

实验报告：磁盘调度模拟实验

姓名：陈晓彤 学号：PB14000556 时间：2016.5.21

实验要求：用 C 语言设计模拟一个磁盘调度算法，并测试其正确性。

输入：磁头所在的位置及对块的调度序列

输出：对块的调度访问序列

实验背景：基本的磁盘调度算法有 先来先服务算法 最短寻道时间优先算法 扫描算法 循环扫描算法等，本实验选择实现最短寻道时间优先算法，即每次寻找与当前磁头位置距离最近的磁道

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

```
#include<time.h>
```

```
#include<stdlib.h>
```

```
int main()
```

```
{
```

```
    srand((unsigned)time(NULL));
```

```
    int pos[20],temp[20];
```

```
    int i,j;
```

```
    for(i=0;i<20;i++)
```

```
    {
```

```
        pos[i]=rand()%128;//产生随机的磁道序列
```

```
    }
```

```
    printf("磁道序列 : \n");
```

```
    for(i=0;i<20;i++)
```

```
    {
```

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

```
    }
```

```
    printf("\n 输入磁头位置 : \n");
```

```
    int init,a;
```

```
    scanf("%d",&init);
```

```
    for(i=0;i<20;i++)
```

```
    {
```

```
        temp[i]=(pos[i]>init)?(pos[i]-init):(init-pos[i]);//计算磁头与每个磁道的距离
```

```
    }
```

```
    for(i=0;i<20;i++)
```

```
    {
```

```
        for(j=0;j<20-i-1;j++)
```

```
        {
```

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

```
            {
```

```
                a=pos[j];pos[j]=pos[j+1];pos[j+1]=a;
```

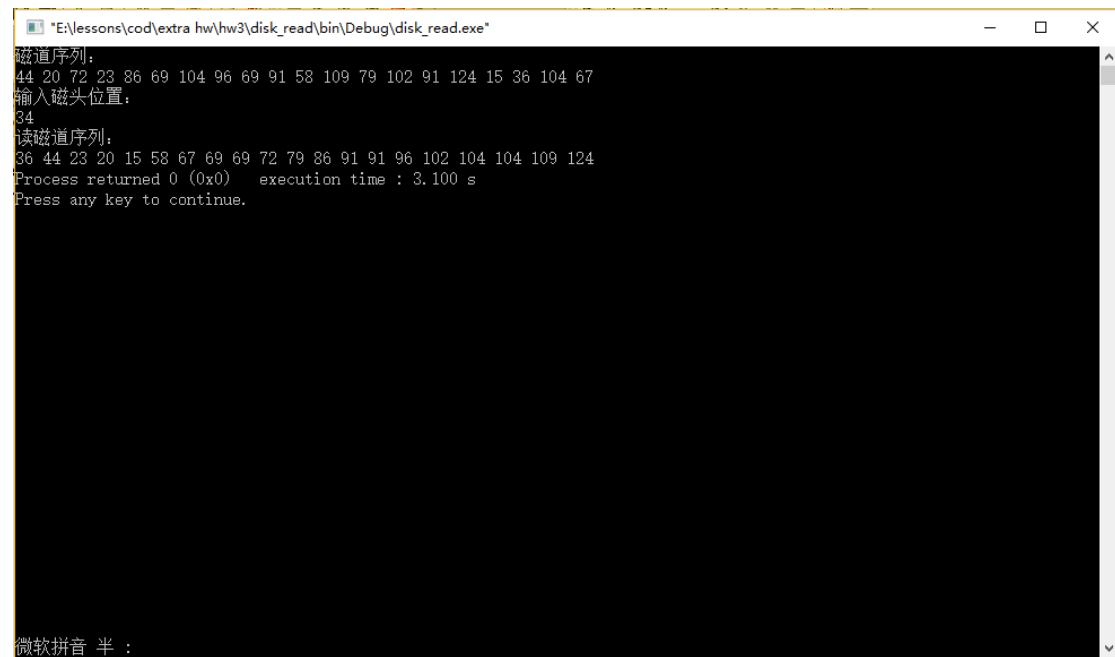
```
                a=temp[j];temp[j]=temp[j+1];temp[j+1]=a;//根据距离排序得到访问序列
```

```
            }
```

```
        }
```

```
    }
```

```
printf("读磁道序列 : \n");  
for(i=0;i<20;i++)  
{  
    printf("%d ",pos[i]);  
}  
return 0;  
}
```



```
E:\lessons\cod\extra hw\hw3\disk_read\bin\Debug\disk_read.exe  
磁道序列:  
44 20 72 23 86 69 104 96 69 91 58 109 79 102 91 124 15 36 104 67  
输入磁头位置:  
34  
读磁道序列:  
36 44 23 20 15 58 67 69 69 72 79 86 91 91 96 102 104 104 109 124  
Process returned 0 (0x0)   execution time : 3.100 s  
Press any key to continue.  
微软拼音 半 :
```

可以看到访问序列正确