Page Replacement

#include <stdio.h>

#include <stdlib.h>

#define MAX 100

int isInFrame(int frames[], int n, int page) {

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

if (frames[i] == page)

return 1;

return 0;

}

int fifo(int pages[], int n, int capacity) {

int frames[capacity], front = 0, count = 0;

for (int i = 0; i < capacity; i++)

frames[i] = -1;

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

if (!isInFrame(frames, capacity, pages[i])) {

frames[front] = pages[i];

front = (front + 1) % capacity;

count++;

}

}

return count;

}

int optimal(int pages[], int n, int capacity) {

int frames[capacity], count = 0;

for (int i = 0; i < capacity; i++)

frames[i] = -1;

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

if (!isInFrame(frames, capacity, pages[i])) {

int replace = -1, farthest = i + 1;

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

int k;

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

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

if (k > farthest) {

farthest = k;

replace = j;

}

break;

}

}

if (k == n) {

replace = j;

break;

}

}

if (replace == -1)

replace = 0;

frames[replace] = pages[i];

count++;

}

}

return count;

}

int lru(int pages[], int n, int capacity) {

int frames[capacity], recent[capacity], count = 0;

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

frames[i] = -1;

recent[i] = -1;

}

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

int found = 0;

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

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

recent[j] = i;

found = 1;

break;

}

}

if (!found) {

int lru\_index = 0;

for (int j = 1; j < capacity; j++) {

if (recent[j] < recent[lru\_index])

lru\_index = j;

}

frames[lru\_index] = pages[i];

recent[lru\_index] = i;

count++;

}

}

return count;

}

int main() {

int pages[MAX], n, capacity;

printf("Enter number of pages: ");

scanf("%d", &n);

printf("Enter page reference string:\n");

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

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

printf("Enter number of frames: ");

scanf("%d", &capacity);

printf("\nPage Faults:\n");

printf("FIFO: %d\n", fifo(pages, n, capacity));

printf("Optimal: %d\n", optimal(pages, n, capacity));

printf("LRU: %d\n", lru(pages, n, capacity));

return 0;

}

