Ex. No.: 11(a) Date: 15/04/25

NAME:SOMILA ROLLNO:231901052

### FIFO PAGE REPLACEMENT

#### Aim:

To find out the number of page faults that occur using First-in First-out (FIFO) page replacement technique.

#### **Algorithm:**

- 1. Declare the size with respect to page length
- 2. Check the need of replacement from the page to memory
- 3. Check the need of replacement from old page to new page in memory
- 4. Form a queue to hold all pages
- 5. Insert the page require memory into the queue
- 6. Check for bad replacement and page fault
- 7. Get the number of processes to be inserted
- 8. Display the values

### **Program Code:**

```
#include
<stdio.h>
#define MAX
100 int main() {

int refStr[MAX], frames[MAX];

int n, frameSize, i, j, k;

int pageFaults = 0, pointer = 0, found;

printf("Enter the size of reference string: ");

scanf("%d", &n);
```

```
for (i = 0; i < n; i++) {
  printf("Enter [ %2d] : ", i + 1);
  scanf("%d", &refStr[i]);
}
// Input number of frames
printf("Enter page frame size : ");
scanf("%d", &frameSize);
// Initialize frames to -1
for (i = 0; i < frameSize; i++)
  frames[i] = -1;
// Process reference string
for (i = 0; i < n; i++) {
  found = 0;
```

```
// Check if page is already in frames
                                                for (j = 0; j < frameSize; j++) {
       if (frames[j] == refStr[i]) {
          found = 1;
          break;
       }
    if (!found) {
       // Page fault, replace oldest (FIFO)
       frames[pointer] = refStr[i];
pointer = (pointer + 1) % frameSize;
       pageFaults++;
       // Print frame content
       printf("%d -> ", refStr[i]);
       for (k = 0; k < frameSize; k++) {
```

```
if (frames[k] != -1)
            printf("%d", frames[k]);
          else
            printf("- ");
        }
       printf("\n");
} else {
       printf("%d -> No Page Fault\n", refStr[i]);
     }
  }
  // Final result
  printf("Total page faults: %d\n", pageFaults);
return 0;
}
Sample Output:
[root@localhost student]# python fifo.py
Enter the size of reference string: 20
Enter [ 1]: 7
```

```
Enter [2]:0
```

## Enter page frame size: 3

$$1 -> 701$$

$$2 -> 201$$

## 0 -> No Page Fault

$$3.1 \quad 0 \to 2$$

30

$$3.0 \quad 2 -> 4$$

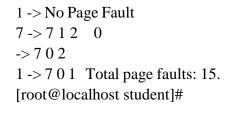
20

$$3 \rightarrow 423 \quad 0 \rightarrow 023$$

-> 0 1 3

$$2 \rightarrow 012$$

0 -> No Page Fault



# **Result:**

Thus the algorithm is executed successfully.