|  |
| --- |
| **EX NO.:12 DISK SCHEDULING ALGORITHM**  **DATE** |

**AIM:**

To implement the various disc scheduling algorithms

**1.FCFS:**

**CODING:**

#include <stdio.h>

#include <stdlib.h>

int main(){

int queue[100], q\_size, head, seek =0, diff;

float avg;

printf("\_\_\_DISK SCHEDULING\_\_\_(FCFS)\n\n");

printf("%s\n", "Enter the size of the queue");

scanf("%d", &q\_size);

printf("%s\n", "Enter queue elements");

for(int i=1; i<=q\_size; i++){

scanf("%d",&queue[i]);

}

printf("%s\n","Enter initial head position");

scanf("%d", &head);

queue[0]=head;

for(int j=0; j<=q\_size-1; j++){

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

seek += diff;

}

printf("\nNumber of cylinders are %d\t",seek);

return 0;}

**OUTPUT:**

\_\_\_DISK SCHEDULING\_\_\_(FCFS)

Enter the size of the queue

9

Enter queue elements

246 1105 2153 324 21 1739 32 2467 500

Enter initial head position

356

Number of cylinders are 11976

**2.CLOOK:**

**CODING:**

#include<stdio.h>

#include<stdlib.h>

int main(){

int n, i, j, head, item[20], dst[20];

int cylinders=0;

printf("\_\_\_DISK SCHEDULING\_\_\_(C-LOOK)\n\n");

printf("Enter no. of locations:");

scanf("%d",&n);

printf("Enter position of head:");

scanf("%d",&head);

printf("Enter elements of disk queue:");

for(i=0;i<n;i++)

{

scanf("%d",&item[i]);

dst[i]=(head-item[i]);

}

//Selection Sort

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

{

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

{

if(dst[j]<dst[i])

{

int temp=dst[j];

dst[j]=dst[i];

dst[i]=temp;

temp=item[i];

item[i]=item[j];

item[j]=temp;

}

}

}

for(i=0;i<n;i++)

{

if(item[i]<=head)

{

j=i;

break;

}

}

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

{

cylinders+= abs(head-item[i]);

head=item[i];

}

for(i=0;i<j;i++)

{

cylinders+= abs(head-item[i]);

head=item[i];

}

printf("\n\nNumber of cylinders are %d\n\n", cylinders );}

**OUTPUT:**

\_\_\_DISK SCHEDULING\_\_\_(C-LOOK)

Enter no. of locations:9

Enter position of head:356

Enter elements of disk queue:246 1105 2153 324 21 1739 32 2467 500

Number of cylinders are 4748

**3.SSTF:**

**CODING:**

#include<stdio.h>

struct head

{

int num;

int flag;

};

int main()

{

struct head h[33];

int array\_1[33], array\_2[33];

int count = 0, j, x, limit, minimum, location, disk\_head, sum = 0;

printf("\_\_\_DISK SCHEDULING\_\_\_(SSTF)\n\n");

printf("\nEnter total number of locations:\t");

scanf("%d", &limit);

printf("\nEnter position of disk head:\t");

scanf("%d", &disk\_head);

printf("\nEnter elements of disk head queue\n");

while(count < limit)

{

scanf("%d", &h[count].num);

h[count].flag = 0;

count++;

}

for(count = 0; count < limit; count++)

{

x = 0;

minimum = 0;

location = 0;

for(j = 0; j < limit; j++)

{

if(h[j].flag == 0)

{

if(x == 0)

{

array\_1[j] = disk\_head - h[j].num;

if(array\_1[j] < 0)

{

array\_1[j] = h[j].num - disk\_head;

}

minimum = array\_1[j];

location = j;

x++;

}

else

{

array\_1[j] = disk\_head - h[j].num;

if(array\_1[j] < 0)

{

array\_1[j] = h[j].num - disk\_head;

}

}

if(minimum > array\_1[j])

{

minimum = array\_1[j];

location = j;

}

}

}

h[location].flag = 1;

array\_2[count] = h[location].num - disk\_head;

if(array\_2[count] < 0)

{

array\_2[count] = disk\_head - h[location].num;

}

disk\_head = h[location].num;

}

count = 0;

while(count < limit)

{

sum = sum + array\_2[count];

count++;

}

printf("\nTotal movements of the cylinders:\t%d", sum);

return 0;

}

**OUTPUT:**

\_\_\_DISK SCHEDULING\_\_\_(SSTF)

Enter total number of locations: 9

Enter position of disk head: 356

Enter elements of disk head queue

246 1105 2153 324 21 1739 32 2467 500

Total movements of the cylinders: 2781

**4.CSCAN:**

**CODING:**

#include <stdlib.h>

#include <stdio.h>

#define HIGH 2499

#define LOW 0

int main(){

int queue[20], q\_size, head, i,j, seek=0, diff, max, temp, queue1[20], queue2[20], temp1=0,

temp2=0;

float avg;

printf("\_\_\_DISK SCHEDULING\_\_\_(C SCAN)\n\n");

printf("%s\t", "Input no of disk locations");

scanf("%d", &q\_size);

printf("%s\t", "Enter initial head position");

scanf("%d", &head);

printf("%s\n","Enter disk positions to be read");

for(i=0; i<q\_size; 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;

}

}

}

if(abs(head-LOW) >= abs(head-HIGH)){

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

queue[i] = queue1[j];

}

queue[i] = HIGH;

queue[i+1] = 0;

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

queue[i] = queue2[j];

}

} else {

for(i=1,j=temp2-1; j>=0; i++,j--){

queue[i] = queue2[j];

}

queue[i] = LOW;

queue[i+1] = HIGH;

for(i=temp2+3, j=temp1-1; j>=0; i++, j--){

queue[i] = queue1[j];

}

}

queue[0] = head;

for(j=0; j<=q\_size+1; j++){

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

seek += diff;

}

printf("The number of cylinders are %d\n", seek);

return 0;}

**OUTPUT:**

\_\_\_DISK SCHEDULING\_\_\_(C SCAN)

Input no of disk locations 9

Enter initial head position 356

Enter disk positions to be read

246 1105 2153 324 21 1739 32 2467 500

The number of cylinders are 4854

**RESULT:**

Thus the disk scheduling algorithms are studied and the output is obtained successfully.