Garv nanwani
19BCS049

Question 15 (a)

Write a program to implement the C-SCAN elevator disk scheduling algorithm. The program should give detail about each disk movement from starting head position (input from the user) and calculate average head movement.

Code :

```
#include<stdio.h>
#include<math.h>
#include <bits/stdc++.h>
using namespace std;
int main()
{
int n,head,i,j,k,seek=0,max,diff,curr;
float avg;
string direction;
vector<int> left, right;
printf("Enter the max range of disk\n");
scanf("%d",&max);
printf("Enter the size of queue request\n");
scanf("%d",&n);
int queue[n];
printf("Enter the queue of disk positions to be read\n");
for(i=0;i<n;i++)
scanf("%d",&queue[i]);
printf("Enter the direction\n");
cin>>direction;
printf("Enter the initial head position\n");
scanf("%d",&head);
left.push_back(0);
right.push_back(max - 1);
for (i = 0; i < n; i++)
{
if (queue[i] <= head)
left.push_back(queue[i]);
if (queue[i] > head)
right.push_back(queue[i]);
}
std::sort(left.begin(), left.end());
std::sort(right.begin(), right.end());
printf("Disk head moves from \t to \t with seek\n" );
int run = 2;
while (run-->0) {if (direction == "left") {
for ( i = left.size() - 1; i >= 0; i--) {
curr = left[i];
diff = abs(curr - head);
printf("%d \t\t %d \t %d\n",head,curr,diff);
seek += diff;
```

```
head = curr;
}
direction = "right";
std::reverse(right.begin(), right.end());
}
else if (direction == "right") {
for ( i = 0; i < right.size(); i++) {
curr = right[i];
diff = abs(curr - head);
printf("%d \t\t %d \t %d\n",head,curr,diff);
seek += diff;
head = curr;
}
direction = "left";
std::reverse(left.begin(), left.end());
}
}
printf("Total seek time is %d\n",seek);
avg=seek/(float)n;
printf("Average seek time is %f\n",avg);
return 0;
}
```

Output :

```
Enter the max range of disk
200
Enter the size of queue request
8
Enter the queue of disk positions to be read
14 34 56 77 86 34 97 150
Enter the direction
left
Enter the initial head position
56
Disk head moves from     to      with seek
56               56      0
56               34      22
34               34      0
34               14      20
14               0       14
0                199     199
199              150     49
150              97      53
97               86      11
86               77      9
Total seek time is 377
Average seek time is 47.125000
```

Question 15 (b)

Write a program to implement the C-LOOK elevator disk scheduling algorithm. The program should give detail about each disk movement from starting head position (input from the user) and calculate average head movement.

Code :

```
#include<stdio.h>
#include<math.h>
#include <bits/stdc++.h>
using namespace std;
int main()
{
int n,head,i,j,k,seek=0,max,diff,curr;
float avg;
string direction;
vector<int> left, right;
printf("Enter the max range of disk\n");
scanf("%d",&max);
printf("Enter the size of queue request\n");
scanf("%d",&n);
int queue[n];
printf("Enter the queue of disk positions to be read\n");
for(i=0;i<n;i++)
scanf("%d",&queue[i]);
printf("Enter the direction\n");
cin>>direction;
printf("Enter the initial head position\n");
scanf("%d",&head);
for (i = 0; i < n; i++)
{
if (queue[i] <= head)
left.push_back(queue[i]);
if (queue[i] > head)
right.push_back(queue[i]);
}
std::sort(left.begin(), left.end());
std::sort(right.begin(), right.end());
printf("Disk head moves from \t to \t with seek\n" );
int run = 2;
while (run-->0) {
if (direction == "left") {
for ( i = left.size() - 1; i >= 0; i--) {curr = left[i];
diff = abs(curr - head);
printf("%d \t\t %d \t %d\n",head,curr,diff);
seek += diff;
head = curr;
}
direction = "right";
std::reverse(right.begin(), right.end());
}
else if (direction == "right") {
for ( i = 0; i < right.size(); i++) {
```

```
curr = right[i];
diff = abs(curr - head);
printf("%d \t\t %d \t %d\n",head,curr,diff);
seek += diff;
head = curr;
}
direction = "left";
std::reverse(left.begin(), left.end());
}
}
printf("Total seek time is %d\n",seek);
avg=seek/(float)n;
printf("Average seek time is %f\n",avg);
return 0;
}
```

Output :

```
Enter the max range of disk
200
Enter the size of queue request
8
Enter the queue of disk positions to be read
14 34 56 77 86 34 97 150
Enter the direction
left
Enter the initial head position
54
Disk head moves from     to      with seek
54               34      20
34               34      0
34               14      20
14               150     136
150              97      53
97               86      11
86               77      9
77               56      21
Total seek time is 270
Average seek time is 33.750000
```