Write a C program to simulate page replacement algorithms a) FIFO b) LRU c) LFU

FIFO PAGE REPLACEMENT ALGORITHM

#include <iostream>

#include <vector>

using namespace std;

int main() {

int i, j, k, f, pf = 0, count = 0, n;

vector<int> rs(25), m(10, -1);

cout << "\n Enter the length of reference string -- ";

cin >> n;

cout << "\n Enter the reference string -- ";

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

cin >> rs[i];

cout << "\n Enter no. of frames -- ";

cin >> f;

cout << "\n The Page Replacement Process is -- \n";

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

for (k = 0; k < f; k++) {

if (m[k] == rs[i])

break;

}

if (k == f) {

m[count++] = rs[i];

pf++;

}

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

cout << "\t" << m[j];

if (k == f)

cout << "\tPF No. " << pf;

cout << "\n";

if (count == f)

count = 0;

}

cout << "\n The number of Page Faults using FIFO are " << pf;

return 0;

}

INPUT

Enter the length of reference string – 20

Enter the reference string -- 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Enter no. of frames -- 3

OUTPUT

The Page Replacement Process is –

7 -1 -1 PF No. 1

7 0 -1 PF No. 2

7 0 1 PF No. 3

2 0 1 PF No. 4

2 0 1

2 3 1 PF No. 5

2 3 0 PF No. 6

4 3 0 PF No. 7

4 2 0 PF No. 8

4 2 3 PF No. 9

0 2 3 PF No. 10

0 2 3

0 2 3

0 1 3 PF No. 11

0 1 2 PF No. 12

0 1 2

0 1 2

7 1 2 PF No. 13

7 0 2 PF No. 14

7 0 1 PF No. 15

The number of Page Faults using FIFO are 15

**LRU PAGE REPLACEMENT ALGORITHM**

#include <iostream>

#include <vector>

using namespace std;

int main() {

int i, j, k, min, n, f, pf = 0, next = 1;

vector<int> rs(25), m(10), count(10), flag(25);

cout << "Enter the length of reference string -- ";

cin >> n;

cout << "Enter the reference string -- ";

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

cin >> rs[i];

flag[i] = 0;

}

cout << "Enter the number of frames -- ";

cin >> f;

for (i = 0; i < f; i++) {

count[i] = 0;

m[i] = -1;

}

cout << "\nThe Page Replacement process is -- \n";

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

for (j = 0; j < f; j++) {

if (m[j] == rs[i]) {

flag[i] = 1;

count[j] = next;

next++;

}

}

if (flag[i] == 0) {

if (i < f) {

m[i] = rs[i];

count[i] = next;

next++;

} else {

min = 0;

for (j = 1; j < f; j++)

if (count[min] > count[j])

min = j;

m[min] = rs[i];

count[min] = next;

next++;

}

pf++;

}

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

cout << m[j] << "\t";

if (flag[i] == 0)

cout << "PF No. -- " << pf;

cout << "\n";

}

cout << "\nThe number of page faults using LRU are " << pf;

return 0;

}

INPUT

Enter the length of reference string -- 20

Enter the reference string -- 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

Enter the number of frames -- 3

OUTPUT

The Page Replacement process is --

7 -1 -1 PF No. -- 1

7 0 -1 PF No. -- 2

7 0 1 PF No. -- 3

2 0 1 PF No. -- 4

2 0 1

2 0 3 PF No. -- 5

2 0 3

4 0 3 PF No. -- 6

4 0 2 PF No. -- 7

4 3 2 PF No. -- 8

0 3 2 PF No. -- 9

0 3 2

0 3 2

1 3 2 PF No. -- 10

1 3 2

1 0 2 PF No. -- 11

1 0 2

1 0 7 PF No. -- 12

1 0 7

1 0 7

The number of page faults using LRU are 12

LFU PAGE REPLACEMENT ALGORITHM

#include <iostream>

#include <vector>

using namespace std;

int main() {

int rs[50], i, j, k, m, f, cntr[20], a[20], min, pf = 0;

// clrscr(); // Not needed in C++

cout << "\nEnter number of page references -- ";

cin >> m;

cout << "\nEnter the reference string -- ";

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

cin >> rs[i];

cout << "\nEnter the available no. of frames -- ";

cin >> f;

for (i = 0; i < f; i++) {

cntr[i] = 0;

a[i] = -1;

}

cout << "\nThe Page Replacement Process is – \n";

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

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

if (rs[i] == a[j]) {

cntr[j]++;

break;

}

if (j == f) {

min = 0;

for (k = 1; k < f; k++)

if (cntr[k] < cntr[min])

min = k;

a[min] = rs[i];

cntr[min] = 1;

pf++;

}

cout << "\n";

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

cout << "\t" << a[j];

if (j == f)

cout << "\tPF No. " << pf;

}

cout << "\n\n Total number of page faults -- " << pf;

return 0;

}

INPUT

Enter number of page references --10

Enter the reference string -- 1 2 3 4 5 2 5 2 5 1 4 3

Enter the available no. of frames – 3

OUTPUT

The Page Replacement Process is –

1 -1 -1 PF No. 1

1 2 -1 PF No. 2

1 2 3 PF No. 3

4 2 3 PF No. 4

5 2 3 PF No. 5

5 2 3

5 2 3

5 2 1 PF No. 6

5 2 4 PF No. 7

5 2 3 PF No. 8

Total number of page faults -- 8