

Operating System: Page Replacement Algorithms

Prepared By:

Dr. Sanjeev Gangwar

Assistant Professor,

Department of Computer Applications,

VBS Purvanchal University, Jaunpur

PAGE REPLACEMENT ALGORITHMS

In an operating system that uses paging for memory management, a page replacement algorithm is needed to decide which page needs to be replaced when new page comes in.

Principle of Page Replacement

The page replacement is done by swapping. The required pages from backup storage to main memory and vice-versa. This swapping is done by checking the contents of physical memory. If there is a free frame in the memory then swap-in the required page into the frame which is free.

In case, if there is no frame in physical memory then first find the frame which is not currently in use. The content of this frame is swapped-out from memory to backup storage. Then bring the required page in the frame which is now free.

As studied in Demand Paging, only certain pages of a process are loaded initially into the memory. This allows us to get more number of processes into the memory at the same time. but what happens when a process requests for more pages and no free memory is available to bring them in. Following steps can be taken to deal with this problem:

1. Put the process in the wait queue, until any other process finishes its execution thereby freeing frames.
2. Or, remove some other process completely from the memory to free frames.
3. Or, find some pages that are not being used right now, move them to the disk to get free Frames.

This technique is called **Page replacement** and is most commonly used.

Page Fault – A page fault happens when a running program accesses a memory page that is mapped into the virtual address space, but not loaded in physical memory.

Since actual physical memory is much smaller than virtual memory, page faults happen. In case of page fault, Operating System might have to replace one of the existing pages with the newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce the number of page faults.

Page Replacement Algorithm

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. The string of memory references is called reference string. Reference strings are generated artificially or by tracing a given system and recording the address of each memory reference.

To determine the number of page faults for a particular reference string and page replacement algorithm, we also need to know the number of page frames available. As the number of frames available increase, the number of page faults will decrease.

We shall use the following page reference string to explain the page replacement algorithm

5, 0, 1, 0, 2, 3, 0, 2, 4, 3, 3, 2, 0, 2, 1, 2, 7, 0, 1, 1, 0

for a memory with three frames.

FIFO (First In First Out) Page Replacement

- Simplest page replacement algorithm
- Oldest page in main memory is the one which will be selected for replacement. Easy to implement, keep a list, replace pages from the tail and add new pages

		1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
	0	0	0	0	3	3	3	3	3	3	2	2	2	2	2	2	0	0	0	0
5	5	5	5	2	2	2	2	4	4	4	4	4	4	4	4	7	7	7	7	7
			H				H		H	H		H	H		H			H	H	H

Total Page Fault= 11

Number of Hit= 10

Rate of page fault = No. of page fault/No. of frames

$$11/3= 3.6$$

Beledy's Anomaly:

Beledy's anomaly is that the page-fault rate may increase as the number of allocated frames increases.

For example if we consider the following page reference string

0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4 with no of frame 3, we get total 9 page fault as shown in fig (a). But if we increase the no. of frame from 3 to 4 we get 10 page fault as shown in fig (b)

Case 1: Frame=3

		2	2	2	1	1	1	1	1	3	3
	1	1	1	0	0	0	0	0	2	2	2
0	0	0	3	3	3	4	4	4	4	4	4
							H	H			H

Fig (a)

Page Fault= 9

Case 2: Frame=4

			3	3	3	3	3	3	2	2	2
		2	2	2	2	2	2	1	1	1	1
	1	1	1	1	1	1	0	0	0	0	4
0	0	0	0	0	0	4	4	4	4	3	3
				H	H						

Fig (b)

Page Fault= 10

Advantages:

- It is simple and easy to understand & implement.

Disadvantages:

- The process effectiveness is low.
- When we increase the number of frames while using FIFO, we are giving more memory to processes. So, page fault should decrease, but here the page faults are increasing. This problem is called as Belady's anomaly

LRU (Least-Recently-Used) Page Replacement

In this algorithm, the page that has been not used for longest period of time is selected for replacement.

Example: Reference string is

5, 0, 1, 0, 2, 3, 0, 2, 4, 3, 3, 2, 0, 2, 1, 2, 7, 0, 1, 1, 0

		1	1	1	3	3	3	4	4	4	4	0	0	0	0	7	7	7	7	7
	0	0	0	0	0	0	0	0	3	3	3	3	3	1	1	1	0	0	0	0
5	5	5	5	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1
			H			H	H			H	H		H		H				H	H

Total Page Fault= 12

Number of Hit= 9

Rate of page fault= No. of page fault/No. of frames = 12/3= 4

Advantages:

- It is open for full analysis.
- In this, we replace the page which is least recently used, thus free from Belady’s Anomaly.
- Easy to choose page which has faulted and hasn’t been used for a long time.

Disadvantages:

- It requires additional Data Structure to be implemented.
- Hardware assistance is high.

Optimal Page Replacement

This algorithm state that: replace that page which will not be used for longest period of time i.e. future knowledge of reference string is required.

Example: Reference string is

5, 0, 1, 0, 2, 3, 0, 2, 4, 3, 3, 2, 0, 2, 1, 2, 7, 0, 1, 1, 0

		1	1	1	3	3	3	3	3	3	3	3	3	1	1	1	1	1	1	1
	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0	0	0	0	0	0
5	5	5	5	2	2	2	2	2	2	2	2	2	2	2	2	7	7	7	7	7
			H			H	H		H	H	H		H		H		H	H	H	H

Total Page Fault= 9

Number of Hit= 12

Rate of page fault= No. of page fault/No. of frames = 9/3= 3

Advantages:

- Complexity is less and easy to implement.
- Assistance needed is low i.e Data Structure used are easy and light.

Disadvantages:

- Optimal page replacement algorithm is perfect, but not possible in practice as the operating system cannot know future requests.
- Error handling is tough.

Solved Example: consider the following page reference string

1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6

Compare the number of page faults for FIFO, LRU and optimal page replacement. Assume number of frame is 4.

Sol. FIFO page replacement

			4	4	4	4	4	4	1	1	1	1	1	1	2	2	2	2	2
		3	3	3	3	3	3	2	2	2	2	2	6	6	6	6	6	6	6
	2	2	2	2	2	2	6	6	6	6	6	7	7	7	7	7	7	3	3
1	1	1	1	1	1	5	5	5	5	5	3	3	3	3	3	1	1	1	1
				H	H					H				H			H		H

Total Page Fault= 14

LRU page replacement

			4	4	4	4	6	6	6	6	6	7	7	7	7	1	1	1	1
		3	3	3	3	5	5	5	5	5	3	3	3	3	3	3	3	3	3
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6
				H	H			H	H	H				H	H		H	H	H

Total Page Fault= 10

Optimal page replacement

			4	4	4	5	6	6	6	6	6	6	6	6	6	6	6	6
		3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	7	7	7	7	1	1	1
				H	H			H	H	H	H		H	H	H		H	H

Total Page Fault= 8

References:

- (1) Abraham Silberschatz, Galvin & Gagne, Operating System Concepts, John Wiley & Sons, INC.
- (2) Harvay M.Deital, Introduction to Operating System, Addition Wesley Publication Company.
- (3) Andrew S.Tanenbaum, Operating System Design and Implementation, PHI