OPERATING SYSTEM LAB

Name: Thorat Amey Arun Reg. No.: 23MCS1004

LAB EXPERIMENT 10
Page Replacement Algorithm

1. Write a C Program using threads to Implement Page Replacement Algorithm (FIFO, Optimal, and LRU) and determine their number of page fault. Each thread runs a separate algorithm and main thread prints the page faults of each algorithm.
   Input:
   Enter data (length of page reference sequence, page reference sequence and number of frames)

   Example
   Consider the following page reference string:
   1, 2, 3, 4, 2, 1, 4,2,5, 6, 2, 1, 6,5, 2, 3, 7, 5, 4,2, 6, 3, 2, 1, 2, 3, 6,4,2,5.

   How many page faults would occur for the following replacement algorithms, assuming three frames? Remember that all frames are initially empty, so your first unique pages will cost one fault each.
   - LRU replacement
   - FIFO replacement
   - Optimal replacement

   Program:

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#define MAX 50
int page_fault_fifo = 0;
int page_fault_lru = 0;
int page_fault_optimal = 0;
int pages[MAX];
int n;
int capacity;

void *FIFO(void *arg)
{
        int i, j, k, flag, pos, faults = 0;
```

```
        int frames[capacity];
        for (i = 0; i < capacity; i++) {
                frames[i] = -1;
        }
        j = 0;
        for (i = 0; i < n; i++) {
                flag = 0;
                for (k = 0; k < capacity; k++) {
                        if (frames[k] == pages[i]) {
                                flag = 1; break;
                        }
                }
                if (flag == 0) {
                        frames[j] = pages[i];
                        j = (j + 1) % capacity;
                        faults++;
                }
        }
        page_fault_fifo = faults;
        pthread_exit(0);
}

void *LRU(void *arg)
{
        int frames[capacity];
        int counter[MAX];
        int i, j, k, flag, pos, faults = 0;
        for (i = 0; i < capacity; i++) {
                frames[i] = -1;
        }
        for (i = 0; i < n; i++) {
                flag = 0;
                for (k = 0; k < capacity; k++) {
                        if (frames[k] == pages[i]) {
                                counter[k] = i; flag = 1;
                                break;
                        }
```

```
                }
                if (flag == 0) {
                        pos = 0;
                        for (k = 1; k < capacity; k++) {
                                if (counter[k] < counter[pos]) {
                                        pos = k;
                                }
                        }
                        frames[pos] = pages[i];
                        counter[pos] = i; faults++;
                }
        }
        page_fault_lru = faults; pthread_exit(0);
}

void *Optimal(void *arg) {
        int i, j, k, flag, pos, faults = 0;
        int frames[capacity];
        for (i = 0; i < capacity; i++) {
                frames[i] = -1;
        }
        for (i = 0; i < n; i++) {
                flag = 0;
                for (k = 0; k < capacity; k++) {
                        if (frames[k] == pages[i]) {
                                flag = 1;
                                break;
                        }
                }
                if (flag == 0) {
                        pos = 0;
                        if (frames[capacity - 1] != -1) {
                                int temp[MAX];
                                for (j = i + 1; j < n; j++) {
                                        temp[j] = -1;
                                        for (k = 0; k < capacity; k++) {
                                                if (frames[k] == pages[j]) {
```

OPERATING SYSTEM LAB

Name: Thorat Amey Arun Reg. No.: 23MCS1004

```
                                                temp[j] = j; break;
                                        }
                                }
                        }
                        for (j = i + 1; j < n; j++) {
                                if (temp[j] == -1) {
                                        pos = j;
                                        break;
                                }
                        }
                }
                frames[pos] = pages[i]; faults++;
        }
    }
    page_fault_optimal = faults; pthread_exit(0);
}

int main() {
    printf("Enter the length of page reference sequence: ");
    scanf("%d", &n);
    printf("Enter the page reference sequence: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &pages[i]);
    }
    printf("Enter the number of frames: ");
    scanf("%d", &capacity);
    pthread_t fifo_thread, lru_thread, optimal_thread;
    pthread_create(&fifo_thread, NULL, FIFO, NULL);
    pthread_create(&lru_thread, NULL, LRU, NULL);
    pthread_create(&optimal_thread, NULL, Optimal, NULL);
    pthread_join(fifo_thread, NULL);
    pthread_join(lru_thread, NULL);
    pthread_join(optimal_thread, NULL);
    printf("Page Faults using FIFO: %d\n", page_fault_fifo);
    printf("Page Faults using LRU: %d\n", page_fault_lru);
    printf("Page Faults using Optimal: %d\n", page_fault_optimal);
    return 0;
```

OPERATING SYSTEM LAB

Name: Thorat Amey Arun Reg. No.: 23MCS1004

}

Output: