SRES's Sanjivani College of Engineering, Kopargaon
(*An Autonomous Institute*)
Department of Computer Engineering

SPOS Lab Manual

Assignment No. 11

AIM:
Write a program to implement Page Replacement Algorithm.

PROBLEM DEFINITION:
Write a program to implement FIFO, LRU and OPTIMAL Page Replacement Algorithm.

OBJECTIVES:
To find the page faults and page hits by FIFO, LRU and OPTIMAL algorithm

INPUT:
- Number of pages
- Page reference string
- Number of frames

OUTPUT:
- Number of page faults
- Number of page hits

THEORY:
Paging is a process of reading data from, and writing data to, the secondary storage. It is a memory management scheme that is used to retrieve processes from the secondary memory in the form of pages and store them in the primary memory. The main objective of paging is to divide each process in the form of pages of fixed size. These pages are stored in the main memory in frames. Pages of a process are only brought from the secondary memory to the main memory when they are needed.

When an executing process refers to a page, it is first searched in the main memory. If it is not present in the main memory, a page fault occurs.

** Page Fault is the condition in which a running process refers to a page that is not loaded in the main memory.

In such a case, the OS has to bring the page from the secondary storage into the main memory. This may cause some pages in the main memory to be replaced due to limited storage. A Page Replacement Algorithm is required to decide which page needs to be replaced.

Page Replacement Algorithm

Page Replacement Algorithm decides which page to remove, also called swap out when a new page needs to be loaded into the main memory. Page Replacement happens when a requested page is not present in the main memory and the available space is not sufficient for allocation to the requested page.

When the page that was selected for replacement was paged out, and 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 tries to select which pages should be replaced so as to minimize the total number of page misses. There are many different page replacement algorithms. These algorithms are evaluated by running them on a particular string of memory reference and computing the number of page faults. The fewer is the page faults the better is the algorithm for that situation.

** If a process requests for page and that page is found in the main memory then it is called **page hit**, otherwise **page miss** or **page fault**.

Page Replacement Algorithms:

- First In First Out (FIFO)
- Least Recently Used (LRU)
- Optimal Page Replacement

A] First In First Out (FIFO)

This is the simplest page replacement algorithm. In this algorithm, the OS maintains a queue that keeps track of all the pages in memory, with the oldest page at the front and the most recent page at the back.

When there is a need for page replacement, the FIFO algorithm, swaps out the page at the front of the queue, that is the page which has been in the memory for the longest time.

For Example:

page reference string of size = 12
page reference string: 1, 2, 3, 4, 5, 1, 3, 1, 6, 3, 2, 3
frame size = 4(i.e. maximum 4 pages in a frame).

1	2	3	4	5	1	3	1	6	3	2	3

1	1	1	1	5	5	5	5	5	5	2	2
	2	2	2	2	1	1	1	1	1	1	1
		3	3	3	3	3	3	6	6	6	6
			4	4	4	4	4	4	3	3	3
M	M	M	M	M	M	H	H	M	M	M	H

M = Miss
H = Hit

Total Page Fault = 9

Initially, all 4 slots are empty, so when 1, 2, 3, 4 came they are allocated to the empty slots in order of their arrival. This is page fault as 1, 2, 3, 4 are not available in memory.

When 5 comes, it is not available in memory so page fault occurs and it replaces the oldest page in memory, i.e., 1.

When 1 comes, it is not available in memory so page fault occurs and it replaces the oldest page in memory, i.e., 2.

When 3,1 comes, it is available in the memory, i.e., Page Hit, so no replacement occurs.

When 6 comes, it is not available in memory so page fault occurs and it replaces the oldest page in memory, i.e., 3.

When 3 comes, it is not available in memory so page fault occurs and it replaces the oldest page in memory, i.e., 4.

When 2 comes, it is not available in memory so page fault occurs and it replaces the oldest page in memory, i.e., 5.

When 3 comes, it is available in the memory, i.e., Page Hit, so no replacement occurs.

Page Fault ratio = 9/12 i.e., total miss/total possible cases

Advantages

- Simple and easy to implement.
- Low overhead.

Disadvantages

- Poor performance.
- Doesn't consider the frequency of use or last used time, simply replaces the oldest page.
- Suffers from Belady's Anomaly(i.e. more page faults when we increase the number of page frames).

B] Least Recently Used (LRU)

Least Recently Used page replacement algorithm keeps track of page usage over a short period of time. It works on the idea that the pages that have been most heavily used in the past are most likely to be used heavily in the future too.

In LRU, whenever page replacement happens, the page which has not been used for the longest amount of time is replaced.

For Example

1	2	3	4	5	1	3	1	6	3	2	3

1	1	1	1	5	5	5	5	5	5	2	2
	2	2	2	2	1	1	1	1	1	1	1
		3	3	3	3	3	3	3	3	3	3
			4	4	4	4	4	6	6	6	6
M	M	M	M	M	M	H	H	M	H	M	H

M = Miss
H = Hit
Total Page Fault = 8

Initially, all 4 slots are empty, so when 1, 2, 3, 4 came they are allocated to the empty slots in order of their arrival. This is page fault as 1, 2, 3, 4 are not available in memory.

When 5 comes, it is not available in memory so page fault occurs and it replaces 1 which is the least recently used page.

When 1 comes, it is not available in memory so page fault occurs and it replaces 2.

When 3,1 comes, it is available in the memory, i.e., Page Hit, so no replacement occurs.

When 6 comes, it is not available in memory so page fault occurs and it replaces 4.

When 3 comes, it is available in the memory, i.e., Page Hit, so no replacement occurs.

When 2 comes, it is not available in memory so page fault occurs and it replaces 5.

When 3 comes, it is available in the memory, i.e., Page Hit, so no replacement occurs.

Page Fault ratio = 8/12

Advantages

- Efficient.
- Doesn't suffer from Belady's Anomaly.

Disadvantages

- Complex Implementation.
- Expensive.
- Requires hardware support.

C] Optimal Page Replacement

Optimal Page Replacement algorithm is the best page replacement algorithm as it gives the least number of page faults. It is also known as OPT, clairvoyant replacement algorithm, or Belady's optimal page replacement policy.

In this algorithm, pages are replaced which would not be used for the longest duration of time in the future, i.e., the pages in the memory which are going to be referred farthest in the future are replaced.

This algorithm was introduced long back and is difficult to implement because it requires future knowledge of the program behaviour. However, it is possible to implement optimal page replacement on the second run by using the page reference information collected on the first run.

For Example

1	2	3	4	5	1	3	1	6	3	2	3

1	1	1	1	1	1	1	1	6	6	6	6
	2	2	2	2	2	2	2	2	2	2	2
		3	3	3	3	3	3	3	3	3	3
			4	5	5	5	5	5	5	5	5
M	M	M	M	M	H	H	H	M	H	H	H

M = Miss
H = Hit
Total Page Fault = 6

Initially, all 4 slots are empty, so when 1, 2, 3, 4 came they are allocated to the empty slots in order of their arrival. This is page fault as 1, 2, 3, 4 are not available in memory.

When 5 comes, it is not available in memory so page fault occurs and it replaces 4 which is going to be used farthest in the future among 1, 2, 3, 4.

When 1,3,1 comes, they are available in the memory, i.e., Page Hit, so no replacement occurs.

When 6 comes, it is not available in memory so page fault occurs and it replaces 1.

When 3, 2, 3 comes, it is available in the memory, i.e., Page Hit, so no replacement occurs.

Page Fault ratio = 6/12

Advantages

- Easy to Implement.
- Simple data structures are used.

- Highly efficient and less complex.

Disadvantages

- Requires future knowledge of the program.
- Time-consuming.
- Practical Implementation is not possible because the operating system is unable to track the future request

ALGORITHM:

A] FIFO Page Replacement Algorithm

It is a very simple way of Page replacement and is referred to as First in First Out. This algorithm mainly replaces the oldest page that has been present in the main memory for the longest time.

- This algorithm is implemented by keeping the track of all the pages in the queue.
- As new pages are requested and are swapped in, they are added to the tail of a queue and the page which is at the head becomes the victim.
- This is not an effective way of page replacement but it can be used for small systems.

B] LRU Page Replacement Algorithm in OS

This algorithm stands for "Least recent used" and this algorithm helps the Operating system to search those pages that are used over a short duration of time frame.

- The page that has not been used for the longest time in the main memory will be selected for replacement.
- This algorithm is easy to implement.
- This algorithm makes use of the counter along with the even-page.

C] Optimal Page Replacement Algorithm

This algorithm mainly replaces the page that will not be used for the longest time in the future. The practical implementation of this algorithm is not possible.

- Practical implementation is not possible because we cannot predict in advance those pages that will not be used for the longest time in the future.
- This algorithm leads to less number of page faults and thus is the best-known algorithm

Also, this algorithm can be used to measure the performance of other algorithms.

- For the best result, the implementation of data structures is very easy

CONCLUSION:

In this assignment we learn about Page Replacement algorithms FIFO, LRU and OPTIMAL, also page fault and page hit in each of the replacement policy.

References:

Operating System Concepts by Abraham Silberschatz, Peter B. Galvin, Greg Gagne

Prepared by
Prof.N.G.Pardeshi
Subject Teacher

Approved by
Dr. D.B.Kshirsagar
HOD