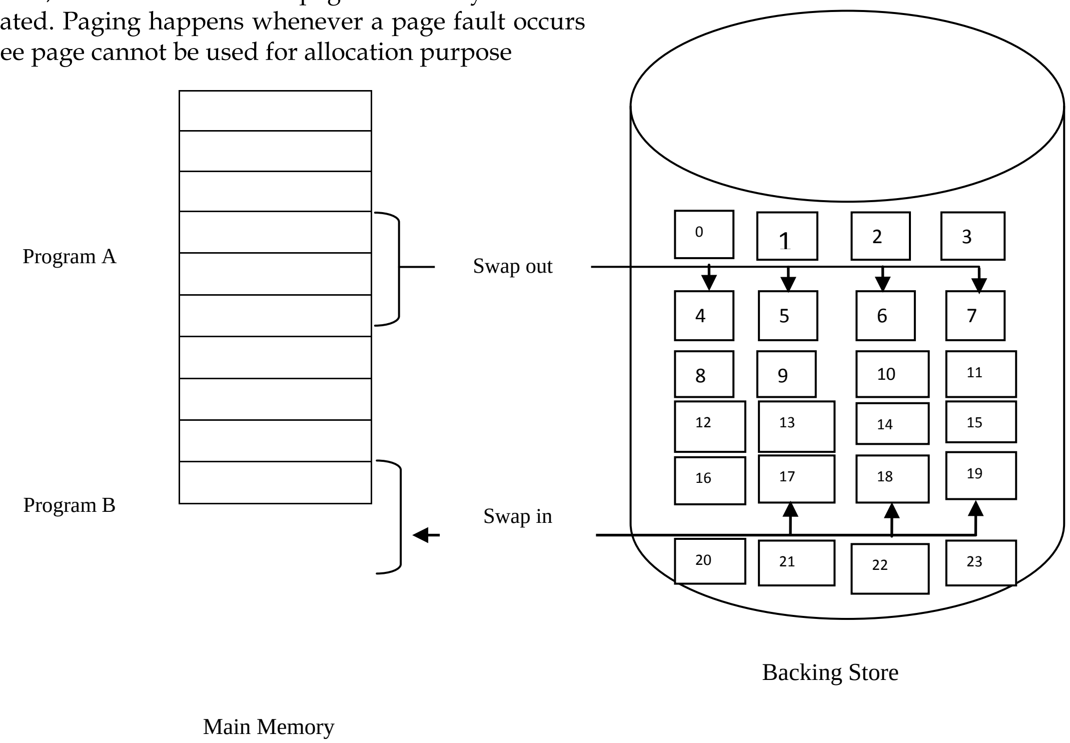


Figure1. Process for page in and page out in memory

accounting to reason that pages are not available or the number of free pages is lower than required pages.

When the page that was selected for replacement and was paged out, is referenced again, it has to read in from disk, and this requires for I/O completion. This process determines the quality of the page replacement algorithm: the lesser the time waiting for page-ins, the better is the algorithm.

A page replacement algorithm looks at the limited information about accessing the pages provided by hardware, and tries to select which pages should be replaced to minimize the total number of page misses, while balancing it with the costs of primary storage and processor time of the algorithm itself. There are many different page replacement algorithms. We evaluate an algorithm by running it on a particular string of memory reference and computing the number of page faults.

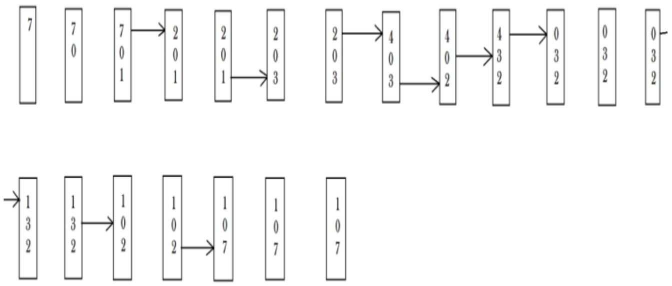
3.1 FIFO (First in First out) Page Replacement Algorithm − It is one of the simplest page replacement algorithm. The oldest page, which has spent the longest time in memory is chosen and replaced. This algorithm is implemented with the help of FIFO queue to hold the pages in memory. A page is inserted at the rear end of the queue and is replaced at the front of the queue.

Advantages and disadvantages of FIFO Algorithm:-

- FIFO is easy to understand.
- It is very easy to implement.
- Not always good at performance. It may replace an active page to bring a new one and thus may cause a page fault of that page immediately.
- Another unexpected side effect is the FIFO anomaly or Belady's anomaly. This anomaly says
- that the page fault rate may increase as the number of allocated page frames increases.

Example

String - 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1



Page fault=12
Page replacemet=9

Implementation

FIFO

1-frame = No of page fault - 20
2-frame= No of page fault -15
3-frame= No of page fault -15
4-frame= No of page fault -10
5-frame= No of page fault - 9

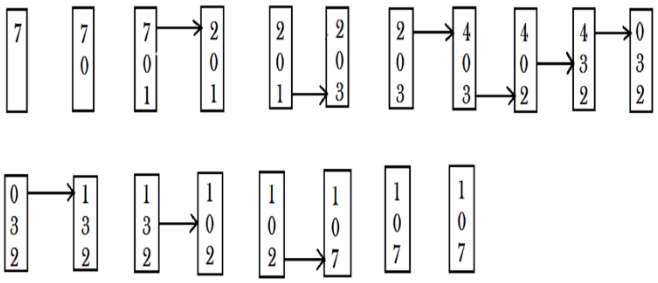
3.2 LRU (Least Recently Used) Algorithm – The Least Recently algorithm replaces the page that has not been used for the longest period of time. It is based on the observation that pages that have not been used for long time will probably remain unused for the longest time and are to be replaced.

Advantages and disadvantages of LRU Algorithm:-

- LRU page replacement algorithm is quiet efficient.
- It does not suffer from Belady's Anomaly.
- Its implementation is not very easy.
- Its implementation may require substantial hardware assistance used (LRU)

Example

String -7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1



No.of faults=12
No.of replacement=9

Implementation

LRU

1-frame= No of page fault -20
2-frame= No of page fault -17
3-frame= No of page fault - 12
4-frame= No of page fault - 8
5-frame=No of page fault -7

3.3 Optimal Page Replacement Algorithm

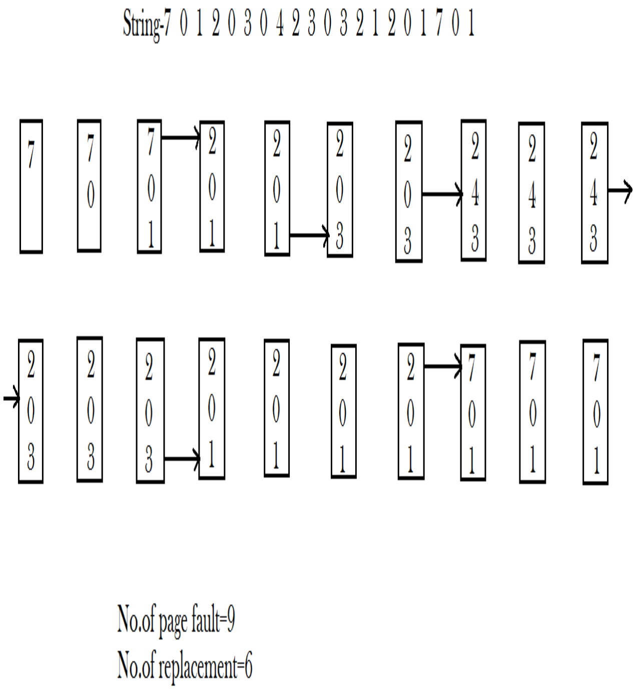
An optimal page-replacement algorithm has the lowest page-fault rate of all algorithms. An optimal page-replacement algorithm exists, and has been called OPT or MIN.

Replace the page that will not be used for the longest period of time. Use the time when a page is to be used.

Advantages and disadvantages of Optimal Page Replacement

- Lowest page fault rate.
- Twice as good as first in first out page replacement algorithm.
- Difficult to implement.
- It need forecast.

Example



Implementation
OPR
1-frame=No of page fault-20
2-frame=No of page fault-13
3-frame=No of page fault-9
4-frame=No of page fault-8
5-frame=No of page fault-7

String 2-1 2 3 4 2 1 5 6 2 1 2 3 7 6 3 2 1 2 3 6

]

4. Result Analysis

TABLE 1:
COMPARISON BETWEEN FIFO, LRU AND OPR

	NUMBER OF FRAMES	NO. OF PAGE FAULT IN FIFO	NO. OF PAGE FAULT IN LRU	NO. OF PAGE FAULT IN OPR
STING 1	1-FRAME	20	20	20
	2-FRAME	15	17	13
	3-FRAME	15	12	9
	4-FRAME	10	8	8
	5-FRAME	9	7	7
STRING 2	1-FRAME	20	20	20
	2-FRAME	18	18	15
	3-FRAME	15	16	11
	4-FRAME	10	14	8
	5-FRAME	8	10	7

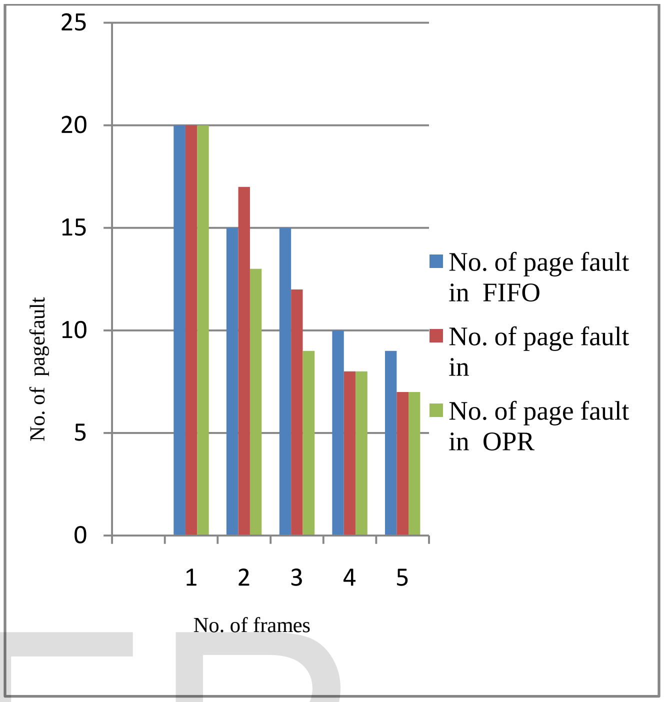


Figure2- performance analysis of page replacement algorithms for sting 1

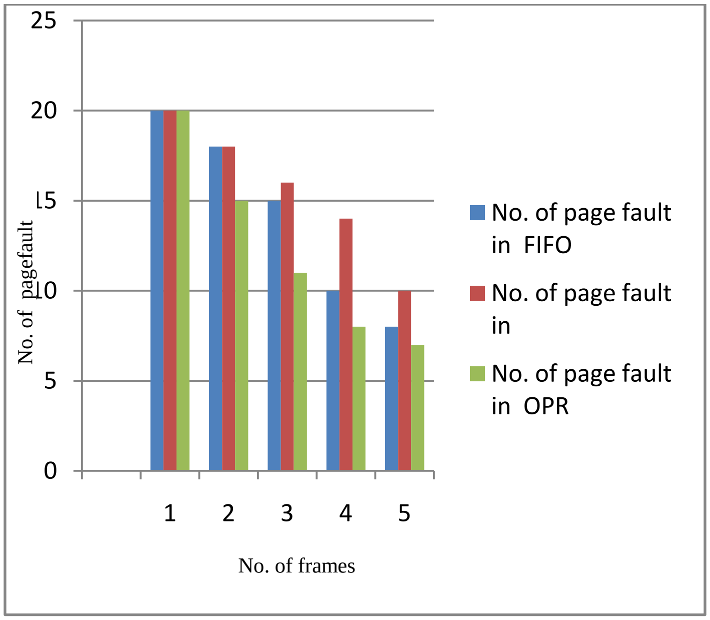


Figure3- performance analysis of page replacement algorithms for sting 2

Page fault for two different input strings is shown in figure 2 and figure3 for different page replacement algorithms. Based on analysis the optimal page replacement algorithm (OPR) have a minimum page fault for all the frame size.

5. CONCLUSION

Page Replacement Algorithms is used to prevent over-allocation of memory by modifying page-fault service routine to include page replacement. It will use dirty (modify) bit to reduce overhead of page transfers; only modified pages written back to disk. Page replacement completes separation between logical memory and physical memory; large virtual memory can be provided on smaller physical memory. It will find location of desired page on disk. It will find free frame only if free frame, use it ,if no free frame, page replacement algorithm selects victim frame, if victim dirty write back to disk. Read desired page into (newly) free frame; update page and frame tables. Restart faulting process. Want lowest page-fault rate. Evaluate algorithm by running on given string of memory references (reference string) and compute number of page faults. The Optimal Page Replacement Algorithm; the page that will not be referenced again for the longest time is replaced (prediction of the future; purely theoretical, but useful for comparison.).The Not Recently Used Page Replacement Algorithm; algorithm removes a page at random. The First-In, First-Out (FIFO) Page Replacement Algorithm; FIFO the frames are treated as a circular list; the oldest (longest resident) page is replaced. The Second Chance Page Replacement Algorithm; look for an old page that has not been referenced in the previous clock interval, avoids the problem of throwing out of heavily used page. The Least Recently Used (LRU) Page Replacement Algorithm; LRU the frame whose contents have not been used for the longest time is replaced.

REFERENCES:

[1] Manisha Koranga, Nisha Koranga, “Analysis on Page Replacement Algorithms with Variable Number of Frames”, International Journal of Advanced Research in COMPUTER SCIENCE AND SOFTWARE ENGINEERING, Volume 4, Issue 7, July 2014.

[2] Anvita Saxena , “A Study of Page Replacement Algorithms”, International Journal of Engineering Research and General Science Volume 2, Issue 4, June-July, 2014,ISSN 2091-2730.

[3] 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.

[4] Genta Rexha1 Erand Elmazi2 Igli Tafa, “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.

[5] L. A. Belady, ―A study of replacement algorithms for virtual storage computers, ‖ IBM Sys. J., vol. 5, no. 2, pp. 78–101, 1966.

[6] F. J. Corbat ́o, ―A paging experiment with the multics system, ‖ in In Honor of P. M. Morse, pp. 217–228, MIT Press, 1969. Also as MIT Project MAC Report MAC-M-384, May 1968.

[7] J. E. O'neil, P. E. O'neil and G. Weikum, “An optimality Proof of the LRU-K Page Replacement Algorithm”, Journal of the ACM, pp. 92-112, 1999.

[8] Kavar C. C. Parmar S. S. (2013). Performance analysis of LRU page replacement algorithm. International Journal of Engineering Research and Applications (IJERA) Vol. 3. Issue 1. pp.2070-2076

[9] tutorial.page replacement.

[10]www.youtube.co