



Mawlana Bhashani Science and Technology University

Lab-Report

Lab Report No: 11

Lab Report Name: Implementation of FIFO page replacement algorithm.

Course code: ICT-3110

Course title: Operating System Lab

Date of Performance: 20-09-2020

Date of Submission:

Submitted by

Name: Md Masud Rana

ID: IT-18002

3rd year 1st semester

Session: 2017-18

Dept. of ICT.

Submitted To

Nazrul Islam

Assistant Professor

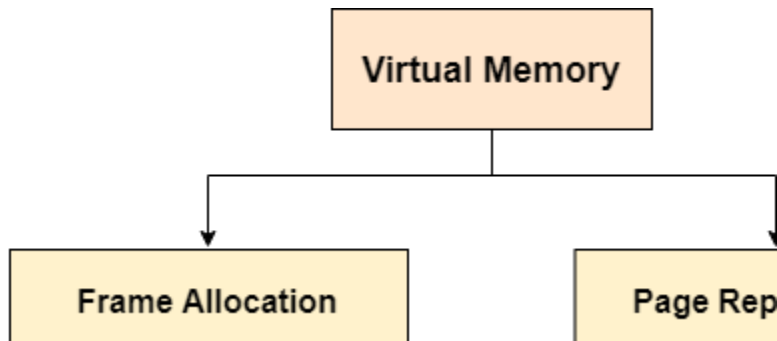
Dept. of ICT,

MBSTU.

Experiment No: 11

Experiment name: Implementation of FIFO page replacement algorithm .

Theory: The page replacement algorithm decides which memory page is to be replaced. The process of replacement is sometimes called swap out or write to disk. Page replacement is done when the requested page is not found in the main memory (page fault).



There are two main aspects of virtual memory, Frame allocation and Page Replacement. It is very important to have the optimal frame allocation and page replacement algorithm. Frame allocation is all about how many frames are to be allocated to the process while the page replacement is all about determining the page number which needs to be replaced in order to make space for the requested page.

Code-

robin.c	×	fifo.c
<pre>#include<stdio.h> int main() { int reference_string[10], page_faults = 0, m, n, s, pages, frames; printf("\n Number of Pages:\t"); scanf("%d", &pages); printf("\nEnter values of Reference String:\n"); for(m = 0; m < pages; m++) { printf(" Num. %d:\t", m + 1); scanf("%d", &reference_string[m]); } printf("\n Frames:\t"); { scanf("%d", &frames); } int temp[frames]; for(m = 0; m < frames; m++) { temp[m] = -1; } for(m = 0; m < pages; m++) { s = 0; for(n = 0; n < frames; n++) { if(reference_string[m] == temp[n]) { s++; page_faults--; } } } }</pre>		

```

    }
    page_faults++;
    if((page_faults <= frames) && (s == 0))
    {
        temp[m] = reference_string[m];
    }
    else if(s == 0)
    {
        temp[(page_faults - 1) % frames] = reference_string[m];
    }
    printf("\n");
    for(n = 0; n < frames; n++)
    {
        printf("%d\t", temp[n]);
    }
}
printf("\nTotal Page Faults:\t%d\n", page_faults);
return 0;
}

```

Output:

The screenshot shows a terminal window titled "masud@masud-VirtualBox: ~/program". The user enters the following commands:

```

masud@masud-VirtualBox:~/program$ gcc fifo.c -o fifo.out
masud@masud-VirtualBox:~/program$ ./fifo.out

```

The program output is as follows:

```

Number of Pages:      10

Enter values of Reference String:
Num. 1:      3
Num. 2:      2
Num. 3:      5
Num. 4:      0
Num. 5:      7
Num. 6:      4
Num. 7:      2
Num. 8:      2
Num. 9:      1
Num. 10:     5

Frames:      3

3      -1      -1
3      2      -1
3      2      5
0      2      5
0      7      5
0      7      4
2      7      4
2      7      4
2      1      4
2      1      5
Total Page Faults:      9
masud@masud-VirtualBox:~/program$

```

Discussion: In this lab we learn 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. For our better understanding we have used c language.