

LAB5

Name: Aasem Mohamed Henedy

ID: 5580

Group: 4

Sec: 2

FIFO

It is the simplest form for disk scheduling where we put the head on the first disk and it continuously scans till the end of the disks but it does not provide the fastest service.

```
void FIFO(vector<int> requests, int disk_size, int head_pos, int head_dir){
    requests.insert(requests.begin(), head_pos);
    printReq(requests);
    cout<<endl;
    int head_movements = getHeadMov(requests);
    cout<<"Total head movements = "<<head_movements<<endl;
    float avg_head_mov = float(head_movements) / (requests.size()-1);
    cout<<"Average head movements = "<<avg_head_mov<<endl;
}
```

SCAN

In this method the disk arm starts from first disk and keeps scanning till the other end and servicing requests till it reaches the end then it reverses its direction and do the same operations in the opposite direction.

```
void SCAN(vector<int> requests, int disk_size, int head_pos, int head_dir){
    vector<int>::iterator it;
    if(head_dir == 1)
        sort(requests.begin(),requests.end());
    else
        sort(requests.begin(),requests.end(),greater<>());
    it = find(requests.begin(), requests.end(), head_pos);
    vector<int> req;
    if(it == requests.end()){
```

```

    req.push_back(head_pos);

    if(head_dir == 1)
        it = lower_bound(requests.begin(), requests.end(), head_pos);
    else
        it = upper_bound(requests.begin(), requests.end(), head_pos, greater<>());
}

int idx;

if(it == requests.end())
    idx = requests.size()-1;
else
    idx = it-requests.begin();

for(int i = idx ; i < requests.size() ; i++)
    req.push_back(requests[i]);
for(int i = idx-1 ; i >=0 ; i--)
    req.push_back(requests[i]);

printReq(req);

cout<<endl;

int head_movements = getHeadMov(req);

cout<<"Total head movements = "<<head_movements<<endl;

float avg_head_mov = (float)head_movements / requests.size();

cout<<"Average head movements = "<<avg_head_mov<<endl;
}

```

C-SCAN

This method is like SCAN method but the difference between them is when the disk arm reaches the other end and been reversed it does not serve any request in the opposite direction but it just pass through disks till it returns to the first end again.

```
void CSCAN(vector<int> requests, int disk_size, int head_pos, int head_dir){  
    vector<int>::iterator it;  
    if(head_dir == 1)  
        sort(requests.begin(),requests.end());  
    else  
        sort(requests.begin(),requests.end(),greater<>());  
    it = find(requests.begin(), requests.end(), head_pos);  
    vector<int> req;  
    if(it == requests.end()){  
        req.push_back(head_pos);  
        if(head_dir == 1)  
            it = lower_bound(requests.begin(), requests.end(), head_pos);  
        else  
            it = upper_bound(requests.begin(), requests.end(), head_pos,greater<>());  
    }  
    int idx;  
    if(it == requests.end())  
        idx = requests.size()-1;  
    else  
        idx = it-requests.begin();  
  
    for(int i = idx ; i < requests.size() ; i++)  
        req.push_back(requests[i]);  
    for(int i = 0 ; i < idx ; i++)  
        req.push_back(requests[i]);
```

```

printReq(req);

cout<<endl;

int head_movements = getHeadMov(req);

cout<<"Total head movements = "<<head_movements<<endl;

float avg_head_mov = (float)head_movements / requests.size();

cout<<"Average head movements = "<<avg_head_mov<<endl;
}

```

OUTPUT

```

main.cpp [lab5] - Code::Blocks 20.03
File Edit View Search Project Build Debug
main.cpp
107 int disk_size, requests_num, head_pos, head_dir;
108 vector<int> requests;
109
110 cout<<"Enter the size of disk: ";
111 cin>>disk_size;
112 cout<<"Enter number of requests: ";
113 cin>>requests_num;
114 cout<<"Enter the requests: ";
115 for(int i = 0; i < requests_num; i++)
116 {
117     int temp;
118     cin >> temp;
119     requests.push_back(temp);
120 }
121 cout<<"Enter the head position: ";
122 cin >> head_pos;
123 cout<<"Enter the head direction: ";
124 cin >> head_dir;
125
126 cout<<"FIFO:\n-----";
127 FIFO(requests, disk_size, head_pos, head_dir);
128 cout<<"\nSCAN:\n-----";
129 SCAN(requests, disk_size, head_pos, head_dir);
130 cout<<"\nCSCAN:\n-----";
131 CSCAN(requests, disk_size, head_pos, head_dir);
132 return 0;

```

```

Enter the size of disk: 200
Enter number of requests: 10
Enter the requests: 27 129 110 186 147 41 10 64 120
Enter the head position: 100
Enter the head direction: 1
FIFO:
100 -> 27 -> 129 -> 110 -> 186 -> 147 -> 41 -> 10 -> 64 -> 120
Total head movements = 556
Average head movements = 61.7778
SCAN:
100 -> 110 -> 120 -> 129 -> 147 -> 186 -> 64 -> 41 -> 27 -> 10
Total head movements = 262
Average head movements = 29.1111
CSCAN:
100 -> 110 -> 120 -> 129 -> 147 -> 186 -> 10 -> 27 -> 41 -> 64
Total head movements = 316
Average head movements = 35.1111
Process returned 0 (0x0)   execution time : 00.065 s
Press any key to continue.

```