K. J. Somaiya College of Engineering, Mumbai-77

Batch: A1	**Roll No.: 16010120015**
Experiment / assignment / tutorial No. (Fifo)	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Implementation of **FIFO** Page Replacement Algorithm

AIM: The FIFO algorithm uses the principle that the block in the set which has been in for the longest time will be replaced

Expected OUTCOME of Experiment:
CO 4-Learn and evaluate memory organization and cache structure

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.
3. Dr. M. Usha, T. S. Srikanth, "Computer System Architecture and Organization", First Edition, Wiley-India.

Pre Lab/ Prior Concepts:

T he FIFO algorithm uses the principle that the block in the set which has been in the block for the longest time is replaced. FIFO is easily implemented as a round robin or criteria buffer technique. The data structure used for implementation is a queue. Assume that the number of cache pages is three. Let the request to this cache is shown alongside.

Algorithm:

1. A hit is said to be occurred when a memory location requested is already in the cache.
2. When cache is not full, the number of blocks is added.
3. When cache is full, the block is replaced which was added first

Design Steps:

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 first element (which is in first).
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.

Code:

```
import java.util.*;
class Main {
public static void main(String[] args) { Scanner
        sc=new Scanner(System.in);
        System.out.print("Enter stack size: "); int
        len=sc.nextInt();
        int j=-1, elem, hit=0;
        int arr[]=new int[len];
        System.out.print("Enter number of elements you want to enter: ");
        int n=sc.nextInt();
        for(int i=0;i<n; i++){
                System.out.print("\nEnter number "+(i+1)+": ");
                elem=sc.nextInt();
                boolean result=search(elem, arr);
                if(result)
                {
                        System.out.print("\nStack unchanged: ");
                        print(arr);
                        hit++;
                }
                else{
                        arr[(j+1)%len]=elem; System.out.print("\nCurrent
                        stack: ");
                        print(arr);
                        j++;
                }

                if(i==n-1)
                System.out.println("\nHit: "+hit)
                }
```

```
}
        static boolean search(int elem, int[] arr){
            for(int i=0;i<arr.length;i++){
                if(arr[i]==elem){
                    return true;
                }
            }
            return false;
}
        static void print(int[] arr){
            for(int i=0;i<arr.length;i++){ System.out.print(arr[i]+"
                ");
            }
        }
}
```

Output:

```
Enter stack size: 3
Enter number of elements you want to enter: 6

Enter number 1: 1

Current stack: 1 0 0
Enter number 2: 2

Current stack: 1 2 0
Enter number 3: 5

Current stack: 1 2 5
Enter number 4: 3

Current stack: 3 2 5
Enter number 5: 5

Stack unchanged: 3 2 5
Enter number 6: 2

Stack unchanged: 3 2 5
Hit: 2
```

Post Lab Descriptive Questions

1. What is meant by memory interleaving?

It is a technique for compensating the relatively slow speed of DRAM (Dynamic RAM). In this technique, the main memory is divided into memory banks which can be accessed individually without any dependency on the other.

2. Explain Paging Concept?

Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non – contiguous.

- Logical Address or Virtual Address (represented in bits): An address generated by the CPU
- Logical Address Space or Virtual Address Space(represented in words or bytes): The set of all logical addresses generated by a program
- Physical Address (represented in bits): An address actually available on memory unit
- Physical Address Space (represented in words or bytes): The set of all physical addresses corresponding to the logical addresses

The mapping from virtual to physical address is done by the memory management unit (MMU) which is a hardware device and this mapping is known as paging technique.

- The Physical Address Space is conceptually divided into a number of fixed-size blocks, called **frames**.
- The Logical address Space is also splitted into fixed-size blocks, called **pages**.
- Page Size = Frame Size

Conclusion: Successfully simulated the FIFO Page Replacement Algorithm for various test cases.

Date: 29.11.2021 **Signature of faculty in-charge**