Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_12\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **01** | Implement the FIFO and LRU policies described above in C language. |
| **02** | Execute both programs for the same set of reference strings. What difference did you observe? Comment. |
|  |  |
|  |  |

Submitted On:

15-06-2023

(Date: DD/MM/YY)

**Task 01:** Implement the FIFO and LRU policies described above in C language.

**Solution**:

**FIFO:**

#include <stdio.h>

#include <stdlib.h>

#define MAX\_PAGES 100

int main() {

int pages[MAX\_PAGES], n, capacity, i, j, page\_faults = 0, front = 0, rear = -1;

printf("Enter the number of pages: ");

scanf("%d", &n);

printf("Enter the page reference string: ");

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

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

printf("Enter the capacity of memory: ");

scanf("%d", &capacity);

int memory[capacity];

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

memory[i] = -1;}

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

int page\_found = 0;

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

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

page\_found = 1;

break;}}

if (!page\_found) {

page\_faults++;

if (rear == capacity - 1) {

rear = -1;}

rear++;

memory[rear] = pages[i];}}

printf("Number of page faults: %d\n", page\_faults);

return 0;}

**LRU:**

#include <stdio.h>

#include <stdlib.h>

#define MAX\_PAGES 100

int main() {

int pages[MAX\_PAGES], n, capacity, i, j, page\_faults = 0, time = 0;

printf("Enter the number of pages: ");

scanf("%d", &n);

printf("Enter the page reference string: ");

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

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

printf("Enter the capacity of memory: ");

scanf("%d", &capacity);

int memory[capacity], counters[capacity];

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

memory[i] = -1;

counters[i] = 0;}

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

int page\_found = 0;

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

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

page\_found = 1;

counters[j] = ++time;

break;}}

if (!page\_found) {

page\_faults++;

int min\_counter = counters[0], min\_index = 0;

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

if (counters[j] < min\_counter) {

min\_counter = counters[j];

min\_index = j;}}

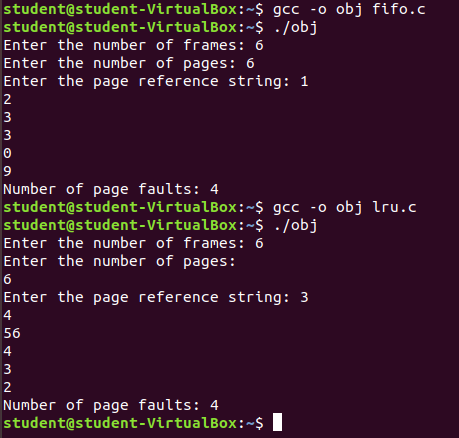
memory[min\_index] = pages[i];

counters[min\_index] = ++time;}}

printf("Number of page faults: %d\n", page\_faults);

return 0;}

**Output:**



**Task 02:** Execute both programs for the same set of reference strings. What difference did you observe? Comment.

**Solution:**

FIFO (First In First Out) and LRU (Least Recently Used) are two different page replacement algorithms. When executing both algorithms for the same set of reference strings, the difference you may observe is in the number of page faults. The number of page faults depends on the specific reference string and the size of the memory. In general, LRU tends to perform better than FIFO because it takes into account the recency of page usage, while FIFO only considers the order in which pages were brought into memory. However, without more information about the specific reference string and memory size, it’s difficult to comment on the exact difference you may observe.