Batch: E-2	Roll No.: 16010123325
Experiment / assignment / tutorial No.___6___	

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: (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.

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.

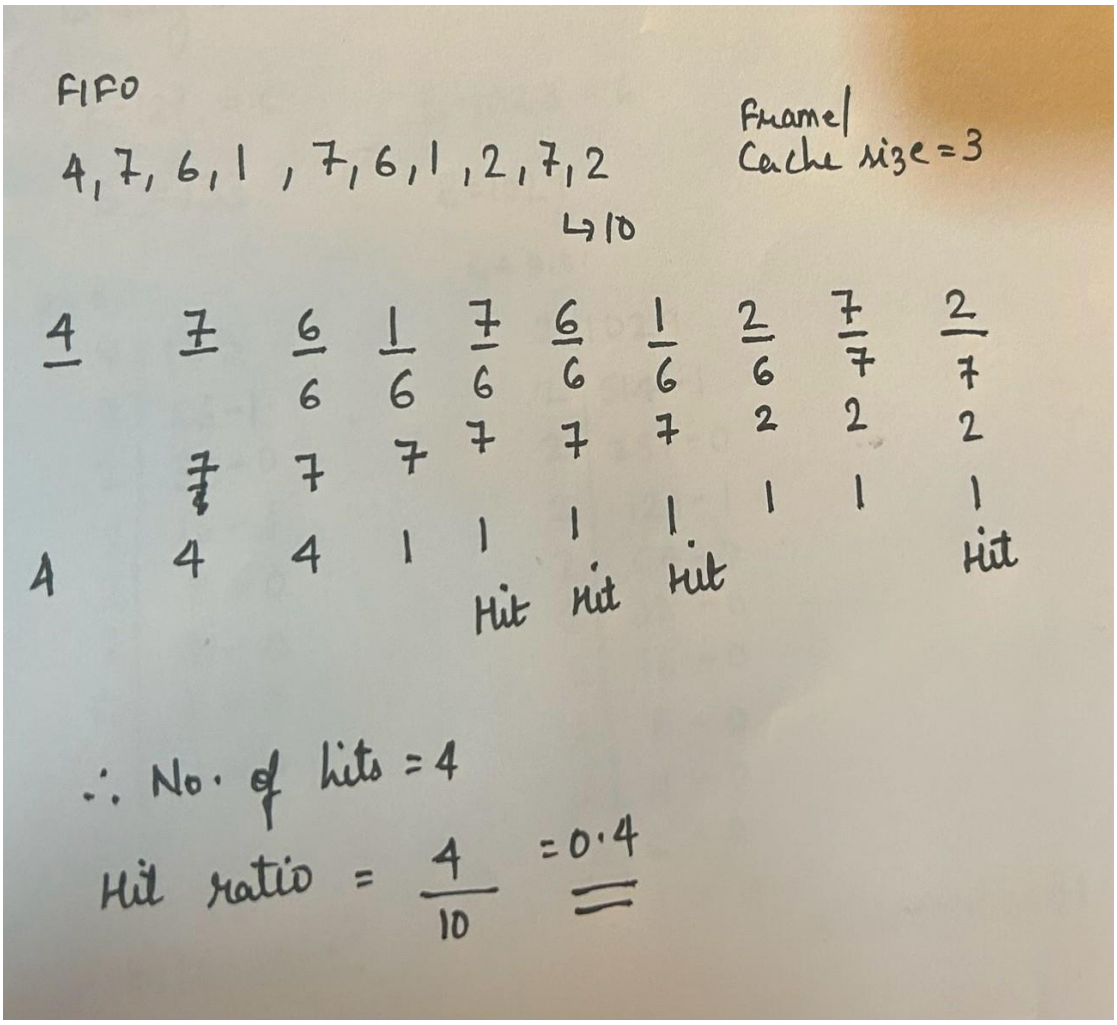
3. When cache is full, the block is replaced which was added first

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.

Example:


FIFO
4, 7, 6, 1, 7, 6, 1, 2, 7, 2
Frame/
Cache size = 3
↳ 10
4 7 6 1 7 6 1 2 7 2
6 6 6 6 6 2 2 2
7 7 7 7 7 1 1 1
4 4 1 1 1 1
A Hit Hit Hit Hit
∴ No. of hits = 4
Hit ratio = 4/10 = 0.4

Code-

```cpp
#include<bits/stdc++.h>
using namespace std;

int main() {
    int numPages, frames;
    cout << "Enter number of pages: ";
    cin >> numPages;

    cout << "Enter number of frames: ";
    cin >> frames;

    vector<int> pages(numPages);
    cout << "Enter page numbers: ";
    for (int i = 0; i < numPages; i++) {
        cin >> pages[i];
    }

    deque<int> cache;
    int hits = 0;
    cout << "\nFIFO Page Replacement Process: ";
    for (int i = 0; i < numPages; i++) {
        int currentPage = pages[i];
        if (find(cache.begin(), cache.end(), currentPage) != cache.end()) {
            hits++;
            cout << "Cache: ";
            for (int page : cache) {
                cout << page << " ";
            }
            cout << endl;
        } else {
            if (cache.size() == frames) {
                cache.pop_front();
            }
            cache.push_back(currentPage);
            cout << "Cache: ";
            for (int page : cache) {
                cout << page << " ";
            }
            cout << endl;
        }
    }

    double hitRatio = (double)hits / numPages;
```

```
    cout << "\nTotal hits: " << hits << endl;
    cout << "Hit ratio: " << hitRatio << endl;

    return 0;
}
```

Output-


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
GITLENS
Code
PS C:\Users\Shrey\OneDrive\Desktop\KJSCE\SEM-3\COA\Programs> cd "c:\Users\Shrey\OneDrive\Desktop\KJSCE\SEM-3\COA\Programs\" ; if ($?) { g++ FIFO.cpp -o FIFO } ; if ($?) { .\FIFO }
Enter number of pages: 10
Enter number of frames: 3
Enter page numbers: 4 7 6 1 7 6 1 2 7 2
FIFO Page Replacement Process: Cache: 4
Cache: 4 7
Cache: 4 7 6
Cache: 7 6 1
Cache: 7 6 1
Cache: 7 6 1
Cache: 7 6 1
Cache: 6 1 2
Cache: 1 2 7
Cache: 1 2 7
Total hits: 4
Hit ratio: 0.4
PS C:\Users\Shrey\OneDrive\Desktop\KJSCE\SEM-3\COA\Programs>

Post Lab Descriptive Questions

1. **What is meant by memory interleaving?**

Memory interleaving is based on the principle of parallelism, where multiple memory operations can be executed simultaneously. It involves dividing the memory into multiple banks or modules and organizing them in such a way that consecutive memory locations are stored in different banks. This allows the system to access multiple memory locations in parallel, reducing the time required to fetch data.

In a system without interleaving, memory is typically organized as a single block, and access to consecutive memory locations requires sequential access. This sequential access can lead to increased latency, especially in systems where memory access times are relatively slow compared to processor speeds.

2. **Explain Paging Concept?**

In Operating Systems, Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages. The main idea behind the paging is to divide each process in the form of pages. The main memory will also be divided in the form of frames.

One page of the process is to be stored in one of the frames of the memory. The pages can be stored at the different locations of the memory but the priority is always to find the contiguous frames. Pages of the process are brought into the main memory only when they are required otherwise they reside in the secondary storage

Conclusion

In this experiment, we implemented the **FIFO caching algorithm** in Java to evaluate the efficiency of a cache system.

Date: _____