**ASHNA .V**

**230701042**

**Ex. No.: 11a)**

**FIFO PAGE REPLACEMENT**

**Aim:**

To find out the number of page faults that occur using First-in First-out(FIFO) page replacement technique.

def fifo():

global a,n,m

f = -1

page\_faults = 0

page = []

for i in range(m):

page.append(-1)

for i in range(n):

flag = 0

for j in range(m):

if(page[j] == a[i]):

flag = 1

break

if flag == 0:

f=(f+1)%m

page[f] = a[i]

page\_faults+=1

print "\n%d ->" % (a[i]),

for j in range(m):

if page[j] != -1:

print page[j],

else:

print "-",

else:

print "\n%d -> No Page Fault" % (a[i]),

print "\n Total page faults : %d." % (page\_faults)

a = []

n = input("\n Enter the size of reference string : ")

for i in range(n):

a.append(input(" Enter [%2d] : " % (i+1))) m = input("\n Enter page frame size : ") fifo()

**Output:**

[root@localhost student]# python fifo.py

Enter the size of reference string : 20

Enter [ 1] : 7

Enter [ 2] : 0

Enter [ 3] : 1

Enter [ 4] : 2

Enter [ 5] : 0

Enter [ 6] : 3

Enter [ 7] : 0

Enter [ 8] : 4

Enter [ 9] : 2

Enter [10] : 3

Enter [11] : 0

Enter [12] : 3

Enter [13] : 2

Enter [14] : 1

Enter [15] : 2

Enter [16] : 0

Enter [17] : 1

Enter [18] : 7

Enter [19] : 0

Enter [20] : 1

Enter page frame size : 3

7 -> 7 - -

0 -> 7 0 -

1 -> 7 0 1

2 -> 2 0 1

0 -> No Page Fault

3 -> 2 3 1

0 -> 2 3 0

4 -> 4 3 0

2 -> 4 2 0

3 -> 4 2 3

0 -> 0 2 3

3 -> No Page Fault

2 -> No Page Fault

1 -> 0 1 3

2 -> 0 1 2

0 -> No Page Fault

1 -> No Page Fault

7 -> 7 1 2

0 -> 7 0 2

1 -> 7 0 1

Total page faults: 15. [root@localhost student]#

**Ex. No.: 11b)**

**LRU**

**Aim:**

To write a c program to implement LRU page replacement algorithm.

**Program Code:**

#include<stdio.h>

int findLRU(int time[], int n){

int i, minimum = time[0], pos = 0;

for(i = 1; i < n; ++i){

if(time[i] < minimum){

minimum = time[i];

pos = i;

}

}

return pos;

}

int main()

{

int no\_of\_frames, no\_of\_pages, frames[10], pages[30], counter = 0, time[10], flag1, flag2, i, j, pos, faults = 0;

printf("Enter number of frames: ");

scanf("%d", &no\_of\_frames);

printf("Enter number of pages: ");

scanf("%d", &no\_of\_pages);

printf("Enter reference string: ");

for(i = 0; i < no\_of\_pages; ++i){

scanf("%d", &pages[i]);

}

for(i = 0; i < no\_of\_frames; ++i){

frames[i] = -1;

}

for(i = 0; i < no\_of\_pages; ++i){

flag1 = flag2 = 0;

for(j = 0; j < no\_of\_frames; ++j){

if(frames[j] == pages[i]){

counter++;

time[j] = counter;

flag1 = flag2 = 1;

break;

}

}

if(flag1 == 0){

for(j = 0; j < no\_of\_frames; ++j){

if(frames[j] == -1){

counter++;

faults++;

frames[j] = pages[i]; time[j] = counter;

flag2 = 1;

break;

}

}

}

if(flag2 == 0){

pos = findLRU(time, no\_of\_frames); counter++;

faults++;

frames[pos] = pages[i];

time[pos] = counter;

}

printf("\n");

for(j = 0; j < no\_of\_frames; ++j){

printf("%d\t", frames[j]); }

}

printf("\n\nTotal Page Faults = %d", faults);

return 0;

}

**Output**

Enter number of frames: 3

Enter number of pages: 6

Enter reference string: 5 7 5 6 7 3

5 -1 -1

5 7 -1

5 7 -1

5 7 6

5 7 6

3 7 6

Total Page Faults = 4

**Ex. No.: 11c)**

**Optimal**

**Aim:**

To write a c program to implement Optimal page replacement algorithm.

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

int i,j,nof,nor,flag=0,ref[50],frm[50],pf=0,victim=-1; int recent[10],optcal[50],count=0;

int optvictim(); void main()

{ clrscr();

printf("\n OPTIMAL PAGE REPLACEMENT

ALGORITHN");

printf("\n..................... ");

............

printf("\nEnter the no.of frames");

scanf("%d",&nof);

printf("Enter the no.of reference string");

scanf("%d",&nor);

printf("Enter the reference string");

for(i=0;i<nor;i++)

scanf("%d",&ref[i]);

clrscr();

printf("\n OPTIMAL PAGE REPLACEMENT ALGORITHM");

printf("\n................................");

printf("\nThe given string");

printf("\n....................\n");

for(i=0;i<nor;i++)

printf("%4d",ref[i]);

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

{

frm[i]=-1;

optcal[i]=0;

}

for(i=0;i<10;i++)

recent[i]=0;

printf("\n");

for(i=0;i<nor;i++)

{

flag=0;

printf("\n\tref no %d ->\t",ref[i]);

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

{

if(frm[j]==ref[i])

{

flag=1;

break;

}

}

if(flag==0)

{

count++;

if(count<=nof)

victim++; else

victim=optvictim(i);

pf++;

frm[victim]=ref[i];

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

printf("%4d",frm[j]);

}

}

printf("\n Number of page faults: %d",pf); getch();

}

int optvictim(int index)

{

int i,j,temp,notfound; for(i=0;i<nof;i++)

{

notfound=1;

for(j=index;j<nor;j++)

if(frm[i]==ref[j])

{

notfound=0;

optcal[i]=j;

break;

}

if(notfound==1) return i;

}

temp=optcal[0];

for(i=1;i<nof;i++)

if(temp<optcal[i])

temp=optcal[i];

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

if(frm[temp]==frm[i]) return i;

return 0;

}

****