



**Mawlana Bhashani Science and Technology University**

## **Lab-Report**

Report No: 11

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance: 16-09-2020

Date of Submission:

### **Submitted by**

Name: Jannatul Ferdush Dhina

ID:IT-18012

3<sup>rd</sup> year 1<sup>st</sup> semester

Session: 2017-2018

Dept. of ICT

MBSTU.

### **Submitted To**

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU.

### Experiment no: 11

**Experiment name:** Implementation of FIFO page replacement algorithm.

#### Theory:

The first-in, first-out (**FIFO**) **page replacement algorithm** is a low-overhead **algorithm** that requires little bookkeeping on the part of the operating system. ... In simple words, on a page fault, the frame that has been in memory the longest is replaced.

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 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														

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.

#### Working Process:

Code in c is:

```
#include<stdio.h>

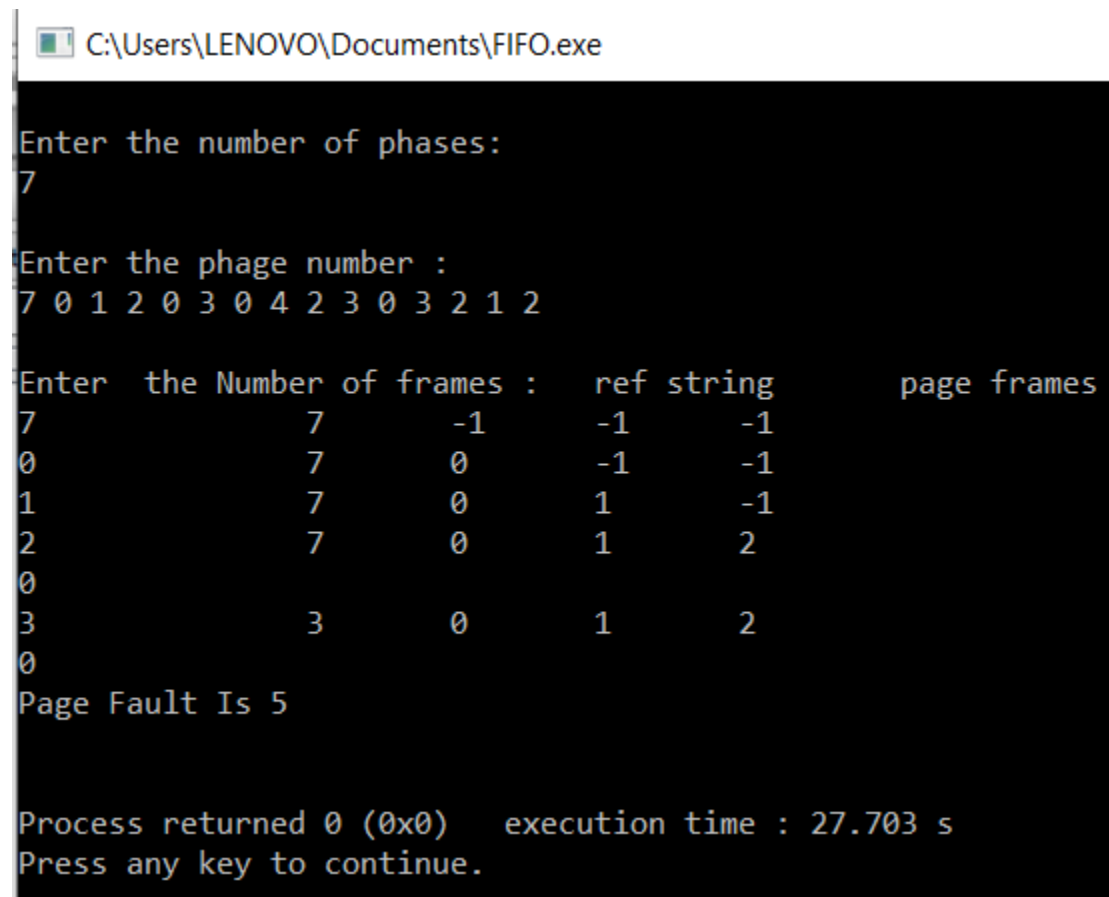
int main()
{
    int i,j,n,a[50],frame[10],no,k,avail,count=0;
    printf("\nEnter the number of phases:\n");
    scanf("%d",&n);
    printf("\nEnter the phage number :\n");
    for(i=1; i<=n; i++)
        scanf("%d",&a[i]);
    printf("\nEnter the Number of frames :");
    scanf("%d",&no);
    for(i=0; i<no; i++)
        frame[i]= -1;
    j=0;
    printf("\tref string\t page frames\n");
    for(i=1; i<=n; i++)
    {
        printf("%d\t\t",a[i]);
        avail=0;
        for(k=0; k<no; k++)
            if(frame[k]==a[i])
                avail=1;
        if (avail==0)
        {
            frame[j]=a[i];
            j=(j+1)%no;
            count++;
        }
    }
}
```

```

        for(k=0; k<no; k++)
            printf("%d\t",frame[k]);
    }
    printf("\n");
}
printf("Page Fault Is %d\n\n",count);
return 0;
}

```

### Output Sample:



```

C:\Users\LENOVO\Documents\FIFO.exe

Enter the number of phases:
7

Enter the phage number :
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2

Enter the Number of frames :
7

ref string      page frames
7              -1      -1
0              0       -1
1              0       -1
2              0        2
0
3              0        2
0

Page Fault Is 5

Process returned 0 (0x0)   execution time : 27.703 s
Press any key to continue.

```

**Discussion:**

To learn FIFO page replacement algorithm, This experiment helped us a lot.