Department Of Computer Science and Engineering

Course Title: Operating System Lab

Course Code: CSE 406

Title: FIFO page replacement Algorithm

Submitted To,

Atia Rahman Orthi
Lecturer
Department Of Computer Science & Engineering

Submitted By,

Md. Atikul Islam Atik
Reg No:21201063
Roll:63
Sec:B1

Submission Date: 24-04-2025

Input: 1 3 0 3 5 6 3

Frame Size= 3

Output: Total Page Faults = 6.

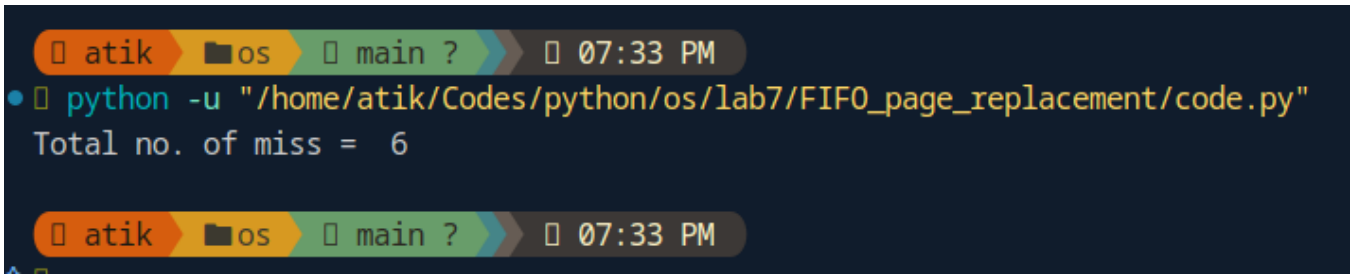
Code Snapshot:

```
from collections import deque

def fifo_page_replacement(page_reference, frame_size):
   memory =[]
   miss = 0
   queue = deque()


   for page in page_reference:
       miss +=1

       if len(memory) < frame_size:
           memory.append(page)
           queue.append(page)
       else:
           oldest_page = queue.popleft()
           memory.remove(oldest_page)
           memory.append(page)
           queue.append(page)

   return miss
```

```
reference_string =[1,3,0,3,5,6]
frame_size = 3

miss = fifo_page_replacement(reference_string, frame_size)
print('Total no. of miss = ', miss)
```

Code in Github

Output:



Code Working Principle:

1. Start by initializing an empty queue to hold the pages in the order they are accessed.
2. Also prepare a structure (like a list or set) to track the current pages in memory.
3. Initialize a counter to track how many page faults occur.
4. For each page in the reference string:
    - If it’s not already in memory, count it as a page fault.

- f the memory has reached its limit:
    - Remove the oldest page (the one that entered first).
- Add the new page to both the queue and the memory tracker.
- If the page is already present, skip to the next one.

5. At the end, return or display the total number of page faults.

Conclusion:

The FIFO (First-In, First-Out) page replacement method removes the page that has been in memory the longest when new space is needed. Its straightforward approach makes it easy to implement, though it may not always provide optimal performance when compared to more advanced strategies like LRU or Optimal.