

Page Replacement Algorithms

Aim: to write c programs that implement page replacement techniques namely, FIFO, LRU, Optimal

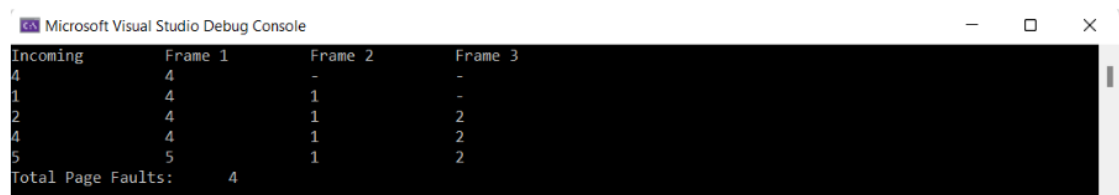
Description:

Page replacement algorithms are an important part of virtual memory management, and it helps the OS to decide which memory page can be moved out making space for the currently needed page. However, the ultimate objective of all page replacement algorithms is to reduce the number of page faults.

- a. **FIFO:** 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.

Algorithm steps:

1. Start to traverse the pages.
2. If the memory holds fewer pages, then the capacity else goes to step 5.
3. Push pages in the queue one at a time until the queue reaches its maximum capacity or all page requests are fulfilled.
4. If the current page is present in the memory, do nothing.
5. Else, pop the topmost page from the queue as it was inserted first.
6. Replace the topmost page with the current page from the string.
7. Increment the page faults.
8. Stop

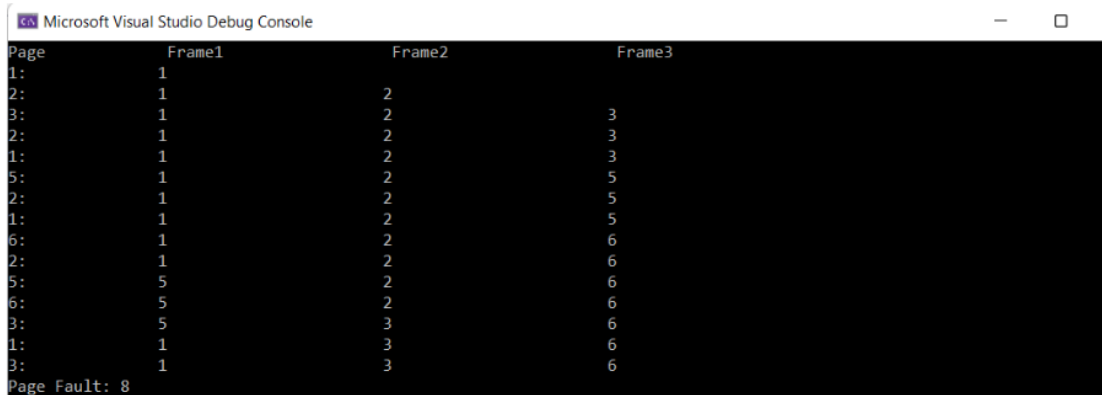


Incoming	Frame 1	Frame 2	Frame 3
4	4	-	-
1	4	1	-
2	4	1	2
4	4	1	2
5	5	1	2
Total Page Faults:		4	

b. **LRU:** In this algorithm page will be replaced which is least recently used

Algorithm steps:

1. Start the process
2. Declare the page size
3. Determine the number of pages to be inserted.
4. Get the value.
5. Declare the counter and stack value.
6. Choose the least recently used page by the counter value.
7. Stack them as per the selection.
8. Display the values.
9. Terminate the process.



Microsoft Visual Studio Debug Console

Page	Frame1	Frame2	Frame3
1:	1		
2:	1	2	
3:	1	2	3
2:	1	2	3
1:	1	2	3
5:	1	2	5
2:	1	2	5
1:	1	2	5
6:	1	2	6
2:	1	2	6
5:	5	2	6
6:	5	2	6
3:	5	3	6
1:	1	3	6
3:	1	3	6

Page Fault: 8

c. OPTIMAL: In this algorithm, pages are replaced which would not be used for the longest duration of time in the future. This algorithm will give us less page fault when compared to other page replacement algorithms.

Algorithm steps:

1. Push the first page in the stack as per the memory demand.
2. Push the second page as per the memory demand.
3. Push the third page until the memory is full.
4. As the queue is full, the page which is least recently used is popped.
5. Repeat step 4 until the page demand continues and until the processing is over.
6. Terminate the program.

Microsoft Visual Studio Debug Console

Stream	Frame1	Frame2	Frame3
7	7	-	-
0	7	0	-
1	7	0	1
2	2	0	1
0	2	0	1
3	2	0	3
0	2	0	3
4	2	4	3
2	2	4	3
3	2	4	3
0	2	0	3
3	2	0	3
2	2	0	3
1	2	0	1
2	2	0	1
0	2	0	1
1	2	0	1
7	7	0	1
0	7	0	1
1	7	0	1

Hits: 11
Faults: 0

Extra Questions:

1. What is meant by page fault?
2. What is meant by paging?
3. What is page hit and page fault rate?
4. List the various page replacement algorithm