Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 11

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance:

Date of Submission: 09/09/2020

Submitted by

Name: Ashikur Rahman Miran

ID:IT-18014

3rd year 1st 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.

Objective:

- i. What is FIFO page replacement algorithm?
- ii. Implementation of FIFO page replacement algorithm.

Theory:

FIFO is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue. When a new page come and this page is not in the queue then it replaces the oldest element and it called page fault. But when the new page is in the queue the it did not do anything and it called hit.

Implementation:

1. Take Input of frame size, page number and pages.
2. Then iterate over the pages and check.
3. If new page is present then increment hit.
4. If not then replace the longest page with new one and increment fault.

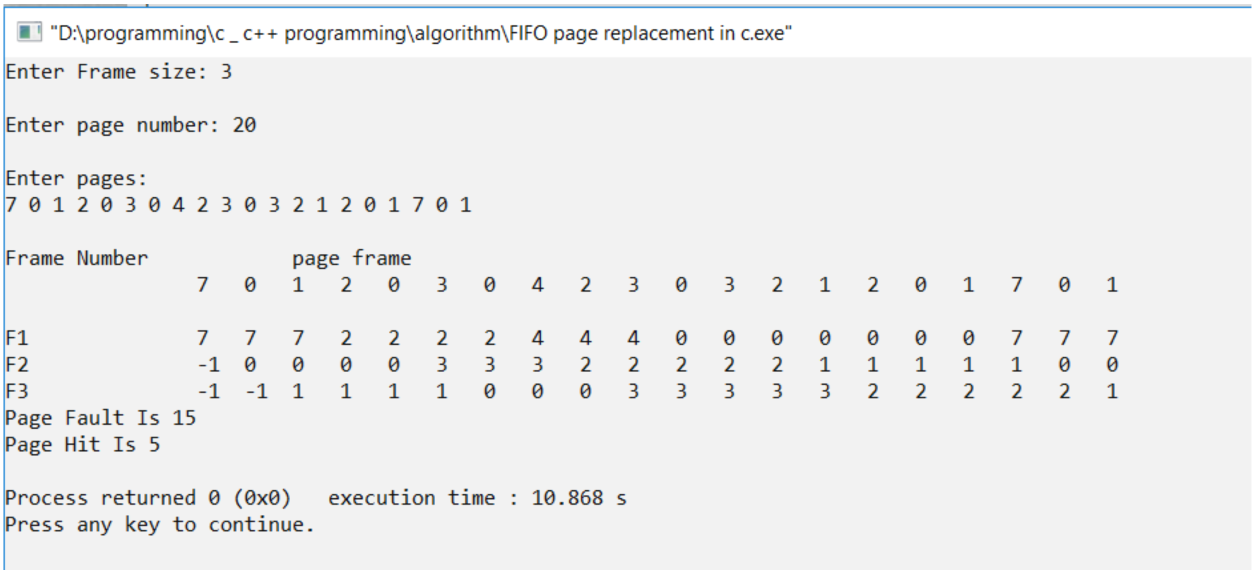
Example:

Frame	Page frame																			
	7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
F1	7	7	7	2	2	2	2	4	4	4	0	0	0	0	0	0	0	7	7	7
F2	-1	0	0	0	0	3	3	3	2	2	2	2	2	1	1	1	1	1	0	0
F3	-1	-1	1	1	1	1	0	0	0	3	3	3	3	3	2	2	2	2	2	1
	M	M	M	M	H	M	M	M	M	M	M	H	H	M	M	H	H	M	M	M

Source code:

```
#include<stdio.h>
int main()
{
    int i,j,fr_size,pno,fault=0,hit=0,pages[1000],frame[10],k,avail,res[50][50];
    printf("Enter Frame size: ");
    scanf("%d",&fr_size);
    printf("\nEnter page number: ");
    scanf("%d",&pno);
    printf("\nEnter pages: \n");
    for(i=0; i<pno; i++)
        scanf("%d",&pages[i]);
    for(i=0; i<fr_size; i++)
        frame[i]= -1;
    j=0;
    printf("\nFrame Number\t\tpage frame\n");
    printf("  \t\t");
    for(i=0; i<pno; i++)
    {
        printf("%d   ",pages[i]);
        avail=0;
        for(k=0; k<fr_size; k++)
        {
            if(frame[k]==pages[i])
            {
                hit++;
                avail=1;
            }
        }
        if (avail==0)
        {
            frame[j]=pages[i];
            j=(j+1)%fr_size;
            fault++;
            for(k=0; k<fr_size; k++)
                res[i][k]=frame[k];
         }
        else
        {
        for(k=0;k<fr_size;k++)
            res[i][k]=frame[k];
        }
    }
    printf("\n\n");
    for(i=0; i<fr_size; i++)
    {
        printf("F%d\t\t",i+1);
        for(j=0; j<pno; j++)
        {
            if(res[j][i]==-1)
                printf("%d  ",res[j][i]);
            else
               printf("%d   ",res[j][i]);
       }
       printf("\n");
    }
    printf("Page Fault Is %d\n",fault);
    printf("Page Hit Is %d\n",hit);
    return 0;
}
```

Output:



"D:\programming\c _ c++ programming\algorithm\FIFO page replacement in c.exe"

```
Enter Frame size: 3

Enter page number: 20

Enter pages:
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Frame Number            page frame
                7   0   1   2   0   3   0   4   2   3   0   3   2   1   2   0   1   7   0   1

F1              7   7   7   2   2   2   2   4   4   4   0   0   0   0   0   0   0   7   7   7
F2              -1  0   0   0   0   3   3   3   2   2   2   2   2   1   1   1   1   1   0   0
F3              -1  -1  1   1   1   1   0   0   0   3   3   3   3   3   2   2   2   2   2   1
Page Fault Is 15
Page Hit Is 5

Process returned 0 (0x0)   execution time : 10.868 s
Press any key to continue.
```

Conclusion:

In this lab I learn the FIFO algorithm and how to implement it in c language.

I also add source code and output of the FIFO algorithm.