

# P6 Disk

《计算机组织结构》课程助教组



南京大學  
NANJING UNIVERSITY

# 作业要求

- 实现6种磁盘调度算法，返回磁头移动的平均距离

```
public double FCFS(int start, int[] request)
```

```
public double SSTF(int start, int[] request)
```

```
public double SCAN(int start, int[] request, boolean direction)
```

```
public double CSCAN(int start, int[] request)
```

```
public double LOOK(int start, int[] request, boolean direction)
```

```
public double CLOOK(int start, int[] request)
```

- 实现磁头寻道的相关方法

```
public void seek(int start)
```

```
public void addPoint()
```



# 磁头寻道

start表示距离磁盘开始的字节数，通过设置track, sector, point实现寻道

```
point = start % BYTE_PER_SECTOR;  
sector = (start / BYTE_PER_SECTOR) % SECTOR_PER_TRACK;  
track = ((start / BYTE_PER_SECTOR) / SECTOR_PER_TRACK) % TRACK_NUM;
```

addPoint表示磁头移动一个字节，如果超过sector/track边界就移动到下一个sector/track，如果超过了最后一个track的最后一个sector，就返回0号磁道。



# 磁盘调度-FCFS/SSTF

先来先服务算法（FCFS）按照请求顺序依次计算所有位置和上一个位置的绝对值的和，每次调度后更新位置。最后除以调度次数。

最短寻道时间优先算法（SSTF）每次计算离当前磁道最近的未执行的请求，然后移动到该请求的位置，将移动的距离相加。使用一个visited布尔类型数组记录访问过的请求。



# 磁盘调度-SCAN/CSCAN

SCAN算法需要先将所有的请求排序，然后根据磁头移动的方向判断。

- 磁头向磁道号增大的方向移动：
  - 如果所有请求磁道号都比当前位置大，距离和 = 最大**请求**磁道号-start
  - 其他情况，距离和 = 最大磁道号-start+最大磁道号-最小**请求**磁道号
- 磁头向磁道号减小的方向移动：（同理）
  - 如果所有请求磁道号都比当前位置小，距离和 = start-最小**请求**磁道号
  - 其他情况，距离和 = start-0+最大**请求**磁道号-0

CSCAN算法（此处默认只向磁道号增大的方向移动相应请求）

先遍历一遍请求找到比start小的最大值和最大请求磁道号，然后计算距离和。

- 如果不存在比当前位置小的最大值，距离和 = 最大**请求**磁道号-start
- 存在，距离和 = 最大磁道号-start+最大磁道号-0+比start小的最大值磁道号



# 磁盘调度-LOOK

LOOK算法需要先将所有的请求排序，然后分三种情况计算距离和：

- 所有请求磁道号  $> \text{start}$ ：距离和 = 最大请求磁道号 - start
- 所有请求磁道号  $< \text{start}$ ：距离和 = start - 最小请求磁道号
- start 介于二者之间：设  $\text{len} = \text{最大请求磁道号} - \text{最小请求磁道号}$ 
  - 磁头向磁道号增大方向移动：距离和 = 最大请求磁道号 - start + len
  - 磁头向磁道号减小方向移动：距离和 = start - 最小请求磁道号 + len



# 磁盘调度-CLOOK

CLOOK算法（此处同样默认磁头向磁道号增大方向移动）

先遍历一遍请求找到比start小的最大值和最大请求磁道号，然后计算距离和。

设  $len = \text{最大请求磁道号} - \text{最小请求磁道号}$

- 所有请求磁道号  $> start$ : 距离和 = 最大请求磁道号 - start
- 所有请求磁道号  $< start$ : 距离和 = start - 最小请求磁道号 + len
- 介于两者之间:

距离和 = 最大请求磁道号 - start + len + 比start小的最大值 - 最小请求磁道号

