CSE 316

Operating System Assignment

Problem no: 11

Name : Priyansh Rawat

Student ID : 11803937

Section : K18SJ

Email : rawatpriyansh@gmail.com

GitHub link : https://github.com/priyansh-rawat/SCAN-Disk-Scheduling-in-c

Problem Statement Q11:

Write a C program to solve the following problem:

Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO

Order is:

86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the SCAN disk-scheduling algorithms?

Problem Analysis:

In this problem the SCAN disk-scheduling algorithm is implemented. In SCAN algorithm the disk arm covers all the cylinders in one direction (left or right depending on the current and previous requests) till the extreme end and then covers all the cylinders in the other direction.

This program can be solved in C language with the help of multiple arrays. The program inputs the current arm position and divides the other requests in 2 separate queues depending on their position with respect to the current position. They shall then be traversed first from either left to right (if current position is greater than previous position) or right to left (if current position is lower than previous position).

Algorithm:

1. input max, n, current, prev. (where max is maximum size of disk. n is number of pending requests. current is cylinder being currently served. prev is the cylinder last served.)
2. int I,j=0,k=0
3. Initialise arrays a, q1, q2
4. for i=0..n step by +1
    1. a[i]=read("Enter request);
    2. if(a[i]<current)
        1. q1[j]=a[i];
        2. j++;
    3. else
        1. q2[k]=a[i];
        2. k++;
5. if(current<prev)
    1.sort(q1,j,1);
    2. sort(q2,k,0);
    3. if(k==0)
        1. total= current;
    4. else
        1.total=current+q2[k-1];
    5. write(total);
6. else
    1.sort(q2,k,0);
    2. sort(q1,j,1);
    3. if(k==0)
        1. total= max - current;
    4. else
        1.total=max + max – current – q1[0];
    5. write(total);
7. free(a) , free(q1), free(q2);
8. end.

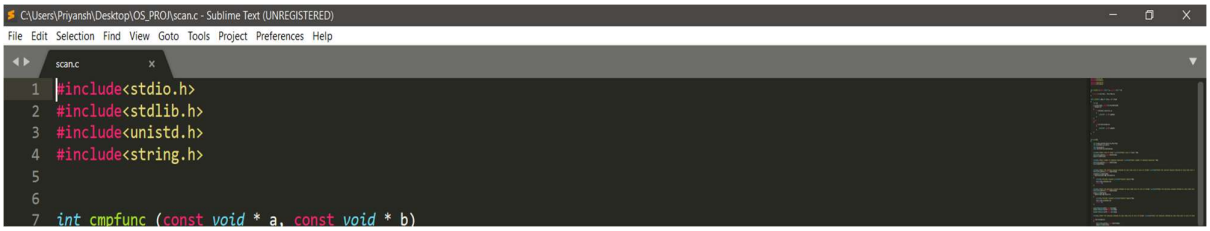
Complexity:

As there are 3 for loops (**not** nested) , each loop running n times, the program will complete its running with an efficient complexity of **O(n).**

The program does not require any storage of data and thus the space complexity of the program is **O(1).**

Code:

Header files –



Main function –

```
int main()
{
    int n,max,current,prev,*a,*q1,*q2;
    int i,total=0,j=0,k=0;
    int z1,z2,z3,f;
    char buffer[10],invalid[1];

    write(1,"Enter size of disk: ",strlen("Enter size of disk: "));
    z1=read(0,buffer,sizeof(buffer));
    max=atoi(buffer);

    write(1,"Enter number of pending requests: ",strlen("Enter number of pending requests: "));
    z1=read(0,buffer,sizeof(buffer));
    n=atoi(buffer);


    write(1,"Enter the current request (Should be less than size of size of disk): ",strlen("Enter the current request (Should b
    z1=read(0,buffer,sizeof(buffer));
    current=atoi(buffer);
    if(current>=max || current<=0)
    {
        write(1,"Invalid request",strlen("Invalid request"));
        z2=read(0,invalid,1);
        return 0;
    }

    write(1,"Enter the previous request (Should be less than size of size of disk): ",strlen("Enter the previous request (Should
    z1=read(0,buffer,sizeof(buffer));
    prev=atoi(buffer);
    if(prev>=max || prev<=0)
    {
```

```
    if(prev>=max || prev<=0)
    {
        write(1,"Invalid request",strlen("Invalid request"));
        z2=read(0,invalid,1);
        return 0;
    }

    a=(int*)malloc(n*(sizeof(int)));
    q1=(int*)malloc(n*(sizeof(int)));
    q2=(int*)malloc(n*(sizeof(int)));

    write(1,"Enter the requests (Should be less than size of size of disk): ",strlen("Enter the requests (Should be less than size of size of

    for(i=0;i<n;i++)
    {
        z1=read(0,buffer,sizeof(buffer));
        a[i]=atoi(buffer);
        if(a[i]>=max || a[i]<=0)
        {
            write(1,"Invalid request",strlen("Invalid request"));
            z2=read(0,invalid,1);
            return 0;
        }
        if(a[i]<current)
        {
            q1[j]=a[i];
            j++;
        }
        else
        {
            q2[k]=a[i];
            k++;
        }

    }

    if(current<prev)
```

```
    if(current<prev)
    {
        printf("%d",current);
        sort(q1,j,1);
        printf(" -> 0");
        sort(q2,k,0);
        if(k==0)
            total= current;
        else
            total=current+q2[k-1];
            printf("/nTotal distance moved by disk arm is %d",total);
    }
    else
    {
        printf("%d",current);
        sort(q2,k,0);
        printf("-> %d",max);
        sort(q1,j,1);
        if(j==0)
            total=max-current;
        else
            total=max-current+max-q1[0];
        printf("\nTotal distance moved by disk arm is %d",total);
    }


    free(a);
    free(q1);
    free(q2);
    return 0;
}
```

Sort and Qsort functions –

```
5
6
7  int cmpfunc (const void * a, const void * b)
8  {
9     return ( *(int*)a - *(int*)b );
10 }
11
12 void sort(int a[],int size, int flag)
13 {
14     int i;
15     qsort(a,size, sizeof(int),cmpfunc);
16     if(flag==1)
17     {
18         for(i=size-1;i>=0;i--)
19         {
20             printf(" -> %d",a[i]);
21         }
22     }
23     else
24     {
25         for(i=0;i<size;i++)
26         {
27             printf(" -> %d",a[i]);
28         }
29     }
30 }
31
```

Output

```
priyansh@LAPTOP-CJU4390G:/mnt/c/users/priyansh/desktop/OS_PROJ$ gcc scan.c
priyansh@LAPTOP-CJU4390G:/mnt/c/users/priyansh/desktop/OS_PROJ$ ./a.out
Enter size of disk: 5000
Enter number of pending requests: 9
Enter the current request (Should be less than size of size of disk): 143
Enter the previous request (Should be less than size of size of disk): 126
Enter the requests (Should be less than size of size of disk): 86
1470
913
1774
948
1509
1022
1750
130
The requests are served in the order:
 143 -> 913 -> 948 -> 1022 -> 1470 -> 1509 -> 1750 -> 1774-> 5000 -> 130 -> 86
Total distance moved by disk arm is 9771
priyansh@LAPTOP-CJU4390G:/mnt/c/users/priyansh/desktop/OS_PROJ$
```