Batch: A3 Roll No.: 16010121051

Experiment / assignment / tutorial No6

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:

Reference string : 4,7,6,1,7,6,1,2,7,2

Memory frames = 3

LRU Page Replacement Algorithm

4	7	6	1	7	6	1	2	7	2
		6	6	6	6	6	6	7	7
	7	7	7	7	7	7	2	2	2
4	4	4	1	1	1	1	1	1	1
				↑ Hit	↑ Hit	↑ Hit			↑ Hit

Hits : 4
faults : 6

Hit ratio : 0.4

Code:

```
#include <stdio.h>
#include <stdlib.h>


int pr[20];
int num[3]={-1}; int count[3]={0};

int main()
{
   int m,hit=0,f=0,j,i,k,l,fram;
   printf("Number of frames: ");
   scanf("%d",&fram);

   int a[1][fram];

   printf("Length of reference string: ");
   scanf("%d",&m);

   for(j=0;j<fram;j++)
   {
      a[0][j]=-1;
   }

   printf("Enter Reference string:");

   for(i=0;i<m;i++)
   {
      scanf("%d",&pr[i]);
   }

   printf("\n");
   printf("Output is:\n");
   int c=0;

   while(c<m)
   {
      int p = pr[c];
```

```
        if(a[0][0]!=-1 && a[0][1]!=-1 && a[0][2]!=-1)
        {
           if(p!=num[0] && (p!=num[1] && p!=num[2]))
           {
              if((a[0][1]==pr[c-1]&&a[0][2]==pr[c-2])||(a[0][1]==pr[c-
2]&&a[0][2]==pr[c-1]))
              {
                 a[0][0]=-1;
                 count[0]=0;
              }

              else          if((a[0][0]==pr[c-1]&&a[0][2]==pr[c-2])||(a[0][0]==pr[c-
2]&&a[0][2]==pr[c-1]))
              {
                 a[0][1]=-1;
                 count[1]=0;
              }

              else
              {
                 a[0][2]=-1; count[2]=0;
              }
           }

        }

        for(j=0;j<fram;j++)
        {
           if(a[0][j]== -1)
           {
              a[0][j] = p;
              f++;
              num[j]=p;

              for(k=0;k<fram;k++)
              {
                 if(a[0][k]>=0)
                 {
                    count[k]++;
```

```
                }
            }

            break;
        }

        else if(a[0][j]== p)
        {
            hit++;
            for(k=0;k<fram;k++)
            {
                if(a[0][k]>=0)
                {
                    count[k]++;
                }
            }
            break;
        }
    }
    for(l=0 ;l<fram;l++)
    {
        if(a[0][l]==-1)
        {
            printf(" ");
        }
        else
        {
            printf("%d ",a[0][l]);
        }
    }
    printf("\n");
    c++;
}

// printing hits, misses and hit ratio
printf("\n");
printf("\nNumber of Hits : %d ",hit);
printf("\nFaults : %d ",f);
double ratio = (double)hit/m;
```

```
   printf("\nHIT Ratio is: %.2lf",ratio);
   return 0;
}
```

Output:

```
C:\Academics\SY\COA\exp6.exe
Number of frames: 5
Length of reference string: 5
Enter Reference string:1
2
3
4
5

Output is:
1
1 2
1 2 3
4 2 3
4 5 3


Number of Hits : 0
Faults : 5
HIT Ratio is: 0.00
--------------------------------
Process exited after 7.535 seconds with return value 0
Press any key to continue . . .
```

Post Lab Descriptive Questions

1. Define hit rate and miss ratio?
Ans: 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?
Ans: 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:

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