

Practical No 8

Aim : Develop, debug and Execute a C program to simulate MRU page replacement algorithms

Apparatus: Mingw compiler for C/C++, and a text editor for developing C code file (Dev C++).

Theory :

What is MRU Page Replacement?

- It is a computer algorithm used to manage the cache area which stores data in the memory.
- MRU is short for Most Recently Used page replacement Algorithm.
- In MRU the most recently used page is removed and replaced with the new page.

Example:

Reference string: 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1

No of frames: 4

F4				2	2	2	2	2	2	3	0	3	2	2	2	0	0	0	0	0
F3			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
F2		0	0	0	0	3	0	4	4	4	4	4	4	4	4	4	4	4	4	4
F1	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
	*	*	*	*	HIT	*	*	*	HIT	*	*	*	*	HIT	HIT	*	HIT	HIT	HIT	HIT

Page fault (*): 12

Page hit (HIT): 8

Page fault ratio = No. of page fault / No. of reference string

$$= 12/20$$

$$= 60\%$$

Page hit ratio = No. of page ratio / No. of reference string

$$= 8/20$$

$$= 40\%$$

Code:

```
1. #include<stdio.h>
2.
3. int MRU(int time[], int n,int pages[], int cur){
4.     int i, minimum = time[0], pos = 0;
5.     int flag = 0;
6.     for(int i = 0; i < n; i++){
7.         //if the frame[i] is equal to last page, return index of page to be replaced
8.         if(time[i] == pages[cur-1]){
9.             pos = i;
10.        }
11.    }
12.
13.
14.    return pos;
15. }
16.
17. int main()
18. {
19.     int no_of_frames, no_of_pages, frames[10], pages[30], counter = 0, time[10], flag1, flag2,
20.     i, j, pos, faults = 0;
21.     printf("Enter number of frames: ");
22.     scanf("%d", &no_of_frames);
23.     int hit;
24.     printf("Enter number of pages: ");
25.     scanf("%d", &no_of_pages);
26.
27.     printf("Enter reference string: ");
28.
29.     for(i = 0; i < no_of_pages; ++i){
30.         scanf("%d", &pages[i]);
31.     }
32.
33.     for(i = 0; i < no_of_frames; ++i){
34.         frames[i] = -1;
35.     }
36.
37.     printf("\n");
38.     for(i = 0; i < no_of_frames; i++)
39.         printf("F%d\t",i+1);
40.
41.     for(i = 0; i < no_of_pages; ++i){
42.         hit = 1;
43.         flag1 = flag2 = 0;
44.
45.         for(j = 0; j < no_of_frames; ++j){
46.             if(frames[j] == pages[i]){
47.                 counter++;
48.                 time[j] = counter;
49.                 flag1 = flag2 = 1;
50.                 break;
51.             }
52.
53.             if(flag1 == 0){
54.                 for(j = 0; j < no_of_frames; ++j){
55.                     if(frames[j] == -1){
56.                         counter++;
57.                         faults++;
58.                         frames[j] = pages[i];
59.                         time[j] = counter;
60.                         flag2 = 1;
61.                         hit = 0;
62.                         Break;
63.                     }
64.                 }
65.             }
```

```

66.
67.     if(flag2 == 0){
68.         pos = MRU(frames, no_of_frames,pages,i);
69.         counter++;
70.         faults++;
71.         hit = 0;
72.         frames[pos] = pages[i];
73.         time[pos] = counter;
74.     }
75.
76.     printf("\n");
77.
78.     for(j = 0; j < no_of_frames; ++j){
79.         if(frames[j] == -1)
80.             printf(" \t");
81.         else
82.             printf("%d\t", frames[j]);
83.     }
84.     printf("%s\t",hit == 0 ? " Page Fault" : " Page Hit");
85. }
86.
87. printf("\n\nTotal Page Faults = %d", faults);
88. printf("\nTotal Page Hits   = %d", no_of_pages - faults);
89. printf("\nPage Fault ratio   = %.2f%%",faults/(float)no_of_pages * 100);
90. printf("\nPage Hit ratio     = %.2f%%",(no_of_pages - faults)/(float)no_of_pages * 100);
91. return 0;
92. }
93.
94. /*
95.
96. 4
97. 20
98. 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
99.
100. */

```

Output:

```
D:\_3rdYrNotes\IT-3rd-year-notes\Operating Systems\Codes\OS 08.exe
Enter number of frames: 4
Enter number of pages: 20
Enter reference string: 7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1

F1      F2      F3      F4
7
7      0
7      0      1
7      0      1      2
7      0      1      2      Page Hit
7      3      1      2      Page Fault
7      0      1      2      Page Fault
7      4      1      2      Page Fault
7      4      1      2      Page Hit
7      4      1      3      Page Fault
7      4      1      0      Page Fault
7      4      1      3      Page Fault
7      4      1      2      Page Fault
7      4      1      2      Page Hit
7      4      1      2      Page Hit
7      4      1      0      Page Fault
7      4      1      0      Page Hit
7      4      1      0      Page Hit
7      4      1      0      Page Hit
7      4      1      0      Page Hit

Total Page Faults = 12
Total Page Hits   = 8
Page Fault ratio  = 60.00%
Page Hit ratio    = 40.00%
-----
Process exited after 8.478 seconds with return value 0
Press any key to continue . . .
```

8.1 Output

Conclusion:

Hence, by performing this practical I got to know about the concept of Page replacement, page fault and page hit. I also learnt about MRU page replacement algorithm i.e. Most Recently Used Page Replacement Algorithm. I also developed, debugged and executed a C program to simulate MRU page replacement algorithm.