

## 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: Shakhera khanom

ID: IT-18033

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.

### **Objectives:**

- What is FIFO page replacement Algorithm?
- How to implement in c?

### **❖ FIFO page replacement Algorithm**

- Round Robin is the preemptive process scheduling algorithm
- Each process is provided a fix time to execute, it is called a quantum.
- Once a process is execute for a given time period, it is preempted and other process executes for a given time period.

**Aim:** To implement page replacement algorithms FIFO (First In First Out) algorithm .

### **Description:**

Page replacement is basic to demand paging. It completes the separation between logical memory and physical memory. With this mechanism, an enormous virtual memory can be provided for programmers on a smaller physical memory. There are many different page-replacement algorithms. Every operating system probably has its own replacement scheme. A FIFO replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen. If the recent past is used as an approximation of the near future, then the page that has not been used for the longest period of time can be replaced.

### **Algorithm:**

**Step 1:** Start the program

**Step 2:** Read the number of frames

**Step 3:** Read the number of pages

**Step 4:** Read the page numbers

**Step 5:** Initialize the values in frames to -1

**Step 6:** Allocate the pages in to frames in First in first out order.

**Step 7:** Display the number of page faults.

**Step 8:** Stop the program

### Corresponding Code:

```
#include<stdio.h>
int main()
{
    int n,i,j,k,page_fault=0,f,r_string[50],count=0,frame[10],temp;
    printf("\nEnter the length of reference string : ");
    scanf("%d",&n);
    printf("\nEnter the reference string :- ");
    for(i=0;i<n;i++)
    {
        scanf("%d",&r_string[i]);
    }
    printf("\nEnter the num of frames : ");
    scanf("%d",&f);

    for(i=0;i<f;i++)
    {
        frame[i]=-1;
    }
    printf(" The Page Replacement Process is :-- \n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<f;j++)
        {
            if(frame[j]==r_string[i])
                break;
        }
        if(j==f)
        {
            frame[count++]=r_string[i];
            page_fault++;
        }
    }
}
```

```

        for(k=0;k<f;k++)
            printf("\t%d",frame[k]);
        if(j==f)
            printf("\tPage Fault no.  %d",page_fault);
        printf("\n");
        if(count==f)
            count=0;
    }
    printf("\n The number of Page Faults using FIFO are : %d\n",page_fault);
    printf("\n The number of Page Hits using FIFO are : %d\n",(n-page_fault));
    return 0;
}

```

## Output:

```

shakhera@shakhera-HP-Notebook-PC: ~/Desktop
File Edit View Search Terminal Help
shakhera@shakhera-HP-Notebook-PC:~/Desktop$ ./FIFO

Enter the length of reference string : 20

Enter the reference string :- 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Enter the num of frames : 3
The Page Replacement Process is :--

```

7	-1	-1	Page Fault no.	1
7	0	-1	Page Fault no.	2
7	0	1	Page Fault no.	3
2	0	1	Page Fault no.	4
2	0	1		
2	3	1	Page Fault no.	5
2	3	0	Page Fault no.	6
4	3	0	Page Fault no.	7
4	2	0	Page Fault no.	8
4	2	3	Page Fault no.	9
0	2	3	Page Fault no.	10
0	2	3		
0	2	3		
0	1	3	Page Fault no.	11
0	1	2	Page Fault no.	12
0	1	2		
0	1	2		
7	1	2	Page Fault no.	13
7	0	2	Page Fault no.	14
7	0	1	Page Fault no.	15

```

The number of Page Faults using FIFO are : 15

The number of Page Hits using FIFO are : 5

```

**Discussion:**

This lab helps to learn First In First Out (FIFO) algorithm. This is the simplest replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the older 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.