

Write a C program for FIFO P.P.
(CRV P.R, Optimal P.R)

#include <stdio.h>

int n, nf;

int m[100];

int b[50];

int int = 0;

int i, j, k;

int pagefaultcnt = 0;

void getData()

{

printf("Input length of page

reference sequence: ");

scanf("%d", &n);

printf("Input the page reference sequence

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

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

printf("Input no. of frames: ");

scanf("%d", &nf);

}

void initialize()

{

pagefaultcnt = 0;

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

b[i] = 9999;

int isHit (int data)

{

hit = 0;

```
for (j=0; j < nf; j++)
```

```
if (p[j] == data)
```

```
hit = 1;  
break;
```

{

}

```
return hit;
```

}

```
int getHitIndex (int data)
```

{

```
int hit mid;
```

```
for (k=0; k < nf; k++)
```

{

```
if (p[k] == data)
```

{

```
hit mid = k;
```

```
break;
```

}

J

```
return hit mid;
```

}

```
void dispPages ()
```

{

```
for (k=0; k < nf; k++)
```

{

```
if (p[k] != 9999)
```

```
printf ("%d", p[k]);
```

Y

J

Date _____
Page _____

```
void dispFaultCnt()
{
    printf ("In total no. of page faults : %d", pgfaultcnt);
}

void tflo()
{
    initialize();
    for (i=0; i<n; i++)
    {
        printf ("In Mem[%d] = ", m[i]);
        if (isHit (m[i])) = 0)
            for (k=0; k < nf-1; k++)
                p[k] = p[k+1];
            p[k] = m[i];
            pgfaultcnt++;
            dispPages();
        }
    else
        printf ("No page fault");
}

void dispFaultCnt()
```

```
void Optimal ()
{
    initialize();
    int near[50];
    for (i=0; i<n; i++)
    {
        // Implementation of Optimal page replacement algorithm
    }
}
```

printf ("In %s %d, in (%i));

if (isHit[in[i]) == 20)

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

int pg = p[j];

int bound = 0;

for (k = i; k < n; k++)

if (pg == in[k])

near[j] = k;

found = 1;

break;

}

else

bound = 20;

}

if (!found)

near[j] = 9999;

}

int max = -9999;

int repindex;

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

if (near[j] > max)

max = near[j];

repindex = j;

↓

↓

Page

```
p[recIndex] = m[i];
pgFaultCount++;
dispPages();
else
    print("No page fault");
    dispPgFaultCount();
}

void run()
{
    initialize();
    int least[50];
    for (i=0; i<n; i++)
        print("in %d %d:", m[i]);
    if (isHit(m[i]) == 0)
        for (j=0; j<n; j++)
            if (pg == p[j])
                found = 1;
        for (k=j-1; k>=0; k--)
            if (pg == m[k])
                least[j] = k;
                found = 1;
                break;
}
```

```
else
    if (!found)
        least[j] = -9999;
    int min = 9999;
    int repindex;
    for (j = 0; j < cut; j++)
        if (least[j] < min)
            min = least[j];
            repindex = j;
    if (repindex == i)
        bgfaultcnt++;
    dispPages();
else
    printf("No page fault!");
    dispPageFault();
int main()
{
    int choice;
    while (1)
    {
```

Print ("In Page Replacement Algorithm")
In 1. Put data in - FIFO is Optimal
In 4. LRU is Six in puts your choice
Scan ("9. d", 4 choice);
switch (choice)
{

case 1:

getData();
break;

case 2:

fifo();
break;

case 3:

optimal();
break;

case 4:

lru();
break;

default:

return 0;
break;

Output

1. Put

2. FIT-O

3. Optim

4- LRU

Enter

Enter

Enter

Enter

Enter

Enter

for

Algorithms
simulations
your choice

Output

1. Putn data

2. FIT-O

3. Optimal

4. LRU

Putn your choice : 1

Putn length of page reference sequence : 5

Putn the page reference sequence :

14 25 34 16 17

Putn no. of frames : 4

Putn choice : 2

M01 14 : 14

M01 25 : 14 25

M01 34 : 14 25 34

M01 16 : 14 25 34 16

M01 17 : 25 34 16 17

Total no. of page faults : 5

Putn your choice : 3

M01 14 : 14

M01 25 : 25

M01 34 : 34

M01 16 : 16

M01 17 : 17

Total no. of page faults : 5

Putn your choice : 4

M01 14 : 14

M01 25 : 14 25

M01 34 : 14 25 34

M01 16 : 14 25 34 16

M01 17 : 17 25 34 16

```
C:\Users\Avani\Desktop\1bm21cs036\df\bin\Debug\df.exe

Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:1

Enter length of page reference sequence:4

Enter the page reference sequence:6
5
4
3

Enter no of frames:3

Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:2

For 6 : 6
For 5 : 6 5
For 4 : 6 5 4
For 3 : 5 4 3
Total no of page faults:4
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:3

For 6 : 6
For 5 : 5
For 4 : 4
For 3 : 3
Total no of page faults:4
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU

For 6 : 6
For 5 : 5
For 4 : 4
For 3 : 3
Total no of page faults:4
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:4

For 6 : 6
For 5 : 6 5
For 4 : 6 5 4
For 3 : 3 5 4
Total no of page faults:4
Page Replacement Algorithms
1.Enter data
2.FIFO
3.Optimal
4.LRU
5.Exit
Enter your choice:
```