



Subject :

Software :

Hardware :

Branch :

Semester :

Page No.

Prog No. 8

PROBLEM STATEMENT

Write a Program for frame sorting technique used in buffers.

ALGORITHM & CODE :

Frame Sorting is a technique used in buffer management, often to manage Page frames or buffer pools in operating systems.

Frame Sorting Program in C:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_FRAMES 10
typedef struct frame {
    int frame_id;
    int access_count;
} frame;

void sort_frames (frame frames [], int n) {
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (frames[j].access_count <
                frames[j+1].access_count) {
                frame temp = frames[j];
                frames[j] = frames[j+1];
                frames[j+1] = temp;
            }
        }
    }
}
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty

Date :

Signature of Student

Date :



Subject :

Software :

Hardware :

Branch :

Semester :

Page No.

Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
void access_frame and update its access count
if (frames[i].frame_id == frame_id) {
    frames[i].access_count++;
    printf("frame %d accessed. Access
count : %d\n", frame_id, frames[i].access_count);
    return;
}

if (*frame_count < MAX_FRAMES) {
    frames[*frame_count].frame_id = frame_id;
    frames[*frame_count].access_count = 1;
    (*frame_count)++;
    printf("frame %d added to buffer. \n",
frame_id);
} else {
    printf("Buffer is full! Cannot add frame %d.
\n", frame_id);
}
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty

Date :

Signature of Student

Date :



Subject :

Software :

Hardware :

Branch :

Semester :

Page No.

Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
Void display-Frames (Frame frames [], int
frame-count) {
    printf ("In current frames in buffer : \n");
    for (int i = 0 ; i < frame-count ; i++) {
        printf ("Frame ID : %d , Access count : %d\n",
frames [i].frame-id , frames[i].access-count);
    }
}

int main () {
    Frame frames [MAX-FRAMES];
    int frame-count = 0;

    access-frame (frames , &frame-count , 1);
    access-frame (frames , &frame-count , 2);
    access-frame (frames , &frame-count , 3);
    access-frame (frames , &frame-count , 1);
    access-frame (frames , &frame-count , 4);
    access-frame (frames , &frame-count , 1);
    access-frame (frames , &frame-count , 2);

    display-Frames (frames , frame-count);
    sort-Frames (frames , frame-count);
    printf ("After sorting frames by access count : \n");
}
```

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty

Date :

Signature of Student

Date :



Subject :

Software :

Hardware :

Branch :

Semester :

Page No.

Prog No.

PROBLEM STATEMENT

ALGORITHM & CODE :

```
display - frames (frames , frame-count);  
return 0;  
}
```

OP

Frame 1 added to buffer.

Frame 2 added to buffer.

Frame 3 added to buffer.

Frame 1 accessed, Access count: 2

Frame 4 added to buffer.

Frame 1 accessed, Access count: 3

Frame 2 accessed, Access count: 2

Current frames in buffer :

Frame ID: 1, Access count: 3

Frame ID: 2, Access count: 2

Frame ID: 3, Access count: 1

Frame ID: 4, Access count: 1

After Sorting frames by access count :

Frame ID: 1, Access count: 3

Frame ID: 2, Access count: 2

Frame ID: 3, Access count: 1

Frame ID: 4, Access count: 1

INPUT GIVEN

OUTPUT OBTAINED

REMARKS

GRADE :

Signature of Faculty

Date :

Signature of Student

Date :