

Page: 6
Date: 11/1

Assignment D1

Title: Implementation of Page Replacement Algorithm.

Date of Completion:

Problem Statement:

Write a Java Program (Using OOP Features) to implement Paging Simulation Using:

- 1) Least Recently Used (LRU)
- 2) Optimal Algorithms

Objectives:

Understand Virtual Memory Management

Analyse the need of page replacement

Compare various page replacement algorithms

Outcomes: I will be able to:

Implement Page Replacement Algorithms like

LRU, Optimal

Compare the page replacement algorithm based on hit ratio.

Software & Hardware Requirements:

64 bit open source Fedora OS

8 GB RAM

500 GB HDD

Eclipse IDE

i5 machine

Theory:

Wherever there is a page reference for which page needed is not present in the memory, that event is called page fault.

In such cases we have to make space in memory for which the existing page is replaced by the new page.

But we cannot replace any random page. Instead we ought to replace a page which is not being used currently.

There are some algorithms based on which we can select an appropriate page replacement policy.

Designing an appropriate algorithm to solve this problem is an important task because disk I/O is expensive.

In an OS that uses paging for memory management, a page replacement algorithm is needed to decide which pages need to be replaced when the new page comes in.

Page Fault:

A page fault occurs when a running program accesses a memory that is mapped into the virtual address space but not loaded in the physical memory. One of the existing pages is replaced by the new one.

Different page replacement algorithm suggests different ways to decide which page to replace. The target for all algorithms is to reduce page faults.

Program Outcomes

Page: 6
Date: 11/10/2023

Page Replacement Algorithms

Optimal Page Replacement: In this algorithm, pages are replaced with pages that would not be used for the longest duration of time in the future, i.e. future knowledge of references string is required. It has a low page fault rate.

Optimal Page Replacement is impractical as OS cannot know future requests in advance. It is used to set a benchmark so that other replacement algorithms can be analysed against it.

Consider the Example:

Page Reference String

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 5, 2, 1, 6, 7, 4

No. of frames = 3

	7	0	1	2	0	3	0	4	2	3	0	3	5	2	1	6	7	4
1	7	0	1	2	0	3	0	4	2	3	0	3	2	1	9			
i	7	0	1	2	0	3	0	4	2	3	0	3	2	1	9			

No. of Page Faults = 9

Page Fault Ratio = $\frac{9}{15} = 0.6$

Page: 6
Date: 11/6

2) Least Recently Used (LRU)

Page that has not been used for the longest time is selected for replacement. LRU is theoretically reliable.

To implement LRU, it is necessary to maintain a linked list of all pages in memory with the most recently used page at the front & least recently used at the rear.

List must be updated on every memory reference.

Finding a page in the list, deleting it & then moving it to the front is a very time consuming operation.

Consider the Example,

Page Reference String:

4 3 2 1 0 4 3 5 0 4 2 3 2 1 0 5

No. of frames = 3

4	3	2	1	4	3	5	4	3	2	1
u	3	2	1	4	3	5	u	3	2	1
u	3	2	1	4	3	5	u	3	2	1
—	—	—	—	—	—	—	—	—	—	—

No. of page faults = 9

Page Fault Ratio = $\frac{9}{11} = 0.8181$

Program Outcomes

Page: 110
Date: 11/10

Test Cases

Reference String : 0 2 1 6 4 0 10 3 1 2 1
No. of frames : 4

Algorithm	Page Fault	Page Fault Ratio
LRU	8	0.67
Optimal	6	0.50

Reference String : 3 2 1 0 3 2 4 3 2 1 0 4

No. of frames : 3

Algorithm	Page Fault	Page Fault Ratio
LRU	10	0.83
Optimal	7	0.58

Analysis :

Page Faults are in the following order :

FIFO > LRU > Optimal

In optimal we know the future references of a page & then replace it

LRU replaces page which was last referenced

Conclusion :

The Page Replacement Algorithms like LRU & Optimal have been successfully implemented & compared with one another.