

**Dr. D.Y.Patil Arts, Commerce , Science College,
Pimpri, Pune-18
T.Y.B.Sc(Comp. Sci) 2022-23
Operating System-II
Solution of Practical Assignment 3: Disk Scheduling**

- 1) Write a simulation program for disk scheduling using FCFS algorithm. Accept total number of disk blocks, disk request string, and current head position from the user. Display the list of request in the order in which it is served. Also display the total number of head moments.

```
#include<stdio.h>
int ReqString[20],nor,nob,start,thm;
main()
{
    int i,j,k;
    printf("\nEnter No.of Requests: ");
    scanf("%d",&nor);
    printf("\nEnter Requests:\n");
    for(i=0;i<nor;i++)
    {
        printf("[%d]=",&i);
        scanf("%d",&ReqString[i]);
    }
    printf("\nEnter No.of Cylinders: ");
    scanf("%d",&nob);
    printf("\nEnter Start Block: ");
    scanf("%d",&start);

    for(i=0;i<nor;i++)
    {
        printf("\n%d-%d",start,ReqString[i]);
        if(ReqString[i]>=start)
        {
            thm+=ReqString[i]-start;
            start=ReqString[i];
        }
        else
        {
            thm+=start-ReqString[i];
            start=ReqString[i];
        }
    }

    printf("\nTotal Head Movement: %d",thm);
}
/*
Enter No.of Requests: 8
```

Enter Requests:

```
[0]=98
[1]=183
[2]=37
[3]=122
[4]=14
[5]=124
[6]=65
```

[7]=67

Enter No.of Cylinders: 200

Enter Start Block: 53

53-98

98-183

183-37

37-122

122-14

14-124

124-65

65-67

Total Head Movement: 640

*/

- 2) Write a simulation program for disk scheduling using SSTF algorithm. Accept total number of disk blocks, disk request string, and current head position from the user. Display the list of request in the order in which it is served. Also display the total number of head moments.
-

```
#include<stdio.h>
```

```
int ReqString[20],nor,nob,start,thm;
```

```
void sort(int RS[])
```

```
{
```

```
    int i,j,temp;
```

```
    for(i=1;i<nor;i++)
```

```
    {
```

```
        for(j=0;j<nor-i;j++)
```

```
        {
```

```
            if(RS[j]>RS[j+1])
```

```
            {
```

```
                temp=RS[j];
```

```
                RS[j]=RS[j+1];
```

```
                RS[j+1]=temp;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
void display(int RS[])
```

```
{
```

```
    int i;
```

```
    for(i=0;i<nor;i++)
```

```
        printf("  %d",RS[i]);
```

```
}
```

```
int searchstartblock(int RS[])
```

```
{
```

```
    int i;
```

```
    for(i=0;i<nor;i++)
```

```
    {
```

```
        if(RS[i]==start)
```

```
            return i;
```

```
    }
```

```

        return 1;
    }

main()
{
    int
    i,j,k,ans,finish=0,leftpos,startpos,rightpos,leftdis,rightdis,initpos,f
    lag=0;

    printf("\nEnter No.of Requests: ");
    scanf("%d",&nor);
    printf("\nEnter Requests:\n");
    for(i=0;i<nor;i++)
    {
        printf("[%d]=",i);
        scanf("%d",&ReqString[i]);
    }
    printf("\nEnter No.of Cylinders: ");
    scanf("%d",&nob);
    printf("\nEnter Start Block: ");
    scanf("%d",&start);
    ans=searchstartblock(ReqString);
    if(ans==1)
        ReqString[nor++]=start;
    sort(ReqString);
    printf("\nReqString After Sorting: ");
    display(ReqString);
    startpos=searchstartblock(ReqString);
    initpos=startpos;
    leftpos=initpos-1;
    rightpos=initpos+1;
    while(1)
    {
        leftdis=ReqString[startpos]-ReqString[leftpos];
        rightdis=ReqString[rightpos]-ReqString[startpos];

        if(leftdis<rightdis)
        {
            printf("\n%d-%d",ReqString[startpos],ReqString[leftpos]);
            thm+=ReqString[startpos]-ReqString[leftpos];
            startpos=leftpos;
            leftpos--;
            if(leftpos==-1)
            {
                break;
            }
        }
        else if(leftdis>rightdis)
        {
            printf("\n%d-
%d",ReqString[rightpos],ReqString[startpos]);
            thm+=ReqString[rightpos]-ReqString[startpos];
            startpos=rightpos;
            rightpos++;
            if(rightpos==nor)

```

```

        {
            break;
        }
    }

    printf("\nEnd of while loop");
    while(leftpos>=0)
    {
        printf("\n%d-%d",ReqString[startpos],ReqString[leftpos]);
        thm+=ReqString[startpos]-ReqString[leftpos];
        startpos=leftpos;
        leftpos--;
    }
    while(rightpos<nor)
    {
        printf("\n%d-%d",ReqString[rightpos],ReqString[startpos]);
        thm+=ReqString[rightpos]-ReqString[startpos];
        startpos=rightpos;
        rightpos++;
    }
    printf("\nTotal Head Movement: %d",thm);
}

```

```

/*
[root@localhost disc shedding]# cc sstf.c
[root@localhost disc shedding]# ./a.out

```

Enter No.of Requests: 7

Enter Requests:

```

[0]=70
[1]=91
[2]=95
[3]=102
[4]=130
[5]=147
[6]=170

```

Enter No.of Cylinders: 200

Enter Start Block: 125

```

RefString After Sorting:  70  91  95  102  125  130  147  170
130-125
147-130
170-147
End of while loop
170-102
102-95
95-91
91-70

```

```

Total Head Movement: 145[root@localhost disc shedding]# cc sstf.c
[root@localhost disc shedding]# ./a.out

```

Enter No.of Requests: 8

Enter Requests:

[0]=11
[1]=34
[2]=62
[3]=64
[4]=95
[5]=119
[6]=123
[7]=180

Enter No.of Cylinders: 200

Enter Start Block: 50\

RefString After Sorting: 11 34 50 62 64 95 119 123 180
62-50
64-62
64-34
34-11
End of while loop
95-11
119-95
123-119
180-123
Total Head Movement: 236*/

-
- 3) Write a simulation program for disk scheduling using SCAN algorithm. Accept total number of disk blocks, disk request string, and current head position from the user. Display the list of request in the order in which it is served. Also display the total number of head moments.
-

```
//scan
#include<stdio.h>

int ReqString[20],nor,nob,start,thm,min[10],max[10];
char direction;

int getmin()
{
    int i,j=0,min=999;
    for(i=0;i<nor;i++)
        if(ReqString[i]<=min)
            min=ReqString[i];
    return min;
}

int getmax()
{
    int i,j=0,max=0;
    for(i=0;i<nor;i++)
```

```

        if(ReqString[i]>max)
            max=ReqString[i];
    return max;
}

main()
{
    int i,j,k,max,min;
    printf("\nEnter No.of Requests: ");
    scanf("%d",&nor);
    printf("\nEnter Requests:\n");
    for(i=0;i<nor;i++)
    {
        printf("[%d]=",i);
        scanf("%d",&ReqString[i]);
    }
    printf("\nEnter No.of Cylinders: ");
    scanf("%d",&nob);
    printf("\nEnter Start Block: ");
    scanf("%d",&start);
    printf("\nEnter Direction: ");
    scanf(" %c",&direction);
    min=getmin();
    max=getmax();
    if(direction=='L')
    {
        printf("\n%d-0",start);
        thm+=start-0;
        start=0;
        printf("\n%d-%d",max,start);
        thm+=max-start;
    }
    else if(direction=='R')
    {
        printf("\n%d-%d",start,nob-1);
        thm+=(nob-1)-start;
        start=nob-1;
        printf("\n%d-%d",start,min);
        thm+=start-min;
    }
    printf("\nTotal Head Movement: %d",thm);
}
/*
[root@localhost disc sheduling]# cc scan.c
[root@localhost disc sheduling]# ./a.out

```

Enter No.of Requests: 8

Enter Requests:

```

[0]=95
[1]=180
[2]=34
[3]=119
[4]=11

```

```
[5]=123
[6]=62
[7]=64
```

Enter No.of Cylinders: 200

Enter Start Block: 50

Enter Direction: L

50-0

180-0

Total Head Movement: 230

*/

-
- 4) Write a simulation program for disk scheduling using LOOK algorithm. Accept total number of disk blocks, disk request string, and current head position from the user. Display the list of request in the order in which it is served. Also display the total number of head moments.
-

```
//LOOK
#include<stdio.h>

int ReqString[20],nor,nob,start,thm,min[10],max[10];
char direction;

int getmin()
{
    int i,j=0,min=999;
    for(i=0;i<nor;i++)
        if(ReqString[i]<=min)
            min=ReqString[i];
    return min;
}

int getmax()
{
    int i,j=0,max=0;
    for(i=0;i<nor;i++)
        if(ReqString[i]>max)
            max=ReqString[i];
    return max;
}

main()
{
    int i,j,k,max,min;
    printf("\nEnter No.of Requests: ");
    scanf("%d",&nor);
    printf("\nEnter Requests:\n");
    for(i=0;i<nor;i++)
    {
```

```

        printf("[%d]=",i);
        scanf("%d",&ReqString[i]);
    }
    printf("\nEnter No.of Cylinders: ");
    scanf("%d",&nob);
    printf("\nEnter Start Block: ");
    scanf("%d",&start);
    printf("\nEnter Direction: ");
    scanf(" %c",&direction);

    min=getmin();
    max=getmax();
    if(direction=='L')
    {
        printf("\n%d-%d",start,min);
        thm+=start-min;
        start=min;
        printf("\n%d-%d",max,start);
        thm+=max-start;
    }
    else if(direction=='R')
    {
        printf("\n%d-%d",start,max);
        thm+=max-start;
        start=max;
        printf("\n%d-%d",start,min);
        thm+=start-min;
    }
    printf("\nTotal Head Movement: %d",thm);
}
/*
[root@localhost disc sheduling]# cc look.c
[root@localhost disc sheduling]# ./a.out

```

Enter No.of Requests: 8

Enter Requests:

```

[0]=95
[1]=180
[2]=34
[3]=119
[4]=11
[5]=123
[6]=62
[7]=64

```

Enter No.of Cylinders: 200

Enter Start Block: 50

Enter Direction: L

50-11

180-11

Total Head Movement: 208

*/

-
- 5) Write a simulation program for disk scheduling using C-SCAN algorithm. Accept total number of disk blocks, disk request string, and current head position from the user. Display the list of request in the order in which it is served. Also display the total number of head moments.
-

```
//cscan
#include<stdio.h>

int ReqString[20],nor,nob,start,thm,min[10],max[10];
char direction;

void sort(int RS[])
{
    int i,j,temp;
    for(i=1;i<nor;i++)
    {
        for(j=0;j<nor-i;j++)
        {
            if(RS[j]>RS[j+1])
            {
                temp=RS[j];
                RS[j]=RS[j+1];
                RS[j+1]=temp;
            }
        }
    }
}

main()
{
    int i,j,k,spos;
    printf("\nEnter No.of Requests: ");
    scanf("%d",&nor);
    printf("\nEnter Requests:\n");
    for(i=0;i<nor;i++)
    {
        printf("[%d]= ",i);
        scanf("%d",&ReqString[i]);
    }
    printf("\nEnter No.of Cylinders: ");
    scanf("%d",&nob);
    printf("\nEnter Start Block: ");
    scanf("%d",&start);
    ReqString[nor++]=start;
    sort(ReqString);
    for(i=0;i<nor;i++)
        printf(" %d",ReqString[i]);
}
```

```

        for(spos=0;spos<=nor && ReqString[spos]!=start; spos++);
printf("\nEnter Direction: ");
scanf(" %c",&direction);
if(direction=='L')
{
    printf("\n%d-0",start);
    thm+=start-0;
    printf("\n%d-%d",nob-1,ReqString[spos+1]);
    thm+=(nob-1)-ReqString[spos+1];
}
else if(direction=='R')
{
    printf("\n%d-%d",nob-1,start);
    thm+=(nob-1)-start;
    printf("\n%d-0",ReqString[spos-1]);
    thm+=ReqString[spos-1]-0;
}
printf("\nTotal Head Movement: %d",thm);
}
/*
[root@localhost disc sheduling]# cc cscan.c
[root@localhost disc sheduling]# ./a.out

```

Enter No.of Requests: 8

Enter Requests:

```

[0]=95
[1]=180
[2]=34
[3]=119
[4]=11
[5]=123
[6]=62
[7]=64

```

Enter No.of Cylinders: 200

Enter Start Block: 50

11 34 50 62 64 95 119 123 180

Enter Direction: L

50-0

199-62

Total Head Movement: 187

*/

- 6) Write a simulation program for disk scheduling using C-LOOK algorithm. Accept total number of disk blocks, disk request string, and current head position from the user. Display

the list of request in the order in which it is served. Also display the total number of head moments.

```
#include<stdio.h>
int ReqString[20],nor,nob,start,thm,min[10],max[10];
char direction;

int getmin()
{
    int i,j=0,min=999;
    for(i=0;i<nor;i++)
        if(ReqString[i]<=min)
            min=ReqString[i];
    return min;
}

int getmax()
{
    int i,j=0,max=0;
    for(i=0;i<nor;i++)
        if(ReqString[i]>max)
            max=ReqString[i];
    return max;
}

void sort(int RS[])
{
    int i,j,temp;
    for(i=1;i<nor;i++)
    {
        for(j=0;j<nor-i;j++)
        {
            if(RS[j]>RS[j+1])
            {
                temp=RS[j];
                RS[j]=RS[j+1];
                RS[j+1]=temp;
            }
        }
    }
}

main()
{
    int i,j,k,max,min,spos;
    printf("\nEnter No.of Requests: ");
    scanf("%d",&nor);
    printf("\nEnter Requests:\n");
    for(i=0;i<nor;i++)
    {
        printf("[%d]=",i);
        scanf("%d",&ReqString[i]);
    }
    printf("\nEnter No.of Cylinders: ");
    scanf("%d",&nob);
    printf("\nEnter Start Block: ");
```

```

scanf("%d",&start);
ReqString[nor++]=start;
sort(ReqString);
for(i=0;i<nor;i++)
    printf(" %d",ReqString[i]);
for(spos=0;spos<=nor && ReqString[spos]!=start; spos++);
printf("\nEnter Direction: ");
scanf(" %c",&direction);
min=getmin();
max=getmax();
if(direction=='L')
{
    printf("\n%d-%d",start,ReqString[0]);
    thm+=start-ReqString[0];
    printf("\n%d-%d",max,ReqString[spos+1]);
    thm+=max-ReqString[spos+1];
}
else if(direction=='R')
{
    printf("\n%d-%d",max,start);
    thm+=max-start;
    printf("\n%d-%d",ReqString[spos-1],min);
    thm+=ReqString[spos-1]-min;
}
printf("\nTotal Head Movement: %d",thm);
}
/*
[root@localhost disc sheduling]# cc clook.c
[root@localhost disc sheduling]# ./a.out

```

Enter No.of Requests: 8

Enter Requests:

```

[0]=95
[1]=180
[2]=34
[3]=119
[4]=11
[5]=123
[6]=62
[7]=64

```

Enter No.of Cylinders: 200

Enter Start Block: 50

11 34 50 62 64 95 119 123 180

Enter Direction: R

180-50

34-11

Total Head Movement: 153

*/