

# 操作系统原理

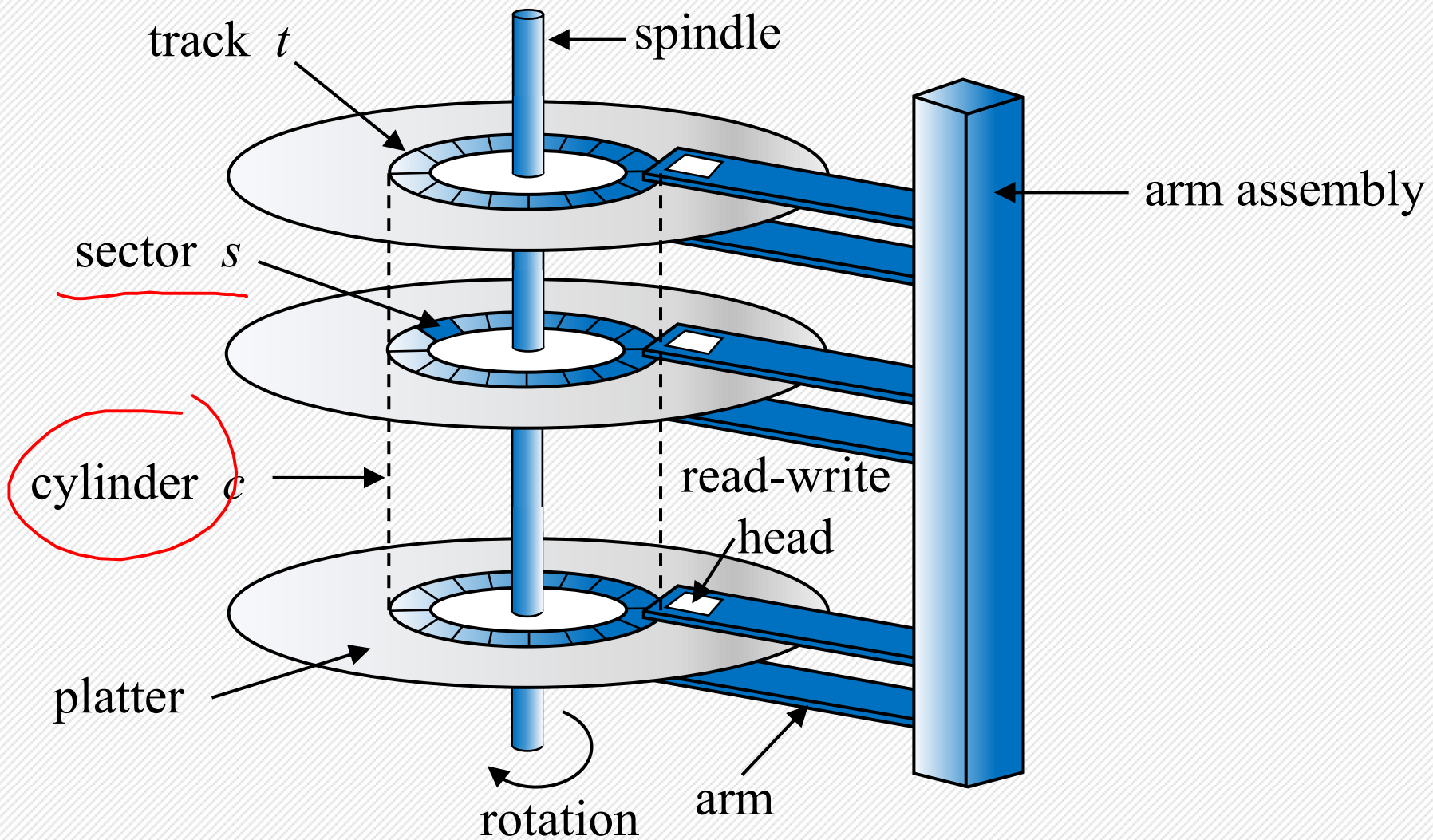
Operating System Principle

田丽华

# 11-1 磁盘结构

# Moving-Head Disk Mechanism

## 磁头移动机制



## 磁盘结构

### 磁盘组



由若干磁盘组成，绕枢轴高速旋转，目前主流磁盘转速7200转/分

### 物理记录定位



柱面号：各盘面所有的读写头同时移动，并定位在同样的垂直位置的磁道上，这些磁道形成了一个柱面。由外向内是：  
 $0, 1, 2, \dots, L$ 。

磁头号：磁盘的全部有效盘面从上到下依次编号：  
 $0, 1, 2, \dots, H$ 。磁头号与盘面号是相对应的。

扇区号：将各盘面分割成若干大小相等的扇区，编号：  
 $0, 1, 2, 3, \dots, n$

## 磁盘结构



Disk drives are addressed as large 1-dimensional arrays of *logical blocks*, where the logical block is the smallest unit of transfer. 磁盘设备是以一种逻辑块的一维大数组的形式编址的，这里的逻辑块是传输的最小单位。

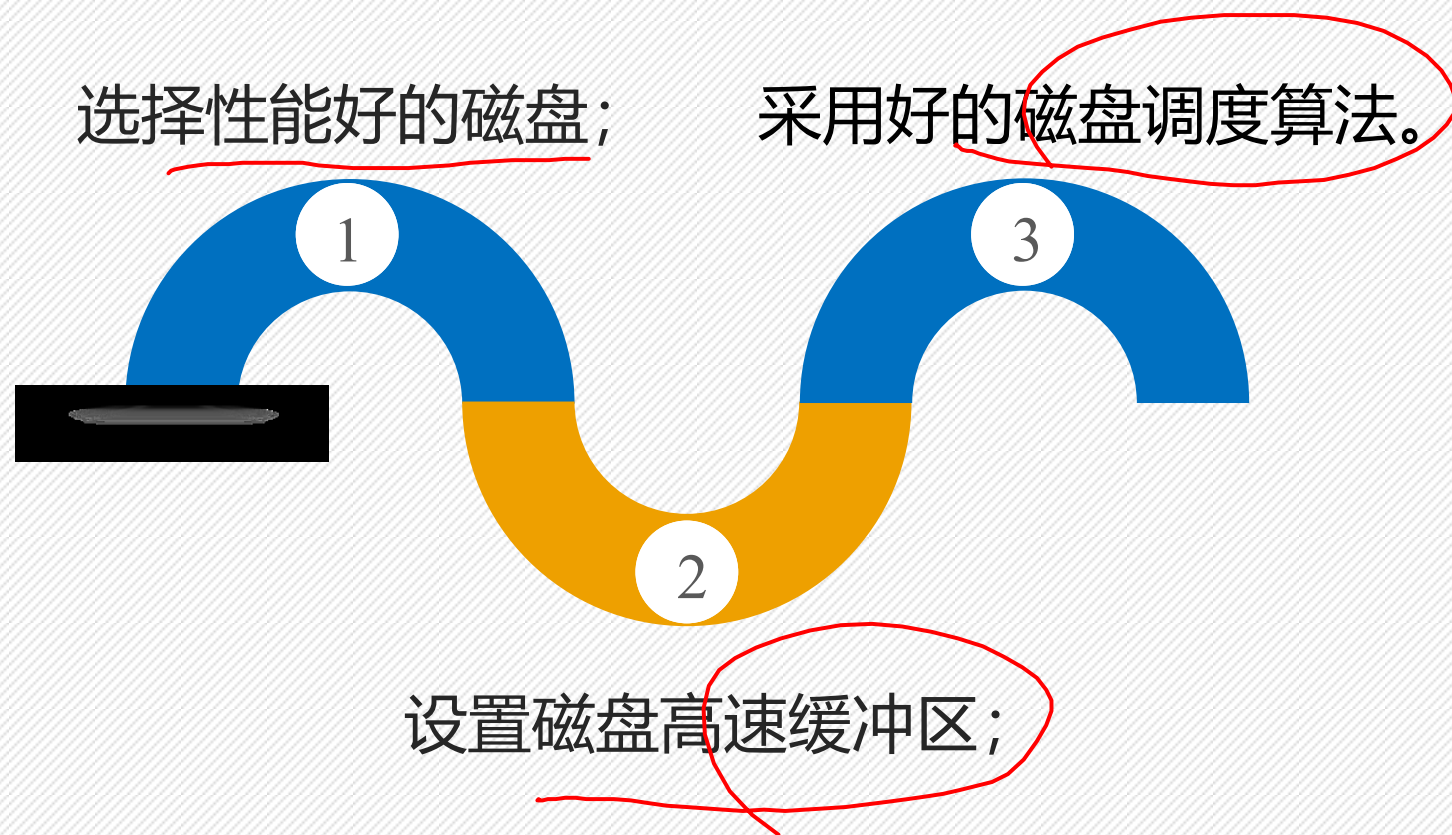


The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially. 逻辑块的一维数组映射到磁盘上一些相连的扇区。

- Sector 0 is the first sector of the first track on the outermost cylinder. 0扇区是最外边柱面的第一个磁道的第一个扇区。
- Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost. 数据首先都映射到一个磁道，其余的数据映射到同一柱面的其他磁道，然后按照从外向里的顺序映射到其余的柱面。

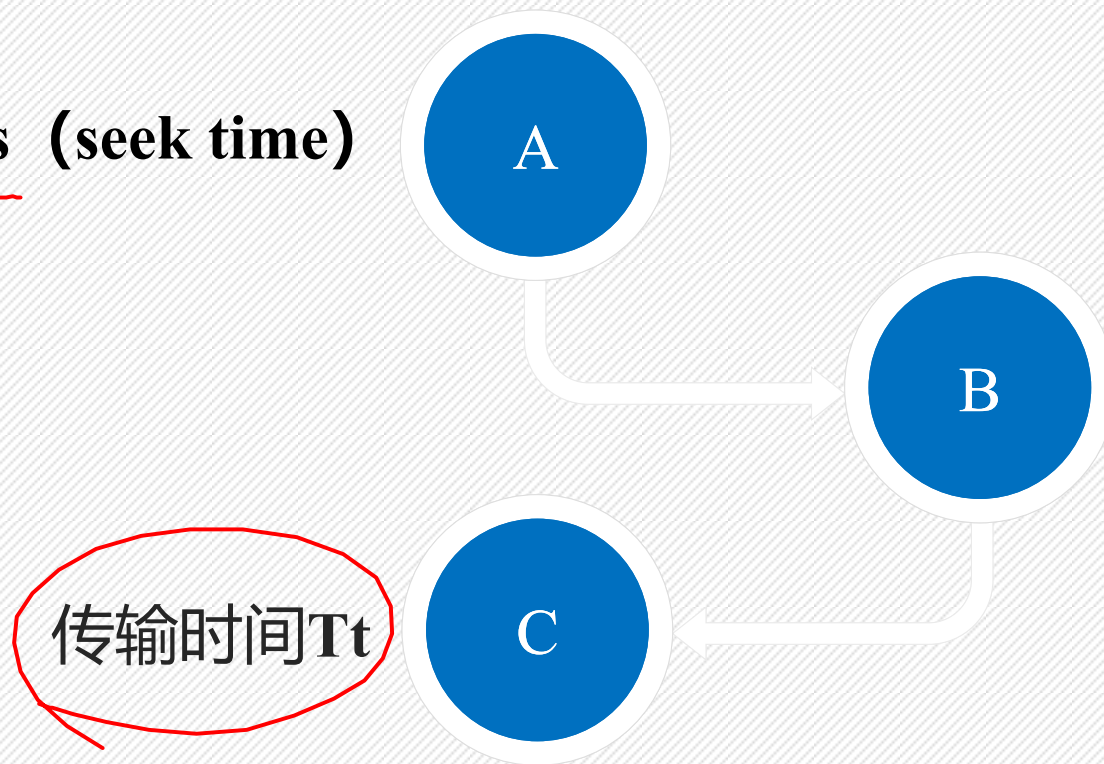
# 如何提高磁盘I/O速度

- ◆ 磁盘I/O速度的高低，将直接影响文件系统的性能。
- ◆ 提高磁盘I/O速度的主要途径有：



# 磁盘访问时间

寻道时间 $T_s$  (seek time)



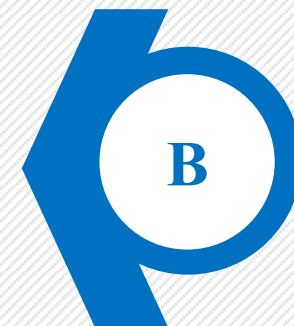
旋转延迟时间 $T_r$   
(rotational latency)

# 寻道时间 $T_s$ (seek time)



把磁头从当前位置移动到指定磁道上所经历的时间，它与磁盘转速、移动的磁道数有关。

该时间是启动磁盘的时间 $s$ 与磁头移动 $n$ 条磁道所花费的时间之和，即  $T_s = m \times n + s$



式中， $m$ 是一常数，它与磁盘驱动器的速度有关。

- 对一般磁盘， $m = 0.2$ ；
- 对高速磁盘， $m \leq 0.1$ ，磁臂启动时间约为2ms。

一般，寻道时间将随寻道距离的增大而增大





# 旋转延迟时间Tr (rotational latency)



$$\underline{Tr = 1/2r}$$



$T_t$ 是指把数据从磁盘读出，或向磁盘写入数据所经历的时间。



- $b$ 为所读 / 写的字节数;
- $r$ 为磁盘以秒计的旋转速度;
- $N$ 为一条磁道上的字节数。

# 磁盘访问时间

访问时间可表示为

$$T_s = m \times n + s$$

$T_a$

- 寻道时间和旋转延迟时间，基本都与所读 / 写数据的多少无关，而且它通常是占据了访问时间的大头。
- 随着磁盘传输速率的不断提高，数据传输时间所占的比例更低。可见，适当地集中数据(不要太零散)传输，将有利于提高传输效率。

# 操作系统原理

Operating System Principle

田丽华

# 11-2 磁盘调度

# Disk Scheduling

## 磁盘调度



Minimize seek time 最小寻道时间

Seek time  $\approx$  seek distance

寻道时间  $\approx$  寻道距离



Disk Scheduling  
磁盘调度

### 磁盘调度 (续)

A

Several algorithms exist to schedule the servicing of disk I/O requests.  
有几种磁盘I/O请求的服务调度算法

B

We illustrate them with a request queue (0-199).  
我们假设一个请求序列。

98, 183, 37, 122, 14, 124, 65, 67

Head pointer 53 磁头当前的位置在53。

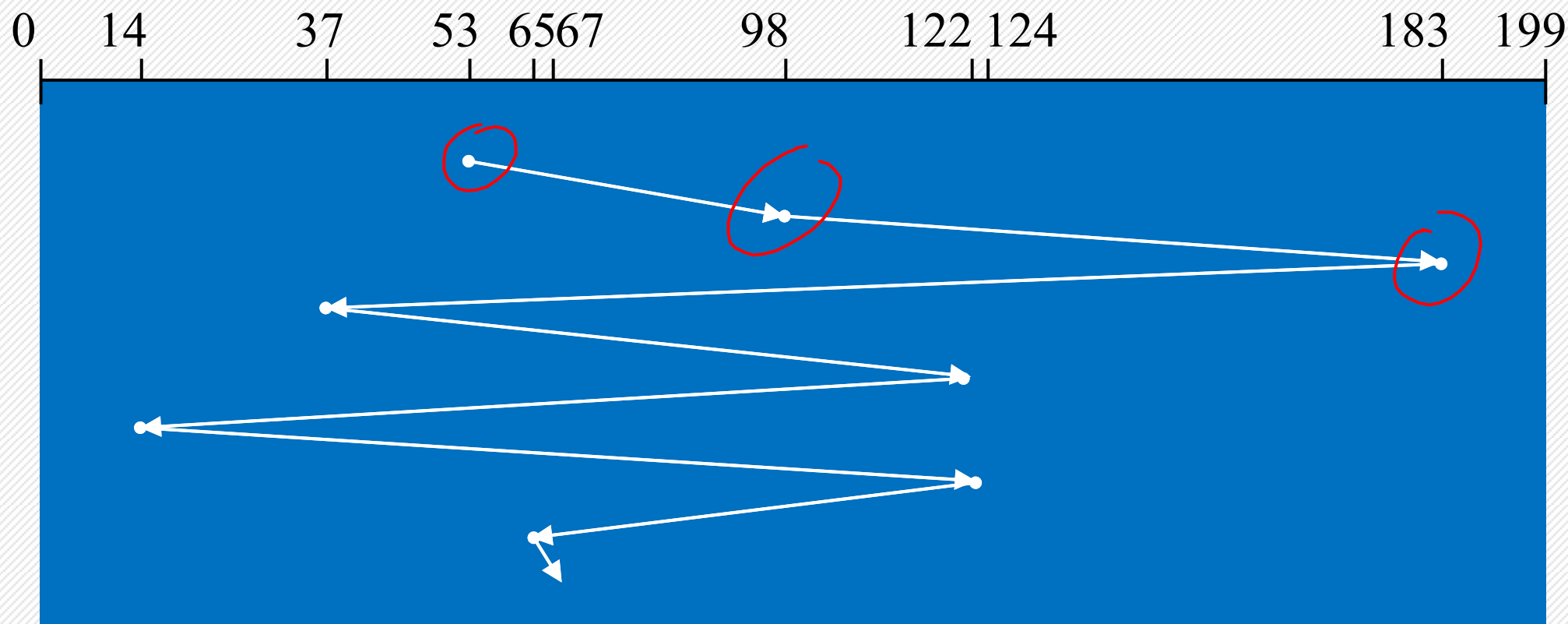
## 先来先服务

Illustration shows total head movement of 640 cylinders.

如下图所示，磁头总共移动了640个柱面的距离。

queue=98,183,37,122,14,124,65,67

head starts at 53

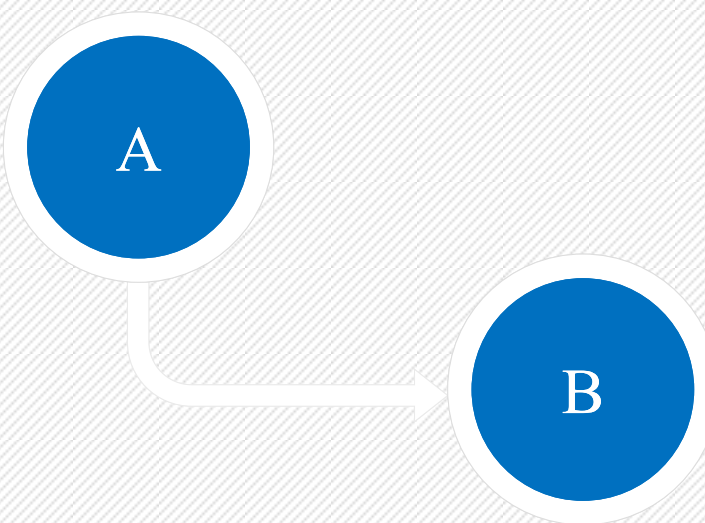




## 最短寻道时间优先

Selects the request with the minimum seek time from the current head position.

选择从当前磁头位置所需寻道时间最短的请求。

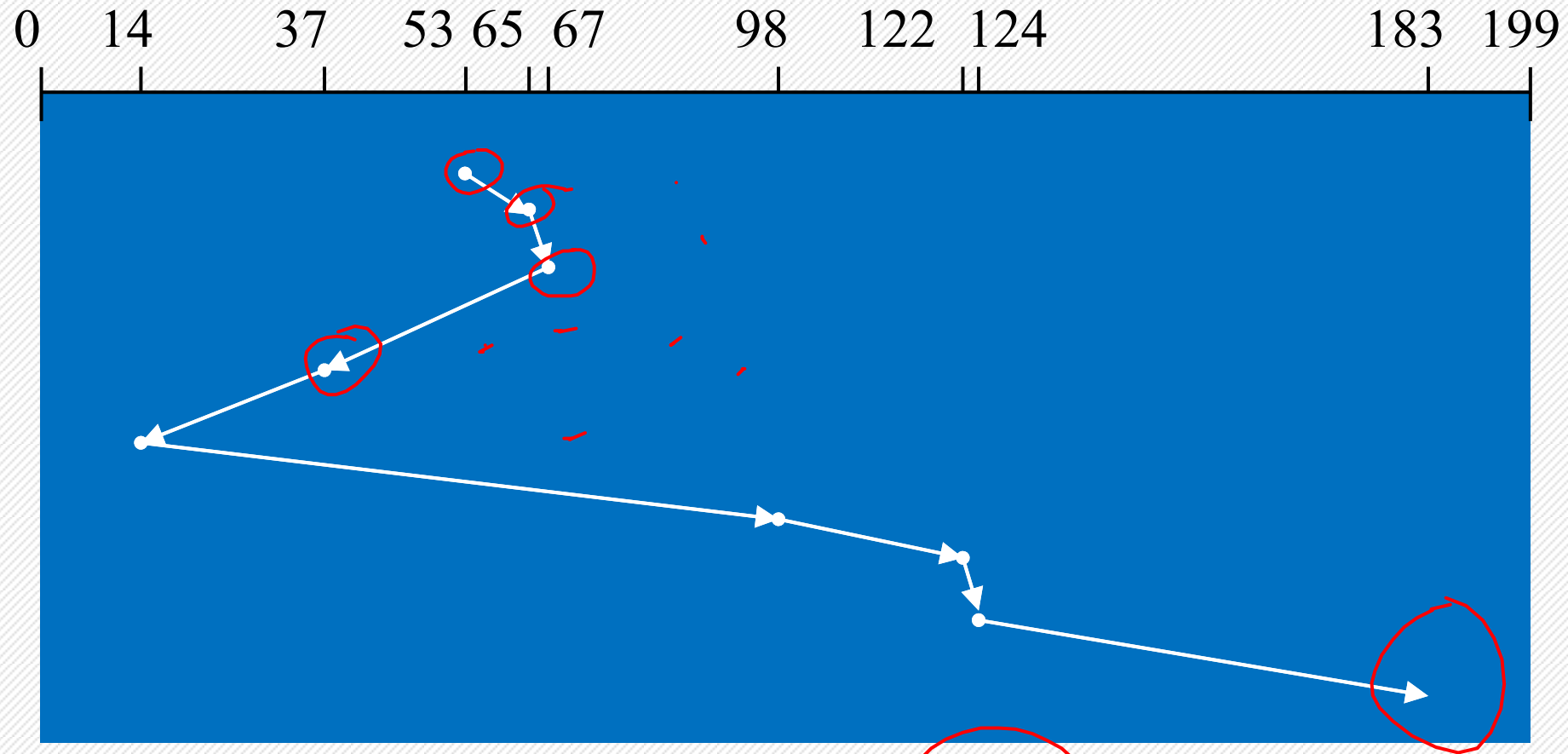


SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests. SSTF是SJF调度的一种形式；有可能引起某些请求的饥饿。

## SSTF (Cont.)

queue=98,183,37,122,14,124,65,67

head starts at 53



磁头移动的总距离是208柱面。

# 进程“饥饿”现象

现象

对于SSTF算法

只要不断有新进程到达，且新进程所要访问的磁道与磁头当前所在磁道的距离较近，则新进程一定优先满足，而某些老进程则会发生“饥饿”现象。

如何防止

如何防止“饥饿”现象

SCAN算法

不仅考虑到欲访问的磁道与磁头当前位置间的距离，更优先考虑的是磁头当前的移动方向。算法所选择的访问对象应当是与磁头当前的移动方向一致且距离最近的。

# SCAN

## 扫描算法

A

The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues. 磁头从磁盘的一端开始向另一端移动, 沿途响应访问请求, 直到到达了磁盘的另一端, 此时磁头反向移动并继续响应服务请求。

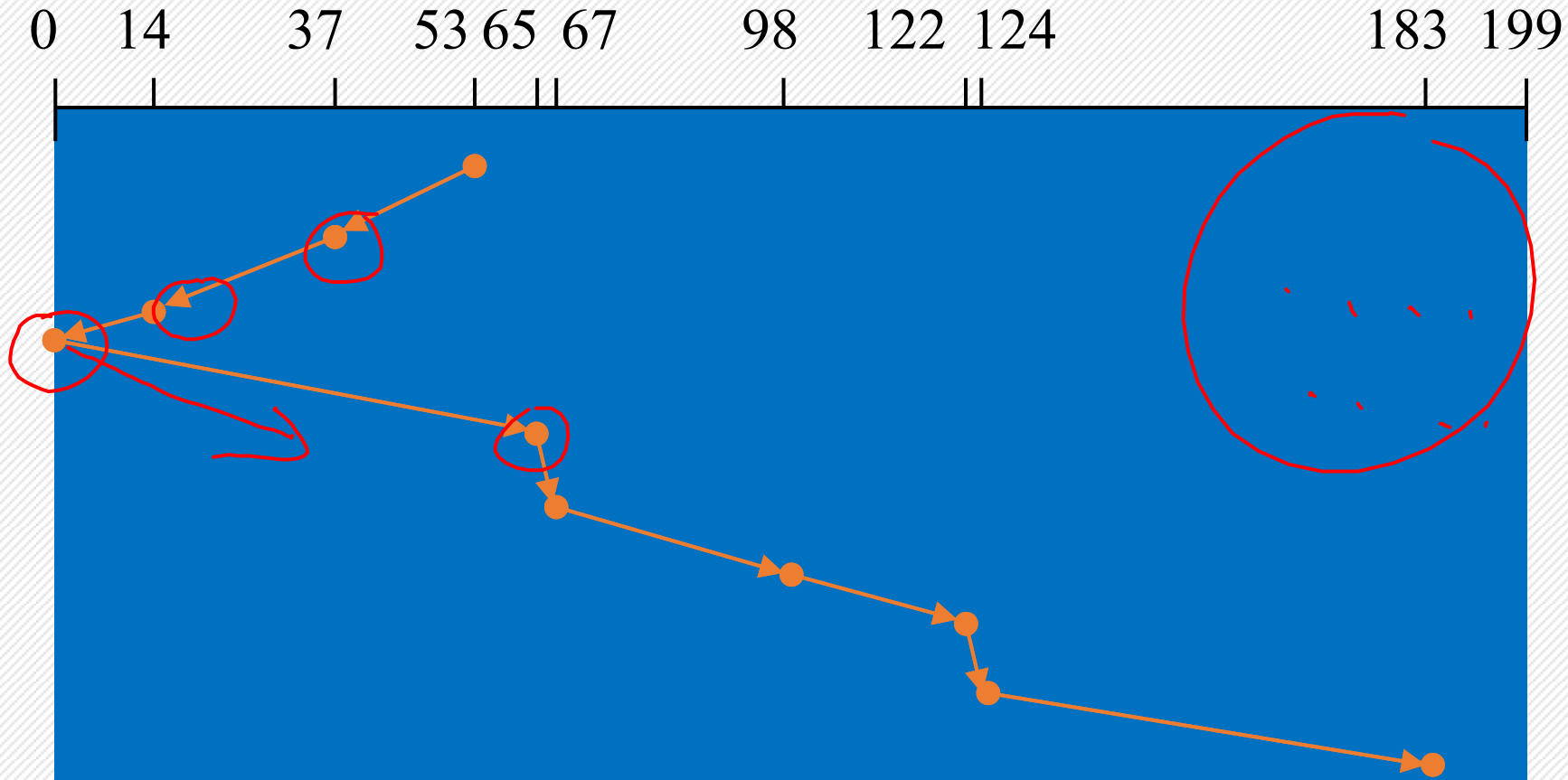
B

Sometimes called the elevator algorithm. 有时也称为电梯算法。

## SCAN (Cont.)

queue=98,183,37,122,14,124,65,67

head starts at 53, 朝磁道号减小的方向移动



磁头移动的总距离是236柱面



**Provides a more uniform wait time than SCAN.**

**提供比扫描算法更均衡的等待时间。**



**The head moves from one end of the disk to the other, servicing requests as it goes. When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip.** 磁头从磁盘的一段向另一端移动，沿途响应请求。当它到了另一端，就立即回到磁盘的开始处，**在返回的途中不响应任何请求。**



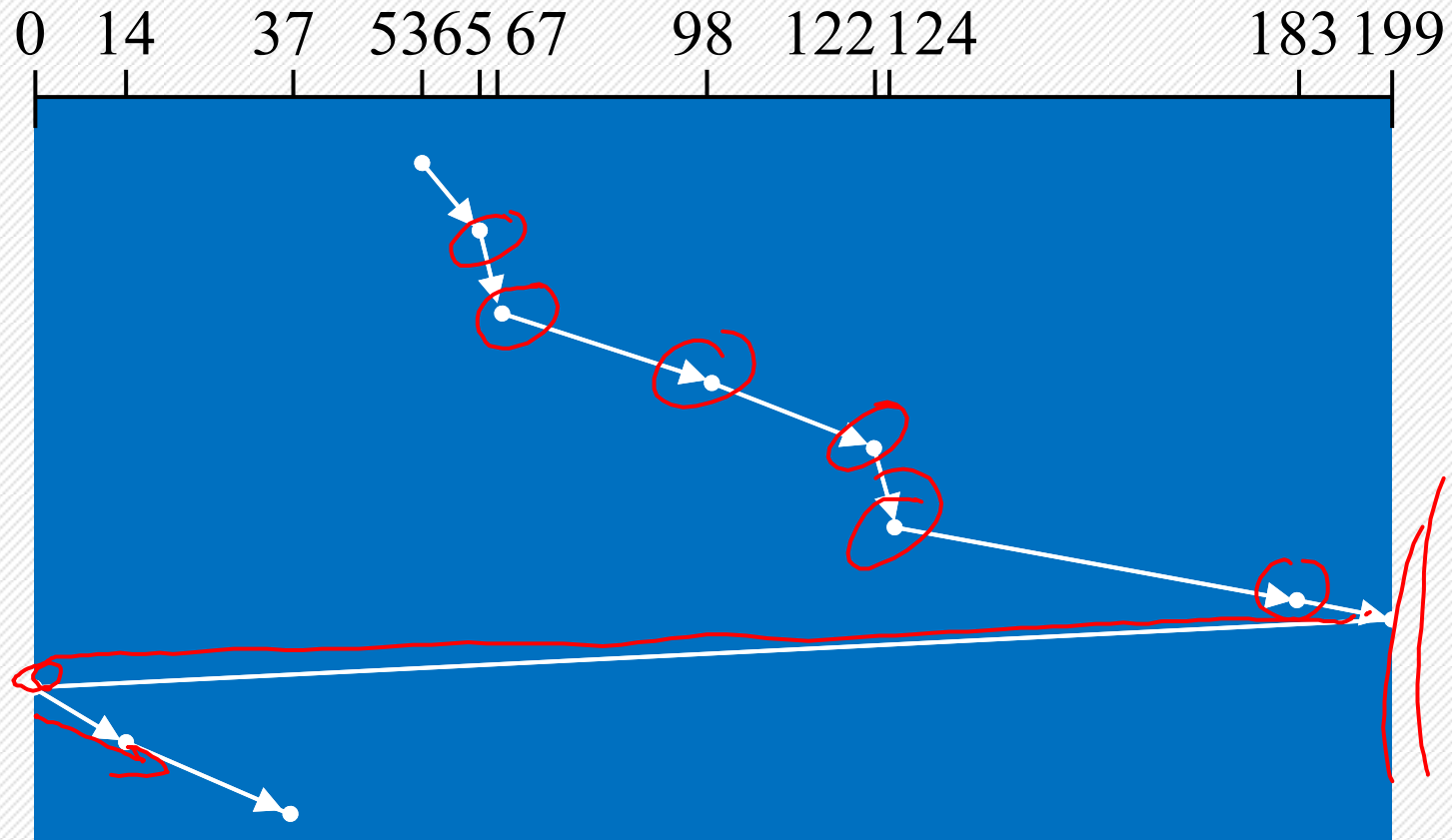
**Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.**

**把所有柱面看成一个循环的序列，最后一个柱面接续第一个柱面。**

## C-SCAN (Cont.)

queue=98,183,37,122,14,124,65,67

head starts at 53, 朝磁道号增加的方向移动

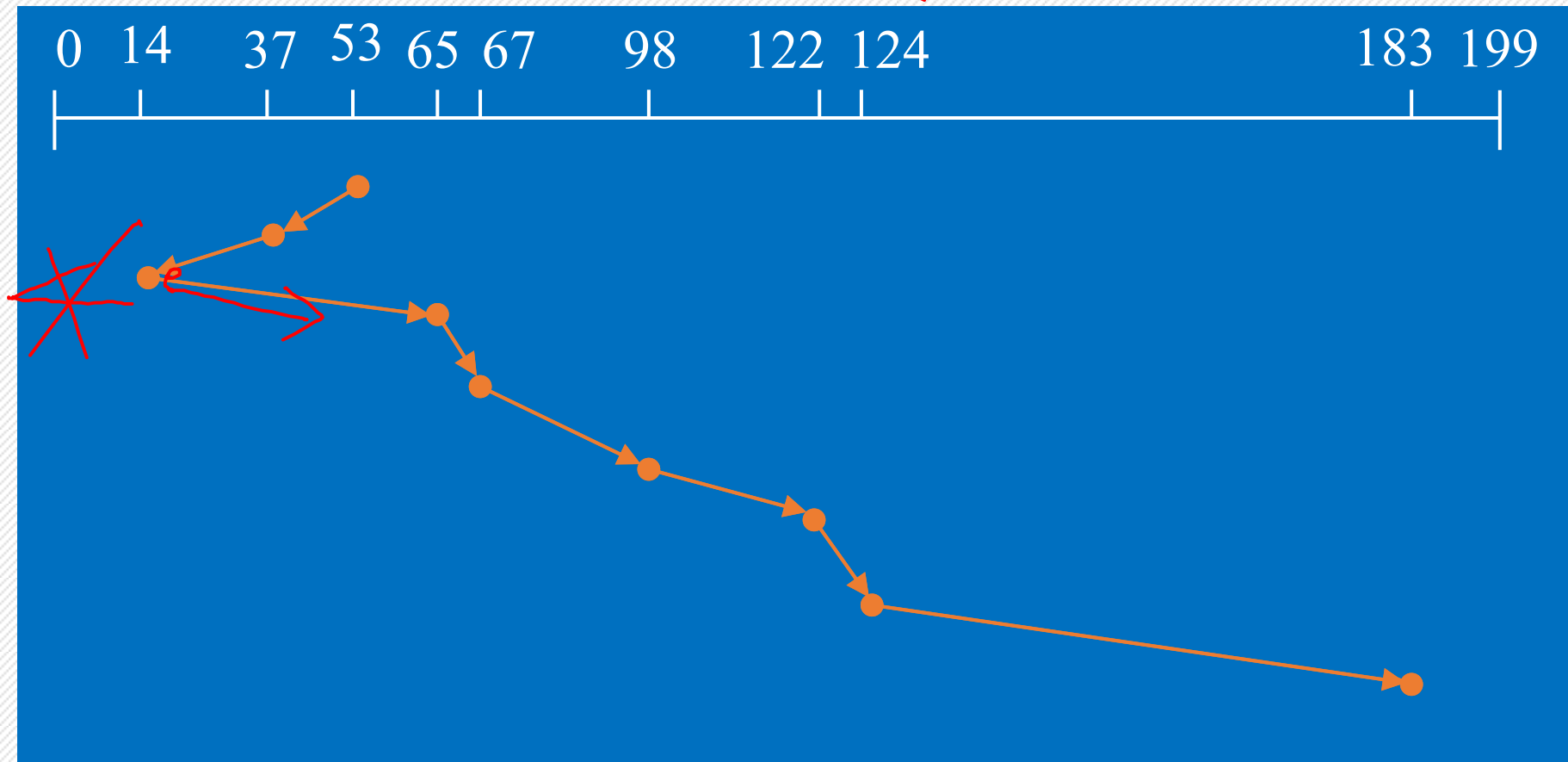


- **Version of SCAN** SCAN算法的一种形式。
- **Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.** 磁臂在每个方向上仅仅移动到最远的请求位置，然后立即反向移动，而不需要移动到磁盘的一端



# LOOK

- Queue=98,183,37,122,14,124,65,67
- head starts at 53, 朝磁道号减小的方向移动
- total head movement of 198 cylinders.



A **Version of C-SCAN** C-SCAN算法的一种形式。

B **Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk.**  
磁臂在每个方向上仅仅移动到最远的请求位置，然后立即反向移动，而不需要移动到磁盘的一端。

## C-LOOK (Cont.)

queue=98,183,37,122,14,124,65,67

head starts at 53, 朝磁道号增加的方向移动

