

LAB10

Write a C program to simulate page replacement algorithms

- a) FIFO
- b) LRU
- c) Optimal

INPUT:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <limits.h>
4  int isPagePresent(int frames[], int frame_count, int page) {
5      for (int i = 0; i < frame_count; i++) {
6          if (frames[i] == page) {
7              return 1;
8          }
9      }
10     return 0;
11 }
12 void printFrames(int frames[], int frame_count) {
13     for (int i = 0; i < frame_count; i++) {
14         if (frames[i] == -1) {
15             printf("- ");
16         } else {
17             printf("%d ", frames[i]);
18         }
19     }
20     printf("\n");
21 }
22 void fifo(int pages[], int page_count, int frame_count) {
23     int frames[frame_count];
24     for (int i = 0; i < frame_count; i++) frames[i] = -1;
25     int index = 0, page_faults = 0;
26     printf("FIFO Page Replacement Process:\n");
27     for (int i = 0; i < page_count; i++) {
28         if (!isPagePresent(frames, frame_count, pages[i])) {
29             frames[index] = pages[i];
30             index = (index + 1) % frame_count;
31             page_faults++;
32             printFrames(frames, frame_count);
33             printf("PF No. %d\n", page_faults);
34         }
35     }
36
37     printf("The number of Page Faults using FIFO are %d\n\n", page_faults);
38 }
39 void lru(int pages[], int page_count, int frame_count) {
40     int frames[frame_count];
41     int counter[frame_count];
42     for (int i = 0; i < frame_count; i++) {
43         frames[i] = -1;
44         counter[i] = 0;
45     }
46
47     int page_faults = 0, time = 0;
48     printf("LRU Page Replacement Process:\n");
49     for (int i = 0; i < page_count; i++) {
50         int least recently used = time. lru index = -1;

```

```

51     for (int j = 0; j < frame_count; j++) {
52         if (frames[j] == pages[i]) {
53             counter[j] = time++;
54             lru_index = -1;
55             break;
56         }
57         if (frames[j] == -1) {
58             lru_index = j;
59             break;
60         }
61         if (counter[j] < least_recently_used) {
62             least_recently_used = counter[j];
63             lru_index = j;
64         }
65     }
66
67     if (lru_index != -1) {
68         frames[lru_index] = pages[i];
69         counter[lru_index] = time++;
70         page_faults++;
71         printFrames(frames, frame_count);
72         printf("PF No. %d\n", page_faults);
73     }
74 }
75
76 printf("The number of Page Faults using LRU are %d\n\n", page_faults);
77 }
78 void optimal(int pages[], int page_count, int frame_count) {
79     int frames[frame_count];
80     for (int i = 0; i < frame_count; i++) frames[i] = -1;
81     int page_faults = 0;
82     printf("Optimal Page Replacement Process:\n");
83     for (int i = 0; i < page_count; i++) {
84         if (!isPagePresent(frames, frame_count, pages[i])) {
85             int index = -1, farthest = i + 1;
86             for (int j = 0; j < frame_count; j++) {
87                 int k;
88                 for (k = i + 1; k < page_count; k++) {
89                     if (frames[j] == pages[k]) {
90                         if (k > farthest) {
91                             farthest = k;
92                             index = j;
93                         }
94                     }
95                 }
96                 break;
97             }
98             if (k == page_count) {
99                 index = j;
100                 break;
101             }

```

```

100         }
101     }
102     if (index == -1) index = 0;
103
104     frames[index] = pages[i];
105     page_faults++;
106     printFrames(frames, frame_count);
107     printf("PF No. %d\n", page_faults);
108 }
109 }
110
111 printf("The number of Page Faults using Optimal are %d\n\n", page_faults);
112 }
113
114 int main() {
115     int page_count, frame_count;
116     printf("Enter the number of pages: ");
117     scanf("%d", &page_count);
118     int pages[page_count];
119     printf("Enter the page sequence: ");
120     for (int i = 0; i < page_count; i++) {
121         scanf("%d", &pages[i]);
122     }
123
124     printf("Enter the number of frames: ");
125     scanf("%d", &frame_count);
126     fifo(pages, page_count, frame_count);
127     lru(pages, page_count, frame_count);
128     optimal(pages, page_count, frame_count);
129     return 0;
130 }
131

```

OUTPUT:

```
Enter the number of pages: 7
Enter the page sequence: 1
3
0
3
5
6
3
Enter the number of frames: 3
FIFO Page Replacement Process:
1 - -
PF No. 1
1 3 -
PF No. 2
1 3 0
PF No. 3
5 3 0
PF No. 4
5 6 0
PF No. 5
5 6 3
PF No. 6
The number of Page Faults using FIFO are 6

LRU Page Replacement Process:
1 - -
PF No. 1
1 3 -
PF No. 2
1 3 0
PF No. 3
5 3 0
PF No. 4
5 3 6
PF No. 5
The number of Page Faults using LRU are 5

Optimal Page Replacement Process:
1 - -
PF No. 1
3 - -
PF No. 2
3 0 -
PF No. 3
3 5 -
PF No. 4
3 6 -
PF No. 5
The number of Page Faults using Optimal are 5
```