**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

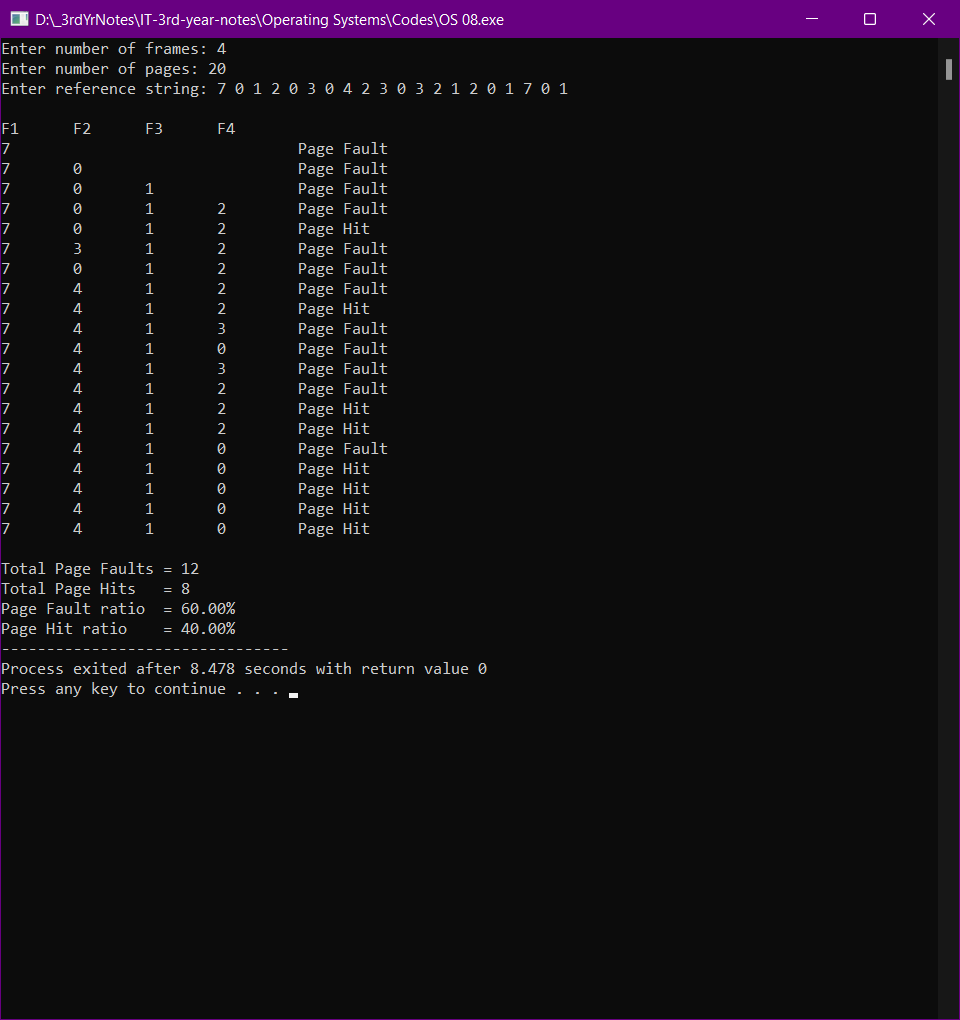
= 40%

**Code:**

1. #include<stdio.h>
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. }

14. return pos;
15. }
17. int main()
18. {
19. int no\_of\_frames, no\_of\_pages, frames[10], pages[30], counter = 0, time[10], flag1, flag2, i, j, pos, faults = 0;
20. printf("Enter number of frames: ");
21. scanf("%d", &no\_of\_frames);
22. int hit;
23. printf("Enter number of pages: ");
24. scanf("%d", &no\_of\_pages);
26. printf("Enter reference string: ");
28. for(i = 0; i < no\_of\_pages; ++i){
29. scanf("%d", &pages[i]);
30. }
32. for(i = 0; i < no\_of\_frames; ++i){
33. frames[i] = -1;
34. }
36. printf("\n");
37. for(i = 0; i < no\_of\_frames; i++)
38. printf("F%d\t",i+1);
40. for(i = 0; i < no\_of\_pages; ++i){
41. hit = 1;
42. flag1 = flag2 = 0;
44. for(j = 0; j < no\_of\_frames; ++j){
45. if(frames[j] == pages[i]){
46. counter++;
47. time[j] = counter;
48. flag1 = flag2 = 1;
49. break;
50. }
51. }
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. }
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. }
76. printf("\n");
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. }
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. }
94. /\*
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
100. \*/

**Output:**



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.