DISK SCHEDULING

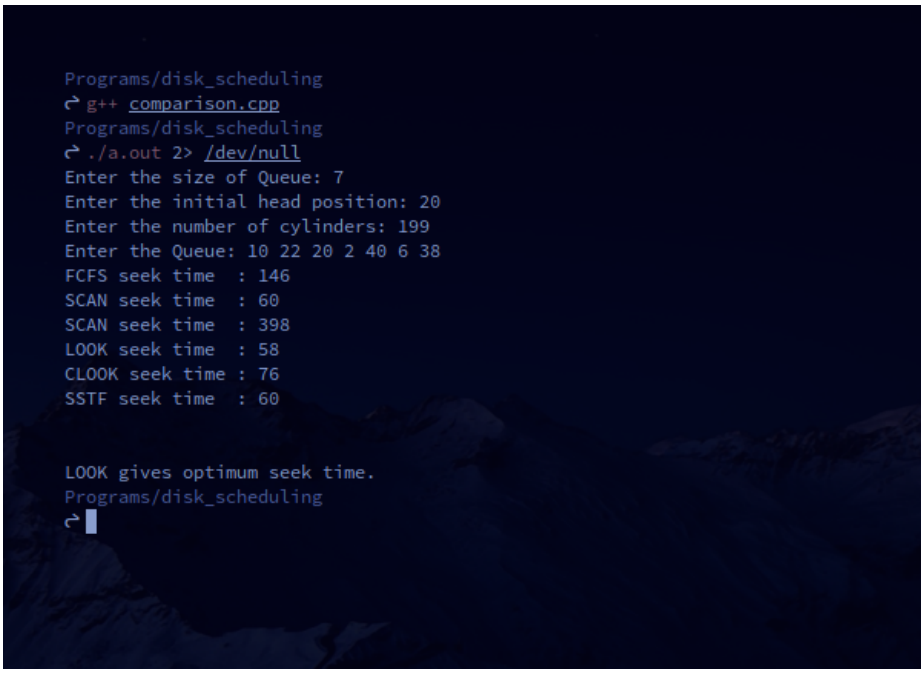
September 8, 2019

Rwithik Manoj
College of Engineering, Trivandrum
Department of Computer Science and Engineering

Comparison

```cpp
#include <bits/stdc++.h>

using namespace std;

namespace fcfs{
  #include "fcfs.cpp"
}
namespace scan{
  #include "scan.cpp"
}
namespace cscan{
  #include "cscan.cpp"
}
namespace look{
  #include "look.cpp"
}
namespace clook{
  #include "clook.cpp"
}
namespace sstf{
  #include "sstf.cpp"
}

int min(int a, int b){
  return (a < b) ? a : b;
}

int main(int argc, char const *argv[]) {
  vector<int> q;
  int head, pos, max;
  int n;

  cout << "Enter the size of Queue: ";
  cin >> n;
  cout << "Enter the initial head position: ";
  cin >> head;
  cout << "Enter the number of cylinders: ";
  cin >> max;
  cout << "Enter the Queue: ";

  for (int i = 0; i < n; ++i){
    cin >> pos;
    q.push_back(pos);
  }

  int fcfs_seek_time = fcfs::seek(n, head, q, false);
  int scan_seek_time = scan::seek(n, head, q, false);
  int cscan_seek_time = cscan::seek(n, head, q, max, false);
```

```
  int look_seek_time = look::seek(n, head, q, false);
  int clook_seek_time = clook::seek(n, head, q, false);
  int sstf_seek_time = sstf::seek(n, head, q, false);

  cout << "FCFS seek time  : " << fcfs_seek_time  << endl;
  cout << "SCAN seek time  : " << scan_seek_time  << endl;
  cout << "SCAN seek time  : " << cscan_seek_time << endl;
  cout << "LOOK seek time  : " << look_seek_time  << endl;
  cout << "CLOOK seek time : " << clook_seek_time << endl;
  cout << "SSTF seek time  : " << sstf_seek_time  << endl;
  cout << "\n\n";
  int smallest = min(min(min(min(min(fcfs_seek_time, scan_seek_time), cscan_seek_time), lo
  cerr << smallest << "<-" << endl;
  if(smallest == fcfs_seek_time){
    cout << "FCFS gives optimum seek time." << endl;
  }
  if(smallest == scan_seek_time){
    cout << "SCAN gives optimum seek time." << endl;
  }
  if(smallest == cscan_seek_time){
    cout << "CSCAN gives optimum seek time." << endl;
  }
  if(smallest == look_seek_time){
    cout << "LOOK gives optimum seek time." << endl;
  }
  if(smallest == clook_seek_time){
    cout << "CLOOK gives optimum seek time." << endl;
  }
  if(smallest == sstf_seek_time){
    cout << "SSTF gives optimum seek time." << endl;
  }

  return 0;
}
```



Header Files

FCFS

```
#include <bits/stdc++.h>

using namespace std;

int seek(int n, int head, vector<int> q, bool print=true){
  int current;
  int total_seek = 0;
  for(int i = 0; i < n; ++i){
    current = q[i];
    if (print)
      cout << "Move from " << head << " to "
                           << current << " with seek "
                           << abs(current - head) << endl;
    total_seek += abs(current - head);
    head = current;
  }
  return total_seek;
}

int main(int argc, char const *argv[]) {
  vector<int> q;
  int head, pos;
```

```
  int n;

  cout << "Enter the size of Queue: ";
  cin >> n;
  cout << "Enter the initial head position: ";
  cin >> head;
  cout << "Enter the Queue: ";

  for (int i = 0; i < n; ++i){
    cin >> pos;
    q.push_back(pos);
  }

  int total_seek = seek(n, head, q);
  cout << "\nTotal seek time is " << total_seek << endl;
  cout << "Average seek time is " << total_seek / float(n) << endl;
  return 0;
}
```

SCAN

```
#include <bits/stdc++.h>

using namespace std;

int seek(int n, int head, vector<int> q, bool print=true){
  std::vector<int> left, right;
  for (auto i: q){
    if (i > head)
      right.push_back(i);
    else
      left.push_back(i);
  }
  left.push_back(0);
  // right.push_back(0);

  sort(left.begin(), left.end(), greater<int>());
  sort(right.begin(), right.end());

  int current;
  int total_seek = 0;

  for(int i: right){
    if (print)
      cout << "Move from " << head << " to " << i
                           << " with seek " << abs(i - head) << endl;
    total_seek += abs(i - head);
    head = i;
  }
```

```
  for(int i: left){
    if (print)
      cout << "Move from " << head << " to " << i
                           << " with seek " << abs(i - head) << endl;
    total_seek += abs(i - head);
    head = i;
  }

  return total_seek;

}

int main(int argc, char const *argv[]) {
  vector<int> q;
  int head, pos;
  int n;

  cout << "Enter the size of Queue: ";
  cin >> n;
  cout << "Enter the initial head position: ";
  cin >> head;
  cout << "Enter the Queue: ";

  for (int i = 0; i < n; ++i){
    cin >> pos;
    q.push_back(pos);
  }

  int total_seek = seek(n, head, q);
  cout << "\nTotal seek time is " << total_seek << endl;
  cout << "Average seek time is " << total_seek / float(n) << endl;
  return 0;
}
```

C-SCAN

```
#include <bits/stdc++.h>

using namespace std;

// #define MAX 199

int seek(int n, int head, vector<int> q, int max, bool print=true){
  std::vector<int> left, right;
  for (auto i: q){
    if (i > head)
      right.push_back(i);
    else
      left.push_back(i);
  }
```

```
  left.push_back(0);
  right.push_back(max);

  sort(left.begin(), left.end());
  sort(right.begin(), right.end());

  int current;
  int total_seek = 0;

  for(int i: right){
    if (print)
      cout << "Move from " << head << " to " << i
                           << " with seek " << abs(i - head) << endl;
    total_seek += abs(i - head);
    head = i;
  }
  for(int i: left){
    if (print)
      cout << "Move from " << head << " to " << i
                           << " with seek " << abs(i - head) << endl;
    total_seek += abs(i - head);
    head = i;
  }

  return total_seek;

}

int main(int argc, char const *argv[]) {
  vector<int> q;
  int head, pos, max;
  int n;

  cout << "Enter the size of Queue: ";
  cin >> n;
  cout << "Enter the initial head position: ";
  cin >> head;
  cout << "Enter the number of cylinders: ";
  cin >> max;
  cout << "Enter the Queue: ";

  for (int i = 0; i < n; ++i){
    cin >> pos;
    q.push_back(pos);
  }

  int total_seek = seek(n, head, q, max);
  cout << "\nTotal seek time is " << total_seek << endl;
  cout << "Average seek time is " << total_seek / float(n) << endl;
  return 0;
```

```
}
```

LOOK

```
#include <bits/stdc++.h>

using namespace std;

int seek(int n, int head, vector<int> q, bool print=true){
  std::vector<int> left, right;
  for (auto i: q){
    if (i > head)
      right.push_back(i);
    else
      left.push_back(i);
  }
  // left.push_back(0);
  // right.push_back(0);

  sort(left.begin(), left.end(), greater<int>());
  sort(right.begin(), right.end());

  int current;
  int total_seek = 0;

  for(int i: right){
    if (print)
      cout << "Move from " << head << " to " << i
                           << " with seek " << abs(i - head) << endl;
    total_seek += abs(i - head);
    head = i;
  }

  for(int i: left){
    if (print)
      cout << "Move from " << head << " to " << i
                           << " with seek " << abs(i - head) << endl;
    total_seek += abs(i - head);
    head = i;
  }



  return total_seek;

}

int main(int argc, char const *argv[]) {
  vector<int> q;
  int head, pos;
```

```
  int n;

  cout << "Enter the size of Queue: ";
  cin >> n;
  cout << "Enter the initial head position: ";
  cin >> head;
  cout << "Enter the Queue: ";

  for (int i = 0; i < n; ++i){
    cin >> pos;
    q.push_back(pos);
  }

  int total_seek = seek(n, head, q);
  cout << "\nTotal seek time is " << total_seek << endl;
  cout << "Average seek time is " << total_seek / float(n) << endl;
  return 0;
}
```

C-LOOK

```
#include <bits/stdc++.h>

using namespace std;

int seek(int n, int head, vector<int> q, bool print=true){
  std::vector<int> left, right;
  for (auto i: q){
    if (i > head)
      right.push_back(i);
    else
      left.push_back(i);
  }

  sort(left.begin(), left.end());
  sort(right.begin(), right.end());

  int current;
  int total_seek = 0;

  for(int i: right){
    if (print)
      cout << "Move from " << head << " to " << i
                           << " with seek " << abs(i - head) << endl;
    total_seek += abs(i - head);
    head = i;
  }
  for(int i: left){
    if (print)
      cout << "Move from " << head << " to " << i
```

```
                          << " with seek " << abs(i - head) << endl;
    total_seek += abs(i - head);
    head = i;
  }

  return total_seek;

}

int main(int argc, char const *argv[]) {
  vector<int> q;
  int head, pos;
  int n;

  cout << "Enter the size of Queue: ";
  cin >> n;
  cout << "Enter the initial head position: ";
  cin >> head;
  cout << "Enter the Queue: ";

  for (int i = 0; i < n; ++i){
    cin >> pos;
    q.push_back(pos);
  }

  int total_seek = seek(n, head, q);
  cout << "\nTotal seek time is " << total_seek << endl;
  cout << "Average seek time is " << total_seek / float(n) << endl;
  return 0;
}
```