Experiment 4 **Surya Narayanan**

Disk Scheduling Algorithms

s5 csb 48

---

1. Algorithm

Step 1. START
Step 2. Include necessary headers.
Step 3. Declare and initialize necessary variables
Step 4. Get user inputs for the total cylinders, current head position, no of request in queue and the requests.
Step 5. If input request value > cylinder numbers, display "invalid input".
Step 6. Repeat the following till user choice is to continue.
Step 6.1. Display the menu and get the user choice
Step 6.2 if choice = 1 (FCFS)
Step 6.2.1 – Display the requests in the same order as entered by users.
Step 6.2.2 – Each time the value is displayed subtract the current value from preceding value of request.
Step 6.2.3 – Display the result of head network.

Step 6.3 - if choice = 2 (SCAN)

Step 6.3.1 - Set count and position as 0. Arrange requests in order.

Step 6.3.2 - Compare the difference between the nearby process request using function absolute.

Step 6.3.3 Add count each time with difference attained from absolute function.

Step 6.3.4 - Display the result of head movements

Step 6.4 if choice

Step 6.4.1 - Set count and position as 0.

Step 6.4.2 Arrange the requests in order.

Step 6.4.3 - Get the difference of requests pair from function absolute

Step 6.4.4 - Add up count values each time with difference obtained.

Step 6.4.5 – Display the result

Step 6.5 – Get user choice for the prgm to continue.

Step 6.6 – If yes, go to step 6.
else go to step 7.

Step 7. STOP

Absolute ()

Step 1 : START

Step 2 : Calculate difference between each pair of requested cylinders.

Step 3 : Return the calculated difference

Step 4 : STOP.

Program

```
#include <stdio.h>
int absolute(int a, int b)
{
    int c;
    c = a - b;
    if (c < 0)
        return -c;
    else
        return c;
}
int main()
{
    printf(" DISK SCHEDULING ALGORITHM");
    int choice, m, n, x, start, i, j, pos, min, a[15], count;
    count = 0;
    printf("\nEnter the number of cylinders :");
    scanf("%d", &m);
    printf("\nEnter the number of requests :");
    scanf("%d", &n);
    printf("\nEnter current position of read-write head :");
    scanf("%d", &start);
    printf("\nEnter the request queue :");
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
        if (a[i] >= m)
        {
            printf("\n Input value has exceeded the cylinder
limit");
            scanf("%d", &a[i]);
        }
    }
    do
    {
        printf("\n\nDISK SCHEDULING
ALGORITHMS\n1.FirstComeFirstServescheduling\n2.SCAN(Elevator)
scheduling\n3.Circular-SCAN scheduling\n");
        printf("\nEnter scheduling choice :");
        scanf("%d", &choice);
```

```
        count = 0;
        x = start;
        switch (choice)
        {
        case 1:
            printf("\nFCFS :\n");
            printf("Scheduling order:\n%d\t", start);
            for (i = 0; i < n; i++)
            {
                x -= a[i];
                if (x < 0)
                    x = -x;
                count += x;
                x = a[i];
                printf("%d\t", x);
            }
            printf("\nTotal Head Movement :%d Cylinders", count);
            break;

        case 2:
            printf("\nSCAN :\n");
            printf("Scheduling order:\n");

            count = 0;
            pos = 0;
            for (i = 0; i < n; i++)
                for (j = 0; j < n - i - 1; j++)
                    if (a[j] > a[j + 1])
                    {
                        x = a[j];
                        a[j] = a[j + 1];
                        a[j + 1] = x;
                    }
            for (i = 0; i < n; i++)
                if (a[i] < start)
                    pos++;
            for (i = 0; i < pos; i++)
                for (j = 0; j < pos - i - 1; j++)
                    if (a[j] < a[j + 1])
                    {
```

```
                    x = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = x;
                }
        x = start;
        printf("%d\t", x);
        for (i = 0; i < pos; i++)
        {
            count += absolute(a[i], x);
            x = a[i];
            printf("%d\t", x);
        }
        count += absolute(x, 0);

        x = 0;
        printf("%d\t", x);
        for (i = pos; i < n; i++)
        {
            count += absolute(a[i], x);
            x = a[i];
            printf("%d\t", x);
        }

        printf("\nTotal Head Movement: %d Cylinders", count);
        break;
    case 3:
        printf("\nC-SCAN :\n");
        printf("Scheduling order:\n%d\t", start);
        count = 0;
        pos = 0;
        for (i = 0; i < n; i++)
            for (j = 0; j < n - i - 1; j++)
                if (a[j] > a[j + 1])
                {
                    x = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = x;
                }
        for (i = 0; i < n; i++)
            if (a[i] < start)
```

```
                    pos++;
            x = start;
            for (i = pos; i < n; i++)
            {
                count += absolute(x, a[i]);

                x = a[i];
                printf("%d\t", x);
            }
            count += absolute(m - 1, x);
            x = 0;
            printf("%d\t%d\t", m - 1, 0);
            for (i = 0; i < pos; i++)
            {
                count += absolute(x, a[i]);
                x = a[i];
                printf("%d\t", x);
            }

            printf("\nTotal Head movement: %d Cylinders", count);
            break;
        }
        printf("\nPress 1 to continue :");
        scanf("%d", &choice);
    } while (choice == 1);
}
```

Output


Ubuntu
/mnt/e/school/ss 06:21 PM > ./a.out
DISK SCHEDULING ALGORITHM
Enter the number of cylinders :200
Enter the number of requests :8
Enter current position of read-write head :50
Enter the request queue :95 180 34 119 11 123 62 64
DISK SCHEDULING ALGORITHMS
1.FirstComeFirstServescheduling
2.SCAN(Elevator) scheduling
3.Circular-SCAN scheduling
Enter scheduling choice :3
C-SCAN :
Scheduling order:
50 62 64 95 119 123 180 199 0 11 34
Total Head movement: 183 Cylinders
Press 1 to continue :1
DISK SCHEDULING ALGORITHMS
1.FirstComeFirstServescheduling
2.SCAN(Elevator) scheduling
3.Circular-SCAN scheduling
Enter scheduling choice :2
SCAN :
Scheduling order:
50 34 11 0 62 64 95 119 123 180
Total Head Movement: 230 Cylinders
Press 1 to continue :1
DISK SCHEDULING ALGORITHMS
1.FirstComeFirstServescheduling
2.SCAN(Elevator) scheduling
3.Circular-SCAN scheduling
Enter scheduling choice :1
FCFS :
Scheduling order:
50 34 11 62 64 95 119 123 180
Total Head Movement :208 Cylinders
Press 1 to continue :0
/mnt/e/school/ss 06:23 PM >
6:23 PM