

Disk Scheduling

FCFS algorithm

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
void main()
{
    int i,n,m,sum=0,h;
    printf("Enter the size of disk\n");
    scanf("%d",&m);
    printf("Enter number of requests\n");
    scanf("%d",&n);
    printf("Enter the requests\n");
    // creating an array of size n
    int a[20];
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<n;i++)
    {
        if(a[i]>m)
        {
            printf("Error, Unknown position %d\n",a[i]);
        }
    }
    printf("Enter the head position\n");
    scanf("%d",&h);
    // head will be at h at the starting
    int temp=h;
    printf("%d",temp);
    for(i=0;i<n;i++){
        printf("-> %d ",a[i]);
        //cout<<" -> "<<a[i]<<' ';
        // calculating the difference for the head movement
        sum+=abs(a[i]-temp);
        // head is now at the next I/O request
        temp=a[i];
    }
    printf("\n");
```

```
printf("Total head movements = %d\n",sum);
getch();
}
```

output:

```
Enter the size of disk
199
Enter number of requests
8
Enter the requests
98 183 37 122 14 124 65 67
Enter the head position
53
53-> 98 -> 183 -> 37 -> 122 -> 14 -> 124 -> 65 -> 67
Total head movements = 640
```

Disk scheduling

SCAN algorithm

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
int main()
{
    int queue[20],n,head,i,j,seek=0,max,diff,temp,queue1[20],queue2[20],
        temp1=0,temp2=0;
    float avg;
    printf("Enter the max range of disk\n");
    scanf("%d",&max);
    printf("Enter the initial head position\n");
    scanf("%d",&head);
    printf("Enter the size of queue request\n");
    scanf("%d",&n);
    printf("Enter the queue of disk positions to be read\n");
    for(i=1;i<=n;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;
        }
    }
}
}
}
for(i=1,j=0;j<temp1;i++,j++)
queue[i]=queue1[j];
queue[i]=max;
for(i=temp1+2,j=0;j<temp2;i++,j++)
queue[i]=queue2[j];
queue[i]=0;
queue[0]=head;

```

```

    for(j=0;j<=n+1;j++)
    {
        diff=abs(queue[j+1]-queue[j]);
        seek+=diff;
        printf("Disk head moves from %d to %d with seek
%d\n",queue[j],queue[j+1],diff);
    }
    printf("Total seek time is %d\n",seek);
    avg=seek/(float)n;
    printf("Average seek time is %f\n",avg);
    getch();
    return 0;
}

```

Output:

```

Enter the queue of disk positions to be read
90 120 35 122 38 128 65 68
Disk head moves from 50 to 65 with seek      15
Disk head moves from 65 to 68 with seek      3
Disk head moves from 68 to 90 with seek      22
Disk head moves from 90 to 120 with seek     30
Disk head moves from 120 to 122 with seek     2
Disk head moves from 122 to 128 with seek     6
Disk head moves from 128 to 200 with seek     72
Disk head moves from 200 to 38 with seek     162
Disk head moves from 38 to 35 with seek       3
Disk head moves from 35 to 0 with seek       35
Total seek time is 350
Average seek time is 43.750000

```

Disk Scheduling

C-SCAN

```

#include<stdio.h>
#include<math.h>
#include<conio.h>
int main()
{
    int queue[20],n,head,i,j,seek=0,max,diff,temp,queue1[20],queue2[20],
        temp1=0,temp2=0;
    float avg;
    printf("Enter the max range of disk\n");
}

```

```

scanf("%d",&max);
printf("Enter the initial head position\n");
scanf("%d",&head);
printf("Enter the size of queue request\n");
scanf("%d",&n);

printf("Enter the queue of disk positions to be read\n");
for(i=1;i<=n;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;
    }
}
for(i=1,j=0;j<temp1;i++,j++)
queue[i]=queue1[j];
queue[i]=max;
queue[i+1]=0;
for(i=temp1+3,j=0;j<temp2;i++,j++)
queue[i]=queue2[j];
queue[0]=head;
for(j=0;j<=n+1;j++)
{
    diff=abs(queue[j+1]-queue[j]);
    seek+=diff;
    printf("Disk head moves from %d to %d with seek %d\n",queue[j],queue[j+1],diff);
}
printf("Total seek time is %d\n",seek);
avg=seek/(float)n;
printf("Average seek time is %f\n",avg);
getch();
return 0;
}

```

Output:

```

Enter the max range of disk
200
Enter the initial head position
50
Enter the size of queue request
8
Enter the queue of disk positions to be read
90 120 35 122 38 128 65 68
Disk head moves from 50 to 65 with seek 15
Disk head moves from 65 to 68 with seek 3
Disk head moves from 68 to 90 with seek 22
Disk head moves from 90 to 120 with seek 30
Disk head moves from 120 to 122 with seek 2
Disk head moves from 122 to 128 with seek 6
Disk head moves from 128 to 200 with seek 72
Disk head moves from 200 to 0 with seek 200
Disk head moves from 0 to 35 with seek 35
Disk head moves from 35 to 38 with seek 3
Total seek time is 388
Average seek time is 48.500000

```