**AIM:**

To implement **C-SCAN** disk scheduling algorithm .

**OBJECTIVE :**

To understand the concept of disk scheduling and the implementation of C-SCAN disk scheduling algorithm.

**SOFTWARE REQUIRED :**

OS – Windows

Software – Putty

**DESCRIPTION :**

Disk schedulingis is done by operating systems to schedule I/O requests arriving for disk. Disk scheduling is also known as I/O scheduling.

Disk scheduling is important because:

* Multiple I/O requests may arrive by different processes and only one I/O request can be served at a time by disk controller. Thus other I/O requests need to wait in waiting queue and need to be scheduled.
* Two or more request may be far from each other so can result in greater disk arm movement.
* Hard drives are one of the slowest parts of computer system and thus need to be accessed in an efficient manner.

In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.

These situations are avoided in *C-SCAN*algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to SCAN algorithm and hence it is known as C-SCAN (Circular SCAN).

Advantages:

* Provides more uniform wait time compared to SCAN
* It improves the unfair situation of the end tracks against the middle tracks .

**ALGORITHM :**

Step 1: Start the program.

Step 2: Enter the maximum range of disk.

Step 3: Enter the initial head position.

Step 4: Enter the size of queue request.

Step 5: Enter the queue of disk positions to be read.

Step 6: The C-SCAN algorithm is performed and the movement of disk head for each

position , the seek time is calculated.

Step 7: The total seek time and the average seek time is computed .

Step 8: Stop the program .

**PROGRAM:**

#include<stdio.h>

int main()

{

int queue[20],n,head,i,j,k,seek=0,max,diff,temp,queue1[20],queue2[20],temp1=0,temp2=0;

float avg;

printf("Enter the max range of disk\n");

scanf("%d",&max);

printf("Enter the initial head position\n");

scanf("%d",&head);

printf("Enter the size of queue request\n");

scanf("%d",&n);

printf("Enter the queue of disk positions to be read\n");

for(i=1;i<=n;i++)

{

scanf("%d",&temp);

if(temp>=head)

{

queue1[temp1]=temp;

temp1++;

}

else

{

queue2[temp2]=temp;

temp2++;

}

}

for(i=0;i<temp1-1;i++)

{

for(j=i+1;j<temp1;j++)

{

if(queue1[i]>queue1[j])

{

temp=queue1[i];

queue1[i]=queue1[j];

queue1[j]=temp;

}

}

}

for(i=0;i<temp2-1;i++)

{

for(j=i+1;j<temp2;j++)

{

if(queue2[i]>queue2[j])

{

temp=queue2[i];

queue2[i]=queue2[j];

queue2[j]=temp;

}

}

}

for(i=1,j=0;j<temp1;i++,j++)

queue[i]=queue1[j];

queue[i]=max;

queue[i+1]=0;

for(i=temp1+3,j=0;j<temp2;i++,j++)

queue[i]=queue2[j];

queue[0]=head;

for(j=0;j<=n+1;j++)

{

diff=abs(queue[j+1]-queue[j]);

seek+=diff;

printf("Disk head moves from %d to %d with seek %d\n",queue[j],queue[j+1],diff);

}

printf("Total seek time is %d\n",seek);

avg=seek/(float)n;

printf("Average seek time is %f\n",avg);

return 0;

}

**OUTPUT:**

[cse15@CSLab2 ~]$ vi pro.c

[cse15@CSLab2 ~]$ cc pro.c

[cse15@CSLab2 ~]$ ./a.out

Enter the max range of disk

200

Enter the initial head position

53

Enter the size of queue request

8

Enter the queue of disk positions to be read

98

183

31

122

14

124

65

67

Disk head moves from 53 to 65 with seek12

Disk head moves from 65 to 67 with seek2

Disk head moves from 67 to 98 with seek31

Disk head moves from 98 to 122 with seek24

Disk head moves from 122 to 124 with seek2

Disk head moves from 124 to 183 with seek59

Disk head moves from 183 to 200 with seek17

Disk head moves from 200 to 0 with seek200

Disk head moves from 0 to 14 with seek14

Disk head moves from 14 to 31 with seek17

Total seek time is 378

Average seek time is 47.250000

**RESULT :**

The implementation of C-SCAN disk scheduling algorithm was executed .