PROBLEM:

Q: 22

Program to implement FIFO page replacement algorithm .First generate a random –page-reference string where page ranges from 0 to 9.

We need to apply random page-reference string to each algorithm , and the we need to record the number of page fault incurred by each algorithm .

In this problem in order to generate 0-9 pages we should use page –replacement-algorithm .. And we should use page demand so that we can generate 1 -7 number of frame pages ..

CODE:

#include<stdio.h>

#include<string.h>

#include<conio.h>

#include<stdlib.h>

#include<time.h>

int random\_check=0;

ypedef long long ll;

void inputvalues(ll\*virtual\_pages,ll total\_pages, ll total\_frames);

int pagehitting(ll \*frames\_list, ll total\_frames , ll goals);

{

for(int i=0; i<total\_frames; i++)

{

if(frames\_list[i]==goal)

return 1;

}

return 0;

}

void genertepages(ll \*virtual\_pages, ll total\_pages)

{

for(int i=0; i<total\_pages; i++)

virtual\_pages[i]==(rand()%10)//for generation of random 0-9 pages

}

int main(int m, const , char\*\*n)

{

srand(time(null));//internal clock to control choice of seed

ll total\_pages;

ll\*virtual\_pages;

if(m==1)

{

printf("enter number of virtual pages");

scanf("%lld",&total\_pages);

virtual\_pages=(ll\*)malloc(sizeof(ll))\*total\_pages);

int generate\_pages(virtual\_pages,total\_pages);

random\_check=1;

}

else

{

total\_page=m-1;

virtual\_pages=(ll\*)malloc(sizeof(ll)\*total\_pages);

int index=0;

for(int i=1;i<m;i++)

virtual\_pages[index++]=atoll(n[i]);

}

ll total\_frames;

printf("enter total number of frames");

scanf("%lld,&total\_frames");

inputs(virtual\_pages, total\_pages, total\_frames);

ll \*frames\_list=(ll\*)malloc(total\_frames\*sizeof(ll));

memoryset(frames\_list,-1,total\_frames\*,sizeof(ll));

ll count\_page\_hitting;

count\_page\_fault;

count\_page\_hitting=count\_page\_fault=0;

ll last=-1;

for(int i=0;i<total\_pages;i++)

{

if(get pagehit(frames\_list,total\_frames,virtual\_pages[i]))

count\_page\_hitting++;

else

{

frames\_list[++last%total\_frames]=virtual\_pages[i];

count\_page\_fault++;

}

}

printf("number of page hits are%lld\n",count\_page\_hitting);

printf("number of page faults are %lld\n",count\_page\_fault);

return 0;

}

void inputs(ll\* virtual\_pages, ll total\_pages, ll total\_frames)

{

printf("entered data as follows");

printf("number of pages : %lld\n", total\_pages);

if(random\_check)

printf("random");

printf("virtual pages");

for(int i=0;i<total\_pages;i++){

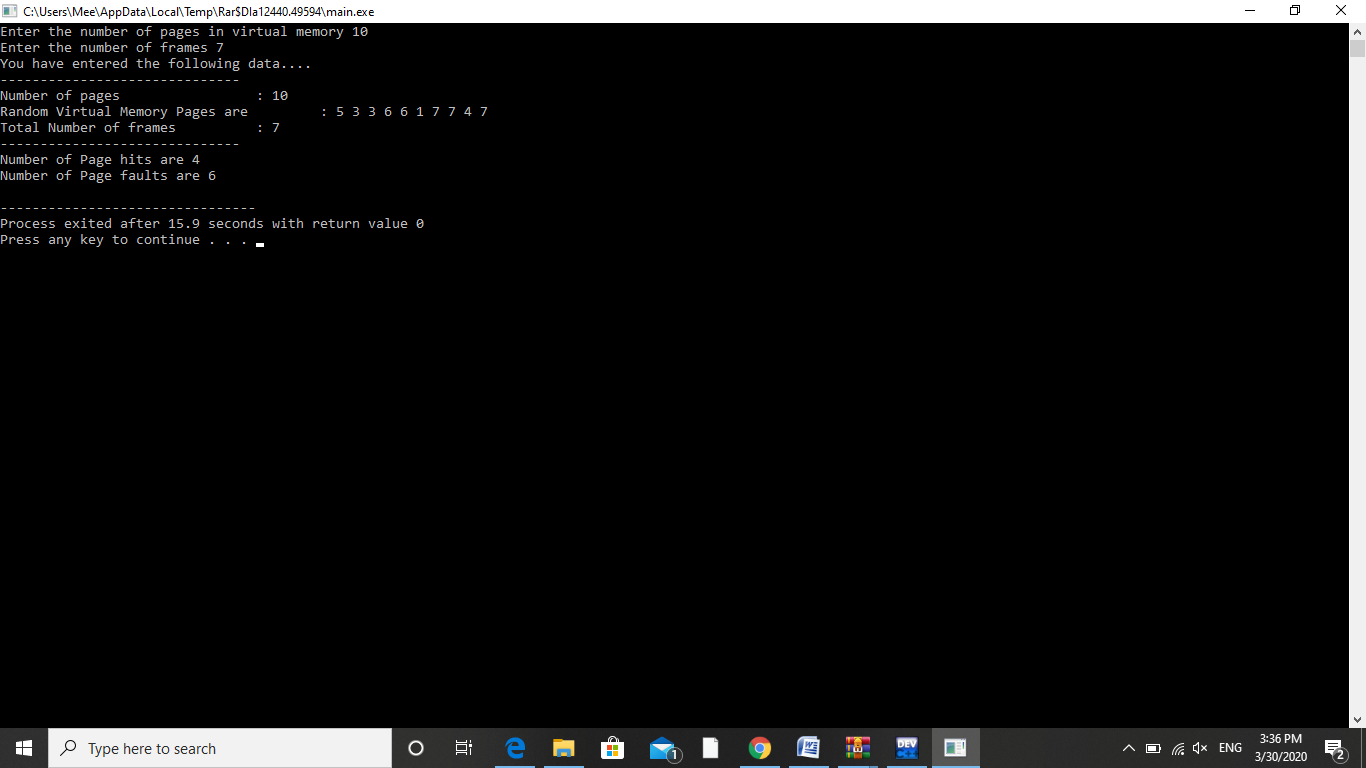
printf("%dlld",virtual\_pages[i]);

printf("number of frames ",total\_frames);

}

}

OUTPUT:



ALGORITHM:

1.Declare necessary variables such as random\_check to check the value, typedef variable.

2.Create input values function to input the values..

3.Use for loop to check the number of frames..

4.Create a generate pages function to generate pages using random function..

5.In main function declare srand() function and declare variables

Then use if statetment to check the values..

6.Then print values of total\_pages and virtual\_pages ..

7.Then use memory set fun to create frame\_list ….Then print all count of frames..

8.Create input function and give inputs..

9.Then print the entered values…and use srand() fun to create pages from 0-7 frames..

10.Print the info we need..

CONCEPTS USED:

FIFO algorithm takes value into queue and gives the first entered input as first oupt…This basic approach helps us to create paging using FIFO page replacement algorithm..

In this algorithm the operating system keeps track of all pages in memory in queue , the oldest page in the front of the queue ...when a page needs to be replaced page in the front of the queue is selected for removal..

Page demand in order to keep track of pages in order..

PURPOSE OF USE:

This project helps us to understand the FIFO algorithm..

This project gives us brief idea about page replacement algorithm and page demand..

We would be able to code for page replacement. a

TEST CASES:

1.Random number generation..

Using Rand() fun helps us to generate pages ..

To check generated pages use Random\_check flag..

Returns 1 for generation..

Returns 0 for not generated..

2.Check generated pages..

Use srand() fun to create frames…then random\_check returns 1 if frames are generated..

It returns 0 if not generated..

3.Page no> 9..

Then the value get mod by 10 and returns the value between 0-9..

4.Page no< 9..

Then the value return as it as mentioned...