SOMAIYA
VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya
TRUST

Batch: A2   **Roll No.: 16010122041**
Experiment / assignment / tutorial No. 6
Grade: AA / AB / BB / BC / CC / CD /DD
Signature of the Staff In-charge with date

TITLE: Implementation of LRU Page Replacement Algorithm.

AIM: The LRU algorithm replaces the least recently used that is the last accessed memory block from user.

Expected OUTCOME of Experiment: (Mention CO/CO's attained here)

Books/ Journals/ Websites referred:

1. Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", Fifth Edition, TataMcGraw-Hill.

2. William Stallings, "Computer Organization and Architecture: Designing for Performance", Eighth Edition, Pearson.

Pre Lab/ Prior Concepts:

It follows a simple logic, while replacing it will replace that page which has least recently used out of all.

a) A hit is said to be occurred when a memory location requested is already in the cache.

   b) When cache is not full, the number of blocks is added.

   c) When cache is full, the block is replaced which is recently used

Algorithm:

1.   Start
2.   Get input as memory block to be added to cache
3.   Consider an element of the array
4.   If cache is not full, add element to the cache array
5.   If cache is full, check if element is already present
6.   If it is hit is incremented

7.   If not, element is added to cache removing least recently used element
8.   Repeat step 3 to 7 for remaining elements
9.   Display the cache at very instance of step 8
10.   Print hit ratio
11.   End

Example:

```c
#include<stdio.h>

int findLRU(int time[], int n){
    int i, minimum = time[0], pos = 0;

        for(i = 1; i < n; i++){
            if(time[i] < minimum){
            minimum = time[i];
            pos = i;
        }
    }
    return pos;
}

int main()
{
    int no_of_frames, no_of_pages, frames[10], pages[30], counter =
0, time[10], flag1, flag2, i, j, pos, faults = 0;

    printf("Enter number of pages: ");
    scanf("%d", &no_of_pages);
    printf("Enter number of frames: ");
    scanf("%d", &no_of_frames);
    printf("Enter reference string: ");
        for(i = 0; i < no_of_pages; i++){
         scanf("%d", &pages[i]);
        }

        for(i = 0; i < no_of_frames; i++){
         frames[i] = -1;
        }
```

```c
for(i = 0; i < no_of_pages; i++){
 flag1 = flag2 = 0;

    for(j = 0; j < no_of_frames; j++){
        if(frames[j] == pages[i]){
        counter++;
        time[j] = counter;
        flag1 = flag2 = 1;
        break;
        }
    }

 if(flag1 == 0){
    for(j = 0; j < no_of_frames; j++){
        if(frames[j] == -1){
        counter++;
        faults++;
        frames[j] = pages[i];
        time[j] = counter;
        flag2 = 1;
        break;
        }
    }
 }

 if(flag2 == 0){
    pos = findLRU(time, no_of_frames);
    counter++;
    faults++;
    frames[pos] = pages[i];
    time[pos] = counter;
 }

 printf("\n");

 for(j = 0; j < no_of_frames; j++){
    printf("%d\t", frames[j]);
 }
```

```
    }
    printf("\n\nTotal Page FAULTS= %d", faults);
    printf("\nTotal page HITS= %d",(no_of_pages - faults));

    return 0;
}
```

```
Enter number of pages: 7
Enter number of frames: 3
Enter reference string: 2 1 0 5 1 8 6

2       -1      -1
2       1       -1
2       1       0
5       1       0
5       1       0
5       1       8
6       1       8

Total Page FAULTS= 6
Total page HITS= 1
```

Post Lab Descriptive Questions

1. Define hit rate and miss ratio?
A hit ratio is a calculation of cache hits, and comparing them with how many total content requests were received.
A miss ratio is the flip side of this where the cache misses are calculated and compared with the total number of content requests that were received.

2. What is the need for virtual memory?
Virtual memory serves two purposes. First, it allows us to extend the use of physical memory by using disk. Second, it allows us to have memory protection, because each virtual address is translated to a physical address.

Conclusion : Successfully executed the given program.

Date: _____   **Signature of faculty in-charge**