**Lab no:4  Date:**

**Title : Write a program to simulate FIFO page replacement algorithm.**

**First In First Out (FIFO) – In this page replacement algorithm, the operating system keeps track of all pages in the memory in a queue. When a page needs to be replaced, page which is in the front of the queue is selected for removal.**

**Algorithm:**

Step 1: Start

Step 2: Input the page sequence

Step 3: Insert the pages directly if frame is empty

Step 4: Insert the pages by replacing the first page that arrived in that session

Step 5: Repeat step 2,3,4 until the input sequence is empty

Step 6: Stop

**Source code:**

|  |
| --- |
| // FIFO page replacement algorithm  #include <stdio.h>  int main()  {  int request[] = {1,2,3,2,1};  int pgFaults = 0; //initially pagefault is zero  int frames = 3;  int a, b, s, pages;  printf("fifo page replacement\n");  pages = sizeof(request)/sizeof(request[0]); //sizeof int is 4 bytes(for 5 pages 20 bytes)  printf("the total no of pages are %d\n", pages);  printf("request\t\t\tframe1\t\t\tframe2\t\t\tframe3");  int temp[frames];  for(a = 0; a < frames; a++)  {  temp[a] = -1;  }  for(a = 0; a < pages; a++)  {  s = 0;  for(b=0; b < frames; b++)  {  if(request[a] == temp[b])  {  s++;  pgFaults--;  }  }  pgFaults++;  if((pgFaults <= frames) && (s == 0))  {  temp[a] = request[a];  }  else if(s == 0)  {  temp[(pgFaults - 1) % frames] = request[a];  }    printf("\n%d\t\t\t",request[a]);  for(b=0; b < frames; b++)  {  if(temp[b] != -1){    printf(" %d\t\t\t", temp[b]);  }  else  printf(" \* \t\t\t");  }  }  printf("\nTotal number of page faults:\t%d\n", pgFaults);  return 0;  } |

**IDE: Dev-C++**

**Programming Language: C-programming**

**Output:**

