Experiment Number: 10

Problem Statement: Write a program to calculate the number of page faults for a reference string for the following page replacement algorithms:

**a) FIFO**

**b) LRU**

**c) Optimal**

NAME: Manoj Dhanraj MuleROLLNO: 71

CLASS: IT-B BATCH: B3

DATE OF PERFORMANCE: 16/10/2024

#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 findOptimal(int pages[], int n, int frame[], int fsize, int index) { int i, j, pos = -1, farthest = index;

for (i = 0; i < fsize; i++) { int found = 0;

for (j = index; j < n; j++) { if (frame[i] == pages[j]) {

found = 1;

if (j > farthest) { farthest = j; pos = i;

}

break;

}

}

if (!found) return i;

}

return (pos == -1) ? 0 : pos;

}

void fifo(int pages[], int n, int frames[], int fsize) { int i, j = 0, pageFaults = 0, found; printf("\nFIFO Page Replacement:\n");

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

for (int k = 0; k < fsize; k++) { if (frames[k] == pages[i]) {

found = 1; break;

}

}

if (!found) {

frames[j] = pages[i]; j = (j + 1) % fsize; pageFaults++;

}

printf("Frame: ");

for (int k = 0; k < fsize; k++) printf("%d ", frames[k]); printf("\n");

}

printf("Total Page Faults (FIFO): %d\n", pageFaults);

}

void lru(int pages[], int n, int frames[], int fsize) { int time[fsize], pageFaults = 0, count = 0, found; printf("\nLRU Page Replacement:\n");

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

for (int j = 0; j < fsize; j++) { if (frames[j] == pages[i]) {

found = 1;

time[j] = ++count; break;

}

}

if (!found) {

int pos = findLRU(time, fsize); frames[pos] = pages[i]; time[pos] = ++count;

pageFaults++;

}

printf("Frame: ");

for (int k = 0; k < fsize; k++) printf("%d ", frames[k]); printf("\n");

}

printf("Total Page Faults (LRU): %d\n", pageFaults);

}

void optimal(int pages[], int n, int frames[], int fsize) { int pageFaults = 0, found;

printf("\nOptimal Page Replacement:\n"); for (int i = 0; i < n; i++) {

found = 0;

for (int j = 0; j < fsize; j++) { if (frames[j] == pages[i]) {

found = 1; break;

}

}

if (!found) {

int pos = findOptimal(pages, n, frames, fsize, i + 1); frames[pos] = pages[i];

pageFaults++;

}

printf("Frame: ");

for (int k = 0; k < fsize; k++) printf("%d ", frames[k]); printf("\n");

}

printf("Total Page Faults (Optimal): %d\n", pageFaults);

}

int main() { int n, fsize;

// Input the number of pages printf("Enter number of pages: "); scanf("%d", &n);

int pages[n];

// Input the page reference string printf("Enter the reference string: "); for (int i = 0; i < n; i++) {

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

}

// Input the number of frames printf("Enter number of frames: "); scanf("%d", &fsize);

int frames[fsize];

// Initialize the frames to -1 (indicating empty) for (int i = 0; i < fsize; i++) frames[i] = -1;

// FIFO Algorithm fifo(pages, n, frames, fsize);

// Reinitialize the frames for LRU

for (int i = 0; i < fsize; i++) frames[i] = -1;

// LRU Algorithm lru(pages, n, frames, fsize);

// Reinitialize the frames for Optimal

for (int i = 0; i < fsize; i++) frames[i] = -1;

// Optimal Algorithm optimal(pages, n, frames, fsize);

return 0;

}

**Output:**

**manoj@manoj-VirtualBox:~/oslab$ gcc lab10.c -o lab10**

**manoj@manoj-VirtualBox:~/oslab$ ./lab10**

**Enter number of pages: 8**

**Enter the reference string: 1 3 0 3 5 6 3 3**

**Enter number of frames: 3**

**FIFO Page Replacement:**

**Frame: 1 -1 -1**

**Frame: 1 3 -1**

**Frame: 1 3 0**

**Frame: 1 3 0**

**Frame: 5 3 0**

**Frame: 5 6 0**

**Frame: 5 6 3**

**Frame: 5 6 3**

**Total Page Faults (FIFO): 6**

**LRU Page Replacement:**

**Frame: 1 -1 -1**

**Frame: 1 -1 3**

**Frame: 0 -1 3**

**Frame: 0 -1 3**

**Frame: 5 -1 3**

**Frame: 5 -1 6**

**Frame: 3 -1 6**

**Frame: 3 -1 6**

**Total Page Faults (LRU): 6**

**Optimal Page Replacement:**

**Frame: 1 -1 -1**

**Frame: 3 -1 -1**

**Frame: 3 0 -1**

**Frame: 3 0 -1**

**Frame: 3 5 -1**

**Frame: 3 6 -1**

**Frame: 3 6 -1**

**Frame: 3 6 -1**

**Total Page Faults (Optimal): 5**

**manoj@manoj-VirtualBox:~/oslab$**

**Enter number of pages: 9**

**Enter the reference string: 2 3 4 2 1 3 7 3 1**

**Enter number of frames: 3**

**FIFO Page Replacement:**

**Frame: 2 -1 -1  
Frame: 2 3 -1  
Frame: 2 3 4  
Frame: 2 3 4  
Frame: 3 4 1  
Frame: 3 4 1  
Frame: 4 1 7  
Frame: 1 7 3  
Frame: 1 7 3**

**Total Page Faults (FIFO): 6**

**LRU Page Replacement:**

**Frame: 2 -1 -1  
Frame: 2 3 -1  
Frame: 2 3 4  
Frame: 2 3 4  
Frame: 2 4 1  
Frame: 4 1 3  
Frame: 1 3 7  
Frame: 1 3 7  
Frame: 7 3 1**

**Total Page Faults (LRU): 7**

**Optimal Page Replacement:**

**Frame: 2 -1 -1  
Frame: 2 3 -1  
Frame: 2 3 4  
Frame: 2 3 4  
Frame: 2 3 1  
Frame: 3 1 4  
Frame: 3 1 7  
Frame: 3 1 7  
Frame: 3 1 7**

**Total Page Faults (Optimal): 5**