CS212: Assignment 8

Md Shabbir Jamal

Department of Computer Science and Engineering
BIT, Mesra, Ranchi
btech10026.20@bitmesra.ac.in

1. Write a program to implement LRU page replacement algorithmFind the number of page faults for the following reference string:0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1Verify the above reference string for 3,4 and 5 number of page frames in memory.

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

int main()
{
    /* n - number of reference elements
       frames - number of frames in memory
       fault - number of page faults
       hit - number of page hits
       front - keep the "first in" element's index
    */
    int n, frames, fault = 0,hit = 0,front = 0;

    cout<<"Enter reference string size : ";
    cin>>n;

    //  ref_s - stores reference string
    int ref_s[n];
    cout<<"Enter reference string : ";
    for(int i = 0;i<n;i++)
    {
        cin>>ref_s[i];
    }
    cout<<"Enter number of page frames : ";
    cin>>frames;

    //table - its to show memory status
    vector<vector<int>> table(frames);
    for(int i = 0;i<frames;i++)
    {
        table[i] = vector<int>(n,-1);
```

```
}

// cur_mem - stores current position of memory
vector<int> cur_mem(frames,-1);

//inlist - it shows if an element was already present in memory or not
bool inlist = false;
for(int i = 0;i<n;i++)
{
    map<int,int> help;
    inlist = false;
    for(int j = 0;j<frames;j++)
    {
        if(cur_mem[j] == ref_s[i])
        {
            hit++;
            inlist = true;
            break;
        }
        if(cur_mem[j] == -1)
        {
            fault++;
            cur_mem[j] = ref_s[i];
            inlist = true;
            break;
        }
    }
    if(inlist == false)
    {
        fault++;
        int pt = INT_MAX,jpt = 0;
        for(int j = 0;j<frames;j++)
        {
            for(int k = i-1;k>=0;k--)
            {
                if(cur_mem[j] == ref_s[k])
                {
                    if(pt > k)
                    {
                        pt = k;
                        jpt = j;
                    }
                    break;
                }
            }
        }
        cur_mem[jpt] = ref_s[i];
    }
```

```
        for(int j = 0;j<frames;j++)
        {
            table[j][i] = cur_mem[j];
        }
    }


    // X - in the ouput means that frame is empty
    cout<<"\nref. str ";
    for(int i = 0;i<n;i++)
    {
        cout<<ref_s[i]<<" ";
    }
    cout<<"\n\n";
    for(int i = 0; i<frames;i++)
    {
        cout<<"Frames : ";
        for(int j =  0;j<n;j++)
        {
            if(table[i][j] == -1)
            {
                cout<<"X"<<" ";
            }
            else
            {
                cout<<table[i][j]<<" ";
            }
        }
        cout<<endl;
    }

    //Result
    cout<<"\tResult"<<endl;
    cout<<"\t\tFaults : "<<fault<<endl;
    cout<<"\t\tHits   : "<<hit<<endl;

    return 0;
}
```

Output

```
1.) Frame = 3

Enter reference string size : 12
Enter reference string : 0 2 1 6 4 0 1 0 3 1 2 1
Enter number of page frames : 3
```

```
ref. str 0 2 1 6 4 0 1 0 3 1 2 1

Frames : 0 0 0 6 6 6 1 1 1 1 1 1
Frames : X 2 2 2 4 4 4 4 3 3 3 3
Frames : X X 1 1 1 0 0 0 0 0 2 2
        Result
                Faults : 9
                Hits   : 3
```

2.) Frame = 4

```
Enter reference string size : 12
Enter reference string : 0 2 1 6 4 0 1 0 3 1 2 1
Enter number of page frames : 4

ref. str 0 2 1 6 4 0 1 0 3 1 2 1

Frames : 0 0 0 0 4 4 4 4 4 4 2 2
Frames : X 2 2 2 2 0 0 0 0 0 0 0
Frames : X X 1 1 1 1 1 1 1 1 1 1
Frames : X X X 6 6 6 6 6 3 3 3 3
        Result
                Faults : 8
                Hits   : 4
```

3.) Frame = 5

```
Enter reference string size : 12
Enter reference string : 0 2 1 6 4 0 1 0 3 1 2 1
Enter number of page frames : 5

ref. str 0 2 1 6 4 0 1 0 3 1 2 1

Frames : 0 0 0 0 0 0 0 0 0 0 0 0
Frames : X 2 2 2 2 2 2 2 3 3 3 3
Frames : X X 1 1 1 1 1 1 1 1 1 1
Frames : X X X 6 6 6 6 6 6 6 2 2
Frames : X X X X 4 4 4 4 4 4 4 4
        Result
                Faults : 7
                Hits   : 5
```