Assignment Number 07

Name: Mihir Unmesh Patil

Roll NO: TYCOC213

Batch: C/C-3

CODE:

```
#include <stdio.h>

#include <stdbool.h>

bool is_page_present(int frames[], int capacity, int page) {
    for (int i = 0; i < capacity; i++) {
        if (frames[i] == page) return true;
    }
    return false;
}

void print_frames(int frames[], int capacity, int page, bool page_fault) {
    printf("Reference: %d | Frames: ", page);
    for (int i = 0; i < capacity; i++) {
        printf("%d ", frames[i]);
    }
    printf("\t");
    printf("| Page Fault: %s\n", page_fault ? "Yes" : "No");
}

int run_algorithm(int reference_string[], int n, int capacity, char* name,
                  void (*replace_page)(int frames[], int capacity, int page, int reference_string[], int n)) {
    int frames[capacity];
    for (int i = 0; i < capacity; i++) {
        frames[i] = -1;
    }
    printf("%s Page Replacement:\n", name);
    int page_faults = 0;
    for (int i = 0; i < n; i++) {
        bool page_fault = !is_page_present(frames, capacity, reference_string[i]);
        if (page_fault) {
            page_faults++;
            replace_page(frames, capacity, reference_string[i], reference_string, n);
        }
        print_frames(frames, capacity, reference_string[i], page_fault);
    }
    return page_faults;
}

void replace_page_fcfs(int frames[], int capacity, int page, int reference_string[], int n) {
    static int frame_index = 0;
    frames[frame_index] = page;
    frame_index = (frame_index + 1) % capacity;
}

void replace_page_lru(int frames[], int capacity, int page, int reference_string[], int n) {
    static int time[10];
    static int current_time = 0;
    for (int i = 0; i < capacity; i++) {
```

```
        if (frames[i] == -1) {
            frames[i] = page;
            time[i] = ++current_time;
            return;
        }
    }
    int lru_index = 0;
    for (int i = 1; i < capacity; i++) {
        if (time[i] < time[lru_index]) lru_index =
i;
    }
    frames[lru_index] = page;
    time[lru_index] = ++current_time;
}
void replace_page_optimal(int frames[], int
capacity, int page, int reference_string[], int n)
{
    for (int i = 0; i < capacity; i++) {
        if (frames[i] == -1) {
            frames[i] = page;
            return;
        }
    }
    int farthest = -1;
    int replace_index = -1;
    for (int i = 0; i < capacity; i++) {
        bool found = false;
        for (int j = n + 1; j < n; j++) {
            if (frames[i] == reference_string[j]) {
                if (j > farthest) {
                    farthest = j;
                    replace_index = i;
                }
                found = true;
                break;
            }
        }
        if (!found) {
            replace_index = i;
            break;
        }
    }
    if (replace_index == -1) replace_index = 0;
    frames[replace_index] = page;
}
int main() {
    int reference_string[] = {7, 0, 1, 2, 0, 3, 0, 4,
2, 3, 0, 3};
    const size_t n = sizeof(reference_string) /
sizeof(reference_string[0]);
    int capacity = 3;
    int fcfs_faults =
run_algorithm(reference_string, n, capacity,
"FCFS", replace_page_fcfs);
    printf("\nFCFS Total Page Faults: %d\n",
fcfs_faults);
    int lru_faults =
run_algorithm(reference_string, n, capacity,
"LRU", replace_page_lru);
    printf("\nLRU Total Page Faults: %d\n",
lru_faults);
    int optimal_faults =
run_algorithm(reference_string, n, capacity,
"Optimal", replace_page_optimal);
    printf("\nOptimal Total Page Faults: %d\n",
optimal_faults);
    return 0;
}
```

OUTPUT:

```
PS D:\Sem_6\OSL\Assignment_07_OSL> cd "d:\Sem_6\OSL\Ass
FCFS Page Replacement:
Reference: 7 | Frames: 7 -1 -1  | Page Fault: Yes
Reference: 0 | Frames: 7 0 -1   | Page Fault: Yes
Reference: 1 | Frames: 7 0 1    | Page Fault: Yes
Reference: 2 | Frames: 2 0 1    | Page Fault: Yes
Reference: 0 | Frames: 2 0 1    | Page Fault: No
Reference: 3 | Frames: 2 3 1    | Page Fault: Yes
Reference: 0 | Frames: 2 3 0    | Page Fault: Yes
Reference: 4 | Frames: 4 3 0    | Page Fault: Yes
Reference: 2 | Frames: 4 2 0    | Page Fault: Yes
Reference: 3 | Frames: 4 2 3    | Page Fault: Yes
Reference: 0 | Frames: 0 2 3    | Page Fault: Yes
Reference: 4 | Frames: 4 3 0    | Page Fault: Yes
Reference: 4 | Frames: 4 3 0    | Page Fault: Yes
Reference: 2 | Frames: 4 2 0    | Page Fault: Yes
Reference: 3 | Frames: 4 2 3    | Page Fault: Yes
Reference: 0 | Frames: 0 2 3    | Page Fault: Yes
Reference: 3 | Frames: 0 2 3    | Page Fault: No

FCFS Total Page Faults: 10
LRU Page Replacement:
Reference: 7 | Frames: 7 -1 -1  | Page Fault: Yes
Reference: 0 | Frames: 7 0 -1   | Page Fault: Yes
Reference: 1 | Frames: 7 0 1    | Page Fault: Yes
Reference: 2 | Frames: 2 0 1    | Page Fault: Yes
Reference: 0 | Frames: 2 0 1    | Page Fault: No
Reference: 3 | Frames: 2 3 1    | Page Fault: Yes
Reference: 0 | Frames: 2 3 0    | Page Fault: Yes
Reference: 4 | Frames: 4 3 0    | Page Fault: Yes
Reference: 2 | Frames: 4 2 0    | Page Fault: Yes
Reference: 3 | Frames: 4 2 3    | Page Fault: Yes
Reference: 0 | Frames: 0 2 3    | Page Fault: Yes
Reference: 3 | Frames: 0 2 3    | Page Fault: No

LRU Total Page Faults: 10
Optimal Page Replacement:
Reference: 7 | Frames: 7 -1 -1  | Page Fault: Yes
Reference: 0 | Frames: 7 0 -1   | Page Fault: Yes
Reference: 1 | Frames: 7 0 1    | Page Fault: Yes
Reference: 2 | Frames: 2 0 1    | Page Fault: Yes
Reference: 0 | Frames: 2 0 1    | Page Fault: No
Reference: 3 | Frames: 3 0 1    | Page Fault: Yes
Reference: 0 | Frames: 3 0 1    | Page Fault: No
Reference: 4 | Frames: 4 0 1    | Page Fault: Yes
Reference: 2 | Frames: 2 0 1    | Page Fault: Yes
Reference: 3 | Frames: 3 0 1    | Page Fault: Yes
Reference: 0 | Frames: 3 0 1    | Page Fault: No
Reference: 3 | Frames: 3 0 1    | Page Fault: No

Optimal Total Page Faults: 8
```