



Green University of Bangladesh

Dept. of CSE

Course Title: Operating System Lab

Lab Report-04

| Submitted By: | | Submitted To: | |
|----------------------------------|--|---------------------------------------|--|
| Md. Tanvir Ahmed | | Most. Rokeya Khatun | |
| ID No : 191015102 | | Designation: Lecturer | |
| Batch No: ED | | Dept. of CSE | |
| Semester : 8th | | Green University Of Bangladesh | |
| Performance Date:2/7/2021 | | Submission Date: 8/7/2021 | |

Experiment No: 04

a) C program for FCFS disk scheduling algorithm.

Introduction:

The simplest form of disk scheduling is, of course, the first-come, first-served algorithm. This algorithm is intrinsically fair, but it generally does not provide the fastest service. Disk Scheduling is the process of deciding which of the cylinder request is in the ready queue is to be accessed next. The access time and the bandwidth can be improved by scheduling the servicing of disk I/O requests in good order.

Algorithm:

- First Come First Serve Scheduling algorithm The operations are performed in order requested.
- There is no reordering of work queue.
- Every request is serviced, so there is no starvation.
- The seek time is calculated.
- Display the seek time and terminate the program.

Code:

```
#include<stdio.h>

int main()
{
    int queue[20],n,head,i,j,k,seek=0,diff;
    float avg;
    printf("Enter the head position: \n");
    scanf("%d",&head);
    printf("Enter the size of Disk queue: \n");
    scanf("%d",&n);
```

```
printf("Enter the queue of disk positions to be read/write: \n");
for(i=1; i<=n; i++)
    scanf("%d",&queue[i]);
queue[0]=head;
for(j=0; j<=n-1; j++)
{
    diff=abs(queue[j+1]-queue[j]);
    seek+=diff;
}
printf("Total seek time is %d\n",seek);
avg=seek/(float)n;
printf("Average seek time is %f\n",avg);
return 0;
}
```

Output:

D:\codeblocks\code\Lab_04\bin\Debug\Lab_04.exe

```
Enter the head position:
50
Enter the size of Disk queue:
8
Enter the queue of disk positions to be read/write:
95
180
34
119
11
123
62
64
Total Head Movement is 644
Average seek time is 80.500000

Process returned 0 (0x0)   execution time : 28.033 s
Press any key to continue.
```

Discussion:

The access time has two major components, Seek time and Rotational Latency. Seek time is the time for disk arm to move the heads to the cylinder containing the desired sector. Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head.

B) C program for SSTF disk scheduling algorithm.

Introduction:

SSTF stands for Shortest Seek Time First. SSTF is very efficient in seek movements. It reduces the total seek time as compared to FCFS. In this algorithm direction of the head pointer matters a lot. If somehow, we encounter a tie between requests then the head will serve the request in its ongoing direction. This algorithm provides less average response time and waiting time.

Algorithm:

- Shortest Seek Time First Scheduling algorithm, this algorithm selects the request with the minimum seek time from the current head position.
- Since seek time increases with the number of cylinders traversed by the head, SSTF chooses the pending request closest to the current head position.
- The seek time is calculated.
- Display the seek time and terminate the program.

Code:

```
#include <stdio.h>

#include <math.h>

int main()
{
    int queue[100], queue2[100], q_size, head, seek=0, temp;
    float avg;
    printf("%s\n", "Enter head position: ");
    scanf("%d", &head);
```

```
printf("%s\n", "Enter the size of the disk queue");
scanf("%d", &q_size);
printf("%s\n", "Enter the all queue elements");
for(int i=0; i<q_size; i++)
{
    scanf("%d",&queue[i]);
}
for(int i=0; i<q_size; i++)
{
    queue2[i] = abs(head-queue[i]);
}
for(int i=0; i<q_size; i++)
{
    for(int j=i+1; j<q_size; j++)
    {
        if(queue2[i]>queue2[j])
        {
            temp = queue2[i];
            queue2[i]=queue2[j];
            queue2[j]=temp;
            temp=queue[i];
            queue[i]=queue[j];
            queue[j]=temp;
        }
    }
}
```

```

        }
    }
}
for(int i=1; i<q_size; i++)
{
    seek = seek+abs(head-queue[i]);
    head = queue[i];
}
printf("\nTotal Head Movement is %d\t",seek);
avg = seek/(float)q_size;
printf("\nAverage seek time is %f\t", avg);
return 0;
}

```

Output:

```
D:\codeblocks\code\Lab-04_SSTF\bin\Debug\Lab-04_SSTF.exe
Enter head position:
50
Enter the size of the disk queue
8
Enter the all queue elements
95
180
34
119
11
123
62
64

Total Head Movement is 208
Average seek time is 26.000000
Process returned 0 (0x0)   execution time : 29.144 s
Press any key to continue.
```

Discussion:

- Disk scheduling algorithms are used to schedule multiple requests for accessing the disk.
- This algorithm services that request next which requires least number of head movements from its current position regardless of the direction.
- It breaks the tie in the direction of head movement.

Thank You