Signature of faculty in-charge

Batch: A3 Roll No.: 16010121051

Experiment / assignment / tutorial No._____

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: (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.

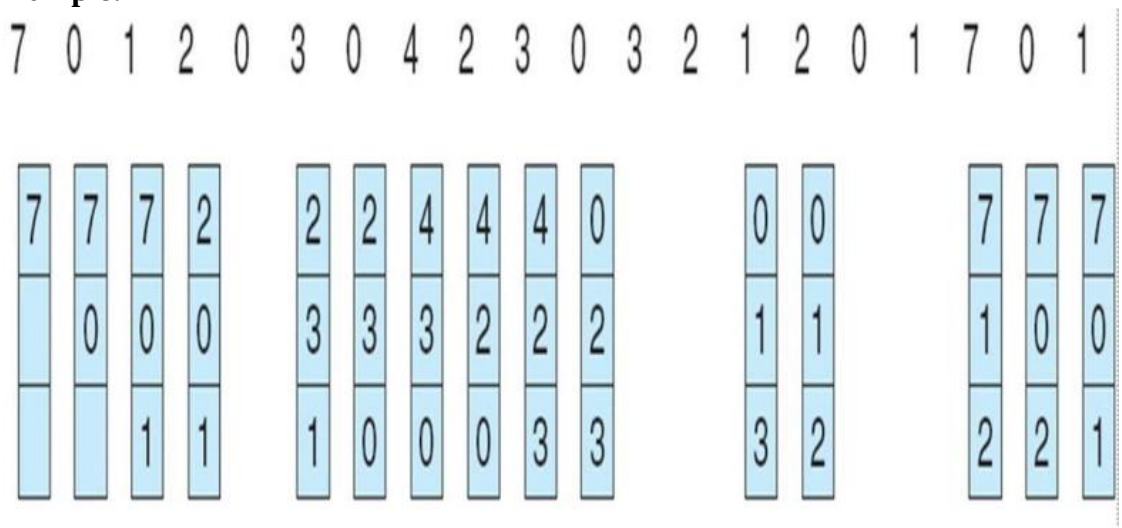
Algorithm:

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:



Implementation:

```
#include<bits/stdc++.h>
using namespace std;
int main(){
   int i, j, k, hitCount = 0;
   int frames = 3;
   int p_count = 14;

   vector <int> processes{7,0,1,2,0,3,0,3,0,3,2,1,2,0};
   vector <int> hit(p_count);

   vector<vector<int>> a(frames);
   for(i = 0; i < frames; i++){
      a[i] = vector <int>(p_count,-1);
   }

   map <int, int> mp;

   for(i = 0; i < p_count ; i++){
      vector<pair<int,int>> c;

      for(auto x: mp)
      {
         c.push_back({x.second, x.first});
```

```
}

sort(c.begin(),c.end());

bool hasCompleted = false;

for(j = 0;j < frames; j++){
   if(a[j][i] == processes[i])
   {
      hitCount++;
      hit[i] = true;

      mp[processes[i]]++;
      hasCompleted = true;
      break;
   }
   if(a[j][i] == -1)
   {
      for(k = i ; k < p_count; k++)
         a[j][k] = processes[i];

      mp[processes[i]]++;
      hasCompleted = true;
      break;
```

```
        }
    }
    if(j == frames || hasCompleted == false){
        for(j = 0;j < frames; j++){
            if(a[j][i] == c[c.size() - 1].second){
                mp.erase(a[j][i]);

                for(k = i; k < p_count ; k++)
                    a[j][k]= processes[i];

                mp[processes[i]]++;
                break;
            }
        }
    }
    for(auto x:mp){
        if(x.first != processes[i]){
            mp[x.first]++;
        }
    }
}
cout << "   ";
for(i = 0 ; i < p_count; i++){
    cout << processes[i] << "  ";
```

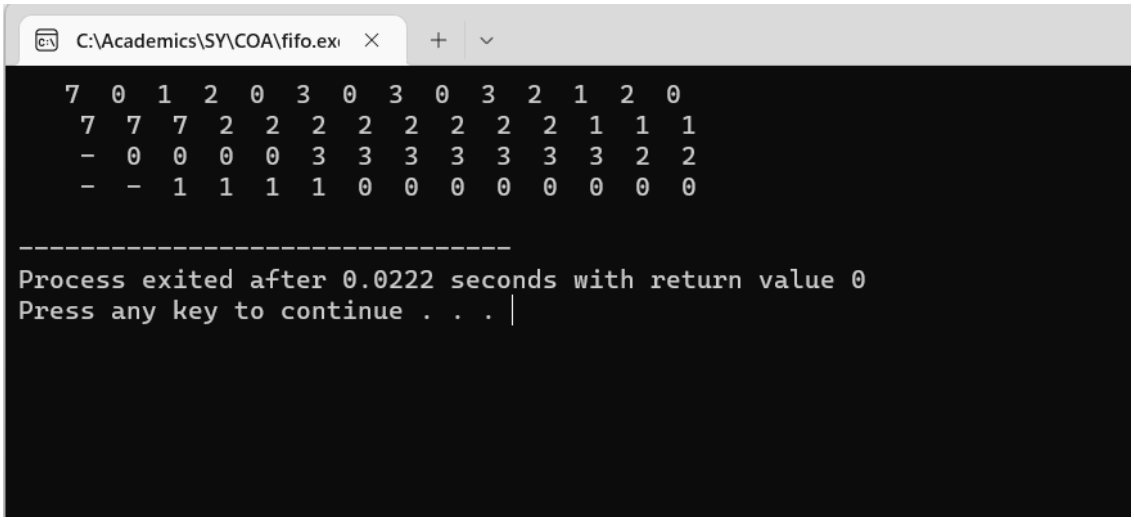
```
    }
    cout << endl;

    for(i = 0; i < frames; i++){
        cout << "    ";
        for(j = 0; j < p_count; j++){
            if(a[i][j] == -1)
                cout << "-  ";
                else
            cout << a[i][j] << "  ";
        }
        cout << endl;
    }




    return 0;
}
```

Output:



Post Lab Descriptive Questions

1. What is meant by memory interleaving?

Memory Interleaving is an abstraction technique which divides memory into a number of modules such that successive words in the address space are placed in the different module.

2. Explain Paging Concept?

Paging is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory in the form of pages. In the Paging method, the main memory is divided into small fixed-size blocks of physical memory, which is called frames. The size of a frame should be kept the same as that of a page to have maximum utilization of the main memory and to avoid external fragmentation. Paging is used for faster access to data, and it is a logical concept.

Conclusion:

We have successfully implemented FIFO Page Replacement Algorithm

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