U18CO018

Shubham Shekhaliya

Assignment – 7 (OS)

1-> To implement Shortest Seek Time First Disk Scheduling Algorithm.

Code:-

#include<bits/stdc++.h>

using namespace std;

int main() {

    int n;

    cout << "Enter number of Requests \n";

    cin >> n;

    cout << "Enter a head \n";

    int head;

    cin >> head;

    list<int> req;

    cout << "Enter All Requests one by one \n";

    for (int i = 0; i < n; i++) {

        int x;

        cin >> x;

        req.push\_back(x);

    }

    vector<int> sequence;

    sequence.push\_back(head);

    int seekCount = 0;

    while (req.size() > 0) {

        auto index = req.begin();

        int minimum = INT\_MAX;

        for (auto it = req.begin(); it != req.end(); it++) {

            if (abs(head - \*it) < minimum) {

                index = it;

                minimum = abs(head - \*it);

            }

        }

        seekCount += minimum;

        head = \*index;

        sequence.push\_back(\*index);

        req.erase(index);

    }

    cout << "Total number of seek operations = " << seekCount << endl;

    cout << "Seek Sequence is \n";

    for (auto it:sequence) {

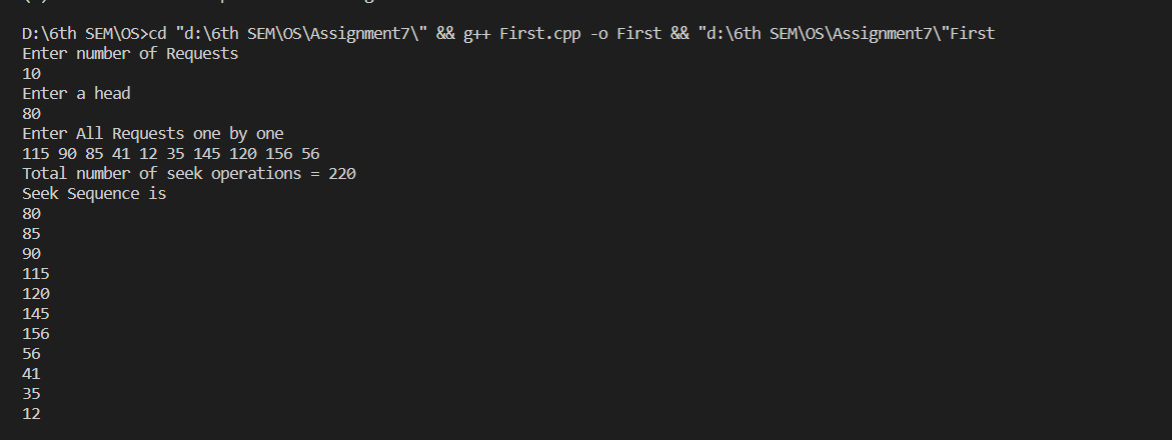
        cout << it << endl;

    }

    return 0;

}

Output:-



2-> To implement SCAN algorithm for Disk Scheduling.

Code:-

#include<bits/stdc++.h>

using namespace std;

const int maxDisk = 200;

int main() {

    int n;

    cout << "Enter number of Requests \n";

    cin >> n;

    int head;

    cout << "Enter a head \n";

    cin >> head;

    int direction;

    cout << "Enter Direction 0 for toward left 1 for toward right \n";

    cin >> direction;

    vector<int> left, right;

    if (direction == 0) left.push\_back(0);

    else right.push\_back(maxDisk - 1);

    cout << "Enter All Requests one by one \n";

    for (int i = 0; i < n; i++) {

        int x;

        cin >> x;

        if (x < head) left.push\_back(x);

        else if (x > head) right.push\_back(x);

    }

    sort(left.begin(), left.end());

    sort(right.begin(), right.end());

    vector<int> sequence;

    sequence.push\_back(head);

    int seekCount = 0;

    for (int i = 0; i < 2; i++) {

        if (direction == 0) {  // left

            for (int i = left.size() - 1; i >= 0; i--) {

                sequence.push\_back(left[i]);

                seekCount += abs(left[i] - head);

                head = left[i];

            }

        } else {   // right

            for (int i = 0; i < right.size(); i++) {

                sequence.push\_back(right[i]);

                seekCount += abs(right[i] - head);

                head = right[i];

            }

        }

        direction ^= 1;

    }

    cout << "Total number of seek operations = " << seekCount << endl;

    cout << "Seek Sequence is \n";

    for (auto it:sequence)

        cout << it << endl;

    return 0;

}

Output:-

