ASSIGNMENT 7

PROGRAM:

#include <iostream>

using namespace std;

bool isInFrame(int frame[], int frameSize, int page) {

for (int i = 0; i < frameSize; i++) {

if (frame[i] == page) return true;

}

return false;

}

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

int min = time[0], pos = 0;

for (int i = 1; i < frameSize; i++) {

if (time[i] < min) {

min = time[i];

pos = i;

}

}

return pos;

}

int findOptimal(int pages[], int frame[], int currentIndex, int frameSize, int pageCount) {

int index[10], flag;

for (int i = 0; i < frameSize; i++) {

flag = 0;

for (int j = currentIndex + 1; j < pageCount; j++) {

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

index[i] = j;

flag = 1;

break;

}

}

if (!flag) return i; // Not used again

}

int farthest = index[0], pos = 0;

for (int i = 1; i < frameSize; i++) {

if (index[i] > farthest) {

farthest = index[i];

pos = i;

}

}

return pos;

}

void printStats(const string& name, int pageFaults, int totalPages) {

int hits = totalPages - pageFaults;

float faultPercent = (pageFaults \* 100.0f) / totalPages;

float hitPercent = (hits \* 100.0f) / totalPages;

cout << "Total Page Faults (" << name << "): " << pageFaults << "\n";

cout << "Total Hits (" << name << "): " << hits << "\n";

cout << "Page Fault %: " << faultPercent << "%\n";

cout << "Hit %: " << hitPercent << "%\n\n";

}

void FCFS(int pages[], int pageCount, int frameSize) {

int frame[10], index = 0, pageFaults = 0;

for (int i = 0; i < frameSize; i++) frame[i] = -1;

cout << "\nFCFS:\n";

for (int i = 0; i < pageCount; i++) {

if (!isInFrame(frame, frameSize, pages[i])) {

frame[index] = pages[i];

index = (index + 1) % frameSize;

pageFaults++;

}

cout << "Page " << pages[i] << ": ";

for (int j = 0; j < frameSize; j++) {

if (frame[j] != -1) cout << frame[j] << " ";

else cout << "- ";

}

cout << endl;

}

printStats("FCFS", pageFaults, pageCount);

}

void LRU(int pages[], int pageCount, int frameSize) {

int frame[10], time[10], count = 0, pageFaults = 0;

for (int i = 0; i < frameSize; i++) {

frame[i] = -1;

time[i] = 0;

}

cout << "\nLRU:\n";

for (int i = 0; i < pageCount; i++) {

bool found = false;

for (int j = 0; j < frameSize; j++) {

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

count++;

time[j] = count;

found = true;

break;

}

}

if (!found) {

int pos = -1;

for (int j = 0; j < frameSize; j++) {

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

pos = j;

break;

}

}

if (pos == -1)

pos = findLRU(time, frameSize);

frame[pos] = pages[i];

count++;

time[pos] = count;

pageFaults++;

}

cout << "Page " << pages[i] << ": ";

for (int j = 0; j < frameSize; j++) {

if (frame[j] != -1) cout << frame[j] << " ";

else cout << "- ";

}

cout << endl;

}

printStats("LRU", pageFaults, pageCount);

}

void Optimal(int pages[], int pageCount, int frameSize) {

int frame[10], pageFaults = 0;

for (int i = 0; i < frameSize; i++) frame[i] = -1;

cout << "\nOptimal:\n";

for (int i = 0; i < pageCount; i++) {

if (!isInFrame(frame, frameSize, pages[i])) {

int pos = -1;

for (int j = 0; j < frameSize; j++) {

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

pos = j;

break;

}

}

if (pos == -1)

pos = findOptimal(pages, frame, i, frameSize, pageCount);

frame[pos] = pages[i];

pageFaults++;

}

cout << "Page " << pages[i] << ": ";

for (int j = 0; j < frameSize; j++) {

if (frame[j] != -1) cout << frame[j] << " ";

else cout << "- ";

}

cout << endl;

}

printStats("Optimal", pageFaults, pageCount);

}

int main() {

int pages[50], pageCount, frameSize;

int choice;

cout << "Enter number of pages: ";

cin >> pageCount;

cout << "Enter the page reference string:\n";

for (int i = 0; i < pageCount; i++) {

cout << "Page " << i + 1 << ": ";

cin >> pages[i];

}

cout << "Enter number of frames: ";

cin >> frameSize;

do {

cout << "\n--- MENU ---\n";

cout << "1. FCFS\n";

cout << "2. LRU\n";

cout << "3. Optimal\n";

cout << "4. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: FCFS(pages, pageCount, frameSize); break;

case 2: LRU(pages, pageCount, frameSize); break;

case 3: Optimal(pages, pageCount, frameSize); break;

case 4: cout << "Exiting...\n"; break;

default: cout << "Invalid choice! Please enter 1-4.\n";

}

} while (choice != 4);

return 0;

}

OUTPUT:

PS C:\Users\User\Desktop\OSL> g++ osl7.cpp

PS C:\Users\User\Desktop\OSL> .\a

Enter number of pages: 20

Enter the page reference string:

Page 1: 7

Page 2: 0

Page 3: 1

Page 4: 2

Page 5: 0

Page 6: 3

Page 7: 0

Page 8: 4

Page 9: 2

Page 10: 3

Page 11: 0

Page 12: 3

Page 13: 2

Page 14: 1

Page 15: 2

Page 16: 0

Page 17: 1

Page 18: 7

Page 19: 0

Page 20: 1

Enter number of frames: 4

--- MENU ---

1. FCFS

2. LRU

3. Optimal

4. Exit

Enter your choice: 1

FCFS:

Page 7: 7 - - -

Page 0: 7 0 - -

Page 1: 7 0 1 -

Page 2: 7 0 1 2

Page 0: 7 0 1 2

Page 3: 3 0 1 2

Page 0: 3 0 1 2

Page 4: 3 4 1 2

Page 2: 3 4 1 2

Page 3: 3 4 1 2

Page 0: 3 4 0 2

Page 3: 3 4 0 2

Page 2: 3 4 0 2

Page 1: 3 4 0 1

Page 2: 2 4 0 1

Page 0: 2 4 0 1

Page 1: 2 4 0 1

Page 7: 2 7 0 1

Page 0: 2 7 0 1

Page 1: 2 7 0 1

Total Page Faults (FCFS): 10

Total Hits (FCFS): 10

Page Fault %: 50%

Hit %: 50%

--- MENU ---

1. FCFS

2. LRU

3. Optimal

4. Exit

Enter your choice: 2

LRU:

Page 7: 7 - - -

Page 0: 7 0 - -

Page 1: 7 0 1 -

Page 2: 7 0 1 2

Page 0: 7 0 1 2

Page 3: 3 0 1 2

Page 0: 3 0 1 2

Page 4: 3 0 4 2

Page 2: 3 0 4 2

Page 3: 3 0 4 2

Page 0: 3 0 4 2

Page 3: 3 0 4 2

Page 2: 3 0 4 2

Page 1: 3 0 1 2

Page 2: 3 0 1 2

Page 0: 3 0 1 2

Page 1: 3 0 1 2

Page 7: 7 0 1 2

Page 0: 7 0 1 2

Page 1: 7 0 1 2

Total Page Faults (LRU): 8

Total Hits (LRU): 12

Page Fault %: 40%

Hit %: 60%

--- MENU ---

1. FCFS

2. LRU

3. Optimal

4. Exit

Enter your choice: 3

Optimal:

Page 7: 7 - - -

Page 0: 7 0 - -

Page 1: 7 0 1 -

Page 2: 7 0 1 2

Page 0: 7 0 1 2

Page 3: 3 0 1 2

Page 0: 3 0 1 2

Page 4: 3 0 4 2

Page 2: 3 0 4 2

Page 3: 3 0 4 2

Page 0: 3 0 4 2

Page 3: 3 0 4 2

Page 2: 3 0 4 2

Page 1: 1 0 4 2

Page 2: 1 0 4 2

Page 0: 1 0 4 2

Page 1: 1 0 4 2

Page 7: 1 0 7 2

Page 0: 1 0 7 2

Page 1: 1 0 7 2

Total Page Faults (Optimal): 8

Total Hits (Optimal): 12

Page Fault %: 40%

Hit %: 60%

--- MENU ---

1. FCFS

2. LRU

3. Optimal

4. Exit

Enter your choice: 4

Exiting...

PS C:\Users\User\Desktop\OSL>