EXPERIMENT 7

Aim: To implement First In First Out page replacement algorithm

Theory: First-In-First-Out (FIFO) Replacement. On a page fault, the frame that has been in memory the longest is replaced. FIFO is not a stack algorithm. In certain cases, the number of page faults can actually increase when more frames are allocated to the process.

This is the simplest page replacement algorithm. In this algorithm, operating system keeps track of all pages in the memory in a queue, 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.

Algorithm:

1- 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.

2. Return page faults.

CODE:

#include<stdio.h>

int main()

{

int reference\_string[10], page\_faults = 0, 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);

}

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--;

}

}

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:

Enter Total Number of Pages: 6

Enter values of Reference String:

Value No. [1]: 2

Value No. [2]: 4

Value No. [3]: 6

Value No. [4]: 8

Value No. [5]: 10

Value No. [6]: 12

Enter Total Number of Frames: 3

2 -1 -1

2 4 -1

2 4 6

8 4 6

8 10 6

8 10 12

Total Page Faults: 6

CONCLUSION: Hence we have implemented and studied FIFO page replacement algorithm.