

**Lab no: 8** **Date: 2079/**

**Title: Write a program to find the number of movements made by r/w head for**

**C-LOOK algorithm.**

**C-LOOK:**

 CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm in spite of going to the end goes only to the last request to be serviced in front of the head and then from there goes to the other end’s last request. Thus, it also prevents the extra delay which occurred due to unnecessary traversal to the end of the disk.

**Algorithm:**

Step 1: Let Request array represents an array storing indexes of the tracks that have

been requested in ascending order of their time of arrival and head is

the position of the disk head.

Step 2: The initial direction in which the head is moving is given and it services in

the same direction.

Step 3: The head services all the requests one by one in the direction it is moving.

Step 4: The head continues to move in the same direction until all the requests in

this direction have been serviced.

Step 5: While moving in this direction, calculate the absolute distance of the tracks

from the head.

Step 6: Increment the total seek count with this distance.

Step 7: Currently serviced track position now becomes the new head position.

Step 8: Go to step 5 until we reach the last request in this direction.

Step 9: If we reach the last request in the current direction then reverse the direction

and move the head in this direction until we reach the last request that is needed

to be serviced in this direction without servicing the intermediate requests.

Step 10: Reverse the direction and go to step 3 until all the requests have not

been serviced.

**Programming Language: C**

**IDE: Dev-C++**

**Source Code:**

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  int main()  {  int REQUEST[100],i,j,n,TotalHeadMovement=0,initial,size,move;  printf("Enter the number of Requests\n");  scanf("%d",&n);  printf("Enter the Requests sequence\n");  for(i=0;i<n;i++)  scanf("%d",&REQUEST[i]);  printf("Enter initial head position\n");  scanf("%d",&initial);  printf("Enter total disk size\n");  scanf("%d",&size);  printf("Enter the head movement direction for high 1 and for low 0\n");  scanf("%d",&move);    for(i=0;i<n;i++)  {  for( j=0;j<n-i-1;j++)  {  if(REQUEST[j]>REQUEST[j+1])  {  int temp;  temp=REQUEST[j];  REQUEST[j]=REQUEST[j+1];  REQUEST[j+1]=temp;  }  }  }  int index;  for(i=0;i<n;i++)  {  if(initial<REQUEST[i])  {  index=i;  break;  }  }    //for high move  if(move==1)  {  for(i=index;i<n;i++)  {  TotalHeadMovement=TotalHeadMovement+abs(REQUEST[i]-initial);  initial=REQUEST[i];  }    for( i=0;i<index;i++)  {  TotalHeadMovement=TotalHeadMovement+abs(REQUEST[i]-initial);  initial=REQUEST[i];    }  }  //for low move  else  {  for(i=index-1;i>=0;i--)  {  TotalHeadMovement=TotalHeadMovement+abs(REQUEST[i]-initial);  initial=REQUEST[i];  }    for(i=n-1;i>=index;i--)  {  TotalHeadMovement=TotalHeadMovement+abs(REQUEST[i]-initial);  initial=REQUEST[i];    }  }    printf("Total head movement is %d",TotalHeadMovement);  return 0;  } |

**Output:**

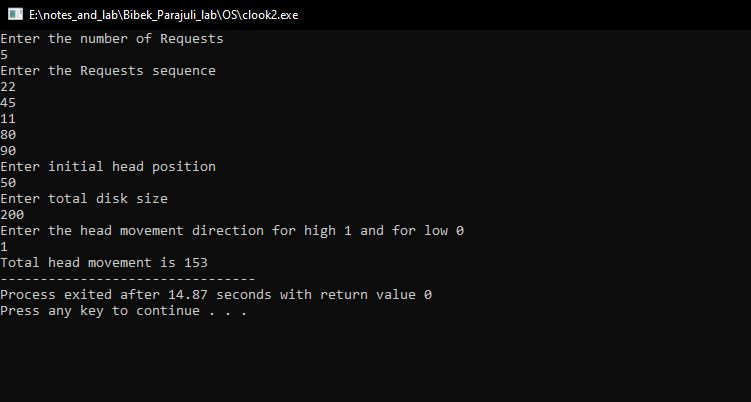


Fig: C-LOOK disk scheduling algorithm