Mawlana Bhashani Science and Technology University

Lab-Report

Report No: 11

Course code: ICT-3110

Course title:Operating Systems Lab

Date of Performance:15-09-2020

Date of Submission:19-09-2020

Submitted by

Name: Sabikun Nahar Piya

ID:IT-18020

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

Theory:

FIFO Page Replacement technique is one of the simplest one to implement amongst other page replacement algorithms. It maintains a queue to keep a track of all the pages in memory. When a page needs to be replaced, page in the front of the queue (oldest page) is selected for removal.The FIFO page replacement technique is not implemented in operating systems nowadays, as it is not suitable for real-time systems.

Implementation:

Step1: Start traversing the pages.

i) If set holds less pages than capacity.

a) Insert page into the set one by one until

the size of set reaches capacity or all

page requests are processed.

b) Simultaneously maintain the pages in the

queue to perform FIFO.

c) Increment page fault

ii) Else

If current page is present in set, do nothing.

Else

a) Remove the first page from the queue

as it was the first to be entered in

the memory.

b) Replace the first page in the queue with

the current page in the string.

c) Store current page in the queue.

d) Increment page faults.

step2: Return page faults.

Working Process:

Code for FIFO base replacement algorithm:

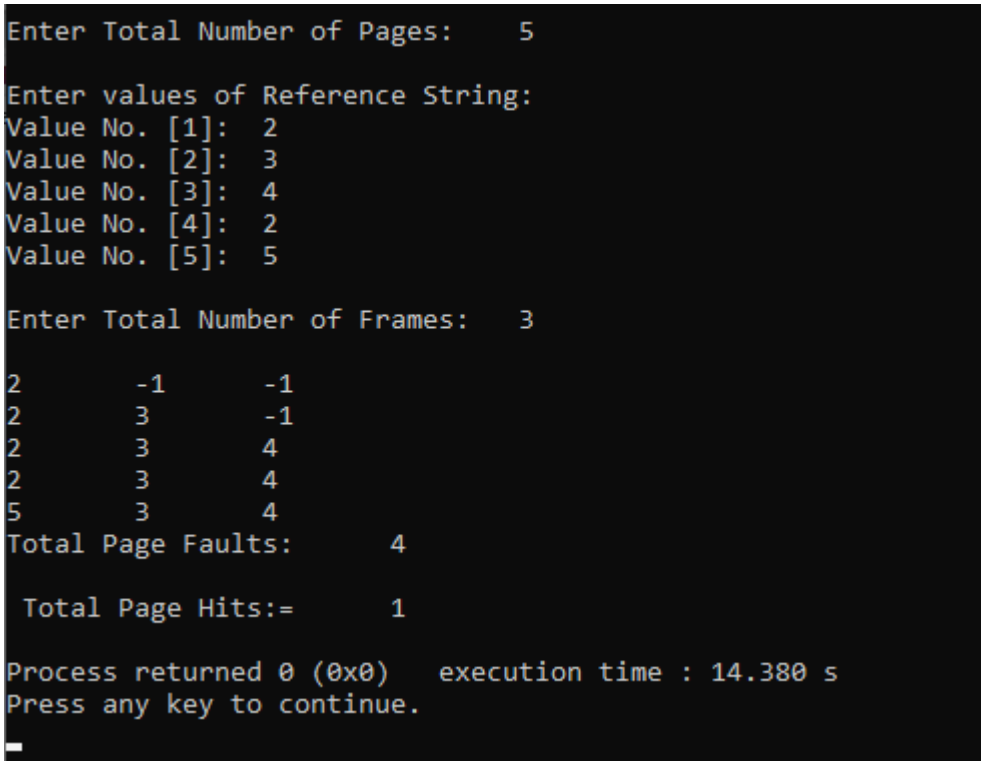
```
#include<stdio.h>

int main()

{

int reference_string[10],page_hits=0, page_faults = 0;

int temp[10],m, n, s, pages, frames;
```

```
printf("\nEnter Total Number of Pages:\t");
scanf("%d", &pages);
printf("\nEnter values of Reference String:\n");
for(m = 0; m < pages; m++)
{
printf("Value No. [%d]:\t", m + 1);
scanf("%d", &reference_string[m]);
}
printf("\nEnter Total Number of Frames:\t");
scanf("%d", &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_hits++;
page_faults--;
}
}
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);

printf("\n Total Page Hits:=\t%d\n",page_hits);

return 0;

}
```

Output:



Discussion:

This is the simplest page replacement algorithm.It is easy to implement and I don’t face any problem whenever Iimplementing the algorithm.I think this is the best replacement algorithm in opertating system.