

LAB 8

8) Write a C program to simulate page replacement algorithms.

a) FIFO

b) LRU

c) Optimal

```
#include <stdio.h>
#include <stdbool.h>
int count = 0;
int findOptimal(int pages[], int frames[], int n, int index, int f) {
    int res = -1, farthest = index;
    for (int i = 0; i < f; i++) {
        int j;
        for (j = index; j < n; j++) {
            if (frames[i] == pages[j]) {
                if (j > farthest) {
                    farthest = j;
                    res = i;
                }
                break;
            }
        }
        if (j == n)
            return i;
    }
    return (res == -1) ? 0 : res;
}

int search(int key, int frames[], int f) {
    for (int i = 0; i < f; i++)
        if (frames[i] == key)
            return i;
    return -1;
}

void printFrames(int frames[], int f) {
    printf("PF No. %2d: ", ++count);
    for (int i = 0; i < f; i++) {
        if (frames[i] == -1)
            printf("- ");
        else
            printf("%d ", frames[i]);
    }
    printf("\n");
}
```

```

}

void fifo(int pages[], int n, int f) {
    int frames[f], front = 0, faults = 0;
    for (int i = 0; i < f; i++) frames[i] = -1;

    printf("\nFIFO Page Replacement Process:\n");
    for (int i = 0; i < n; i++) {
        if (search(pages[i], frames, f) == -1) {
            frames[front] = pages[i];
            front = (front + 1) % f;
            printFrames(frames, f);
            faults++;
        }
    }
    printf("FIFO Page Faults: %d\n", faults);
}

void lru(int pages[], int n, int f) {
    int frames[f], recent[f], faults = 0;
    for (int i = 0; i < f; i++) {
        frames[i] = -1;
        recent[i] = -1;
    }

    printf("\nLRU Page Replacement Process:\n");
    for (int i = 0; i < n; i++) {
        int pos = search(pages[i], frames, f);
        if (pos == -1) {
            int lru_index = 0, min = recent[0];
            for (int j = 1; j < f; j++) {
                if (recent[j] < min) {
                    min = recent[j];
                    lru_index = j;
                }
            }
            frames[lru_index] = pages[i];
            recent[lru_index] = i;
            printFrames(frames, f);
            faults++;
        } else {
            recent[pos] = i;
        }
    }
    printf("LRU Page Faults: %d\n", faults);
}

```

```

void optimal(int pages[], int n, int f) {
    int frames[f], faults = 0;
    for (int i = 0; i < f; i++) frames[i] = -1;

    printf("\nOptimal Page Replacement Process:\n");
    for (int i = 0; i < n; i++) {
        if (search(pages[i], frames, f) == -1) {
            int j;
            for (j = 0; j < f; j++) {
                if (frames[j] == -1) {
                    frames[j] = pages[i];
                    break;
                }
            }
            if (j == f) {
                int index = findOptimal(pages, frames, n, i + 1, f);
                frames[index] = pages[i];
            }
            printFrames(frames, f);
            faults++;
        }
    }
    printf("Optimal Page Faults: %d\n", faults);
}

int main() {
    int n, f;

    printf("Enter the number of Frames: ");
    scanf("%d", &f);
    printf("Enter the length of reference string: ");
    scanf("%d", &n);

    int pages[n];
    printf("Enter the reference string: ");
    for (int i = 0; i < n; i++)
        scanf("%d", &pages[i]);

    count = 0;
    fifo(pages, n, f);

    count = 0;
    lru(pages, n, f);

    count = 0;
    optimal(pages, n, f);

    return 0;
}

```

```
}
```

OUTPUT:

```
PS C:\Users\STUDENT\Desktop\1BF24CS121> cd "c:\Users\STUDENT\Desktop\1BF24CS121\" ; if ($?) { gcc lab8.c -o lab8 } ; if ($?) { .\lab8 }
Enter the number of Frames: 3
Enter the length of reference string: 6
Enter the reference string: 1 5 8 2 5 3

FIFO Page Replacement Process:
PF No. 1: 1 - -
PF No. 2: 1 5 -
PF No. 3: 1 5 8
PF No. 4: 2 5 8
PF No. 5: 2 3 8
FIFO Page Faults: 5

LRU Page Replacement Process:
PF No. 1: 1 - -
PF No. 2: 1 5 -
PF No. 3: 1 5 8
PF No. 4: 2 5 8
PF No. 5: 2 5 3
LRU Page Faults: 5

Optimal Page Replacement Process:
PF No. 1: 1 - -
PF No. 2: 1 5 -
PF No. 3: 1 5 8
PF No. 4: 2 5 8
PF No. 5: 3 5 8
Optimal Page Faults: 5
PS C:\Users\STUDENT\Desktop\1BF24CS121> |
```