

Indian Institute of Information Technology Vadodara

CS266: Operating Systems Lab

Lab 10

Roll No. 201951105

Name: Nishant Andoriya

Question:-

In this assignment, you have to implement page-replacement algorithm FIFO and LRU. Consider different frame sizes 2, 3, 4, 5, and 6.

Input:-

- Take sequence of page reference from user.

Compute:-

- Page fault for different number of frames for each algorithm.
- Show graph of Page-Fault versus Number of Frames for both FIFO and LRU algorithm.

FIFO

Code:-

```
#include <bits/stdc++.h>
using namespace std;
int pageFaults(int pages[], int n, int capacity)
{
    unordered_set<int> s;
    queue<int> indexes;
    int page_faults = 0;
    for (int i = 0; i < n; i++)
    {
        if (s.size() < capacity)
        {
            if (s.find(pages[i]) == s.end())
            {
                s.insert(pages[i]);
                page_faults++;
                indexes.push(pages[i]);
            }
        }
    }
}
```

```

    }
    else
    {
        if (s.find(pages[i]) == s.end())
        {
            int val = indexes.front();
            indexes.pop();
            s.erase(val);
            s.insert(pages[i]);
            indexes.push(pages[i]);
            page_faults++;
        }
    }
}
return page_faults;
}
int main()
{
    cout << "Enter Number of Pages : ";
    int n;
    cin >> n;
    int pages[n];
    cout << "\nEnter Pages : ";
    for (int i = 0; i < n; i++)
    {
        int x;
        cin >> x;
        pages[i] = x;
    }
    int capacity[] = {2, 3, 4, 5, 6};
    for (int i = 0; i < 5; i++)
        cout << "Frame Size : " << capacity[i] << ", Page Faults: " << pageFaults
(pages, n, capacity[i]) << "\n";
    return 0;
}

```

OutPut:-

```
Enter Number of Pages : 10

Enter Pages : 1 2 3 4 5 3 2 1 5 4
Frame Size : 2, Page Faults: 10
Frame Size : 3, Page Faults: 8
Frame Size : 4, Page Faults: 6
Frame Size : 5, Page Faults: 5
Frame Size : 6, Page Faults: 5

...Program finished with exit code 0
Press ENTER to exit console.
```

LRU

Code:-

```
#include <bits/stdc++.h>
using namespace std;
int pageFaults(int pages[], int n, int capacity)
{
    unordered_set<int> s;
    queue<int> indexes;
    int page_faults = 0;
    for (int i = 0; i < n; i++)
    {
        if (s.size() < capacity)
        {
            if (s.find(pages[i]) == s.end())
            {
                s.insert(pages[i]);
                page_faults++;
                indexes.push(pages[i]);
            }
        }
        else
        {
            if (s.find(pages[i]) == s.end())
            {
                int val = indexes.front();
                indexes.pop();
                s.erase(val);
                s.insert(pages[i]);
            }
        }
    }
    return page_faults;
}
```

```

        indexes.push(pages[i]);
        page_faults++;
    }
}
return page_faults;
}
int main()
{
    cout << "Enter Number of Pages : ";
    int n;
    cin >> n;
    int pages[n];
    cout << "\nEnter Pages : ";
    for (int i = 0; i < n; i++)
    {
        int x;
        cin >> x;
        pages[i] = x;
    }
    int capacity[] = {2, 3, 4, 5, 6};
    for (int i = 0; i < 5; i++)
        cout << "Frame Size : " << capacity[i] << ", Page Faults: " << pageFaults
(pages, n, capacity[i]) << "\n";
    return 0;
}

```

OutPut:-

```

Enter Number of Pages : 10

Enter Pages : 1 2 3 4 5 3 2 1 5 4
Frame Size : 2, Page Faults: 10
Frame Size : 3, Page Faults: 9
Frame Size : 4, Page Faults: 7
Frame Size : 5, Page Faults: 5
Frame Size : 6, Page Faults: 5

...Program finished with exit code 0
Press ENTER to exit console.

```

Graph:-

