|  |  |
| --- | --- |
| **Ex.No:13.a** | **PAGE REPLACEMENT ALGORITHMS** |
| **FIFO** |

# AIM:

To write a C program for implementation of FIFO page replacement algorithm.

# ALGORITHM:

Step 1: Start the program.

Step 2: Declare the necessary variables. Step 3: Enter the number of frames.

Step 4: Enter the reference string end with zero.

Step 5: FIFO page replacement selects the page that has been in memory the longest time and when the page must be replaced the oldest page is chosen.

Step 6: When a page is brought into memory, it is inserted at the tail of the queue. Step 7: Initially all the three frames are empty.

Step 8: The page fault range increases as the no of allocated frames also increases. Step 9: Print the total number of page faults.

Step 10: Stop the program.

# PROGRAM:

#include<stdio.h> int main()

{

int i=0,j=0,k=0,i1=0,m,n,rs[30],flag=1,p[30]; system("clear");

printf("FIFO page replacement algorithm. \\n");

printf("enter the no. of frames:"); scanf("%d",&n);

printf("enter the reference string:"); while(1)

{

scanf("%d",&rs[i]); if(rs[i]==0)

break; i++;

}

m=i; for(j=0;j<n;j++) p[j]=0;

for(i=0;i<m;i++)

{

flag=1; for(j=0;j<n;j++) if(p[j]==rs[i])

{

printf("data already in page \n");

flag=0; break;

}

if(flag==1)

{

p[i1]=rs[i]; i1++;

k++;

if(i1==n) i1=0;

for(j=0;j<n;j++)

{

printf("\n page %d:%d",j+1,p[j]); if(p[j]==rs[i])

printf("\*");

}

printf("\n\n");

}

}

printf("total no page faults=%d",k);

}