

Assignment No. 6

Date:

TITLE: Implement the C program for Page Replacement Algorithms: FCFS, LRU, and Optimal for frame size as minimum three.

OBJECTIVE:

- ⑩ Study various memory management techniques
- ⑩ Implement the concept of memory management techniques using C.

SOFTWARE REQUIREMENTS:

1. Ubuntu 16.04
2. GNU C Compiler

THEORY:

Page Replacement Algorithms :

In operating systems that use paging for memory management, page replacement algorithm are needed to decide which page needed to be replaced when new page comes in. Whenever a new page is referred and not present in memory, page fault occurs and Operating System replaces one of the existing pages with newly needed page. Different page replacement algorithms suggest different ways to decide which page to replace. The target for all algorithms is to reduce number of page faults

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.

First Come First Serve (FCFS)

This is the simplest page replacement algorithm. In this algorithm, operating system keeps track of all pages in the memory in a queue, oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal. 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.

This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal.

Example-1 Consider page reference string 1, 3, 0, 3, 5, 6 with 3 page frames. Find number of page faults.

Page reference		1, 3, 0, 3, 5, 6, 3					
1	3	0	3	5	6	3	
		0	0	0	0	3	
	3	3	3	3	6	6	
1	1	1	1	5	5	5	
Miss	Miss	Miss	Hit	Miss	Miss	Miss	
Total Page Fault = 6							

Initially all slots are empty, so when 1, 3, 0 came they are allocated to the empty slots —> **3 Page Faults.**

when 3 comes, it is already in memory so —> **0 Page Faults.**

Then 5 comes, it is not available in memory so it replaces the oldest page slot i.e 1. —> **1 Page Fault.**

6 comes, it is also not available in memory so it replaces the oldest page slot i.e 3 —> **1 Page Fault.**

Finally when 3 come it is not available so it replaces 0 **1 page fault**

• Optimal Page replacement –

In this algorithm, pages are replaced which would not be used for the longest duration of time in the future.

Example-2: Consider the page references 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, with 4 page frame. Find number of page fault.

Page reference	7,0,1,2,0,3,0,4,2,3,0,3,2,3							No. of Page frame - 4						
7	0	1	2	0	3	0	4	2	3	0	3	2	3	
<div></div>	<div></div>	<div></div>	<div>2</div>	<div>2</div>	<div>2</div>	<div>2</div>	<div>2</div>	<div>2</div>	<div>2</div>	<div>2</div>	<div>2</div>	<div>2</div>	<div>2</div>	
<div></div>	<div></div>	<div>1</div>	<div>1</div>	<div>1</div>	<div>1</div>	<div>1</div>	<div>4</div>	<div>4</div>	<div>4</div>	<div>4</div>	<div>4</div>	<div>4</div>	<div>4</div>	
<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	<div>0</div>	
<div>7</div>	<div>7</div>	<div>7</div>	<div>7</div>	<div>7</div>	<div>3</div>	<div>3</div>	<div>3</div>	<div>3</div>	<div>3</div>	<div>3</div>	<div>3</div>	<div>3</div>	<div>3</div>	
Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit	Hit	Hit	Hit	
Total Page Fault = 6														

Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> **4 Page faults**

0 is already there so —> **0 Page fault.**

when 3 came it will take the place of 7 because it is not used for the longest duration of time in the future.—> **1 Page fault.**

0 is already there so —> **0 Page fault..**

4 will takes place of 1 —> **1 Page Fault.**

Now for the further page reference string —> **0 Page fault** because they are already available in the memory.

Optimal page replacement is perfect, but not possible in practice as the operating system cannot know future requests. The use of Optimal Page replacement is to set up a benchmark so that other replacement algorithms can be analyzed against it.

- **Least Recently Used –**

In this algorithm page will be replaced which is least recently used.

Example-3 Consider the page reference string 7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2 with 4 page frames. Find number of page faults.

Page reference	7,0,1,2,0,3,0,4,2,3,0,3,2,3													No. of Page frame - 4
7	0	1	2	0	3	0	4	2	3	0	3	2	3	
			2	2	2	2	2	2	2	2	2	2	2	
		1	1	1	1	1	4	4	4	4	4	4	4	
	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	7	7	7	7	3	3	3	3	3	3	3	3	3	
Miss	Miss	Miss	Miss	Hit	Miss	Hit	Miss	Hit	Hit	Hit	Hit	Hit	Hit	
Total Page Fault = 6														

Here LRU has same number of page fault as optimal but it may differ according to question.

Initially all slots are empty, so when 7 0 1 2 are allocated to the empty slots —> **4 Page faults**

0 is already there so —> **0 Page fault.**

when 3 came it will take the place of 7 because it is least recently used —> **1 Page fault**

0 is already in memory so —> **0 Page fault.**

4 will take place of 1 —> **1 Page Fault**

Now for the further page reference string —> **0 Page fault** because they are already available in the memory.

Conclusion: Thus we have studied and Implemented Page Replacement Algorithms: FCFS, LRU, and Optimal for frame size as minimum three.