第十一章 I/O管理和磁盘调度

1. I/O的**关键**

   - 对于输入/输出最关键的问题是**性能**。
   - 考虑计算机的内部操作，可以看到处理器的速率在不断地提高，如果一个处理器不够快，则可以配置多个处理器来加快处理速率；内存的访问速率尽管没有处理器的速率提高得快，但是它也在不断提高。而且**通过使用一级、两级甚至更多级的高速缓存，主存访问时间应该可以跟上处理器的速率。**
   - I/O，特别是**磁盘存储**仍然面临着较大的挑战。

2. I/O的重要性体现

   - CPU 性能越高，输入输出设备性能同 CPU 性能不匹配的反差也越大。如何**解决这一矛盾，而又尽量不降低处理机的性能。**
   - I/O 设备千变万化，如何对它们实现**统一的管理**，从而方便用户使用。
   - I/O 设备能否及时将各种信息传送给计算机系统，计算机发出的各种命令能否通过 I/O 设备及时传送给执行部件。
   - 由设备传送的数据应该是安全和保密的，数据不能被破坏或被泄露。多用户多任务环境中的设备使用应该通过协调，避免冲突，不能破坏设备。

3. **设备管理的任务** **重点**

   - **设备驱动**：**逻辑设备名**转换成设备的**物理地址**，启动指定的 I/O 设备，完成程序规定的 I/O 操作，并对由设备发来的中断请求进行及时响应，根据中断类型进行相应的处理。
   - **设备无关性**：用户在编制程序时，应尽量避免直接使用实际的设备名而使用逻辑设备名。这样，有利于解决设备的故障和增加设备分配的灵活性。
   - **虚拟设备**：把**一次仅允许一个进程使用的设备称为独占设备**。独占设备使得系统效率很低，并且可能产生死锁。**以大容量外存为支持，通过虚拟技术将一台独占设备改造成能被多个进程共享的设备，以提高设备的利用率。**这种经过虚拟技术改造后的设备，即**逻辑上的设备称为虚拟设备。**
   - **缓冲管理**： CPU 与设备之间、设备与设备之间交换信息时，**都要利用缓冲区来缓解速度不匹配的矛盾，提高 CPU 与设备之间、设备与设备之间操作的并行程度。**
   - **设备分配**：系统根据进程所请求的设备类型，**按分配算法对设备和设备相应的控制器和通道进行分配，建立从设备到内存之间传输信息的通路**。 在进程的 I/O 完成后，系统应及时回收设备，以便重新分配给其他进程使用。将未获得所需设备的进程放进相应设备的等待队列。

4. I/O设备类型
    - 人可读设备
        - 和用户进行交互
        - 打印机
        - 视频显示终端
            - 显示器
            - 键盘
            - 鼠标
    - 机器可读设备
        - 和电子设备交互
        - 磁盘和磁带
        - 传感器
    - 通信设备
        - 用来和远程设备通信
        - 网卡
        - Modems

5. **I/O设备的差异**
    - **数据传输速率**
        - 可能会相差几个数量级
    - 应用差别
        - 磁盘用来存储文件，需要文件管理系统
        - 磁盘存储虚拟内存页面，因此需要特殊的软硬件支持
        - 系统管理员使用的终端具有更高的优先权

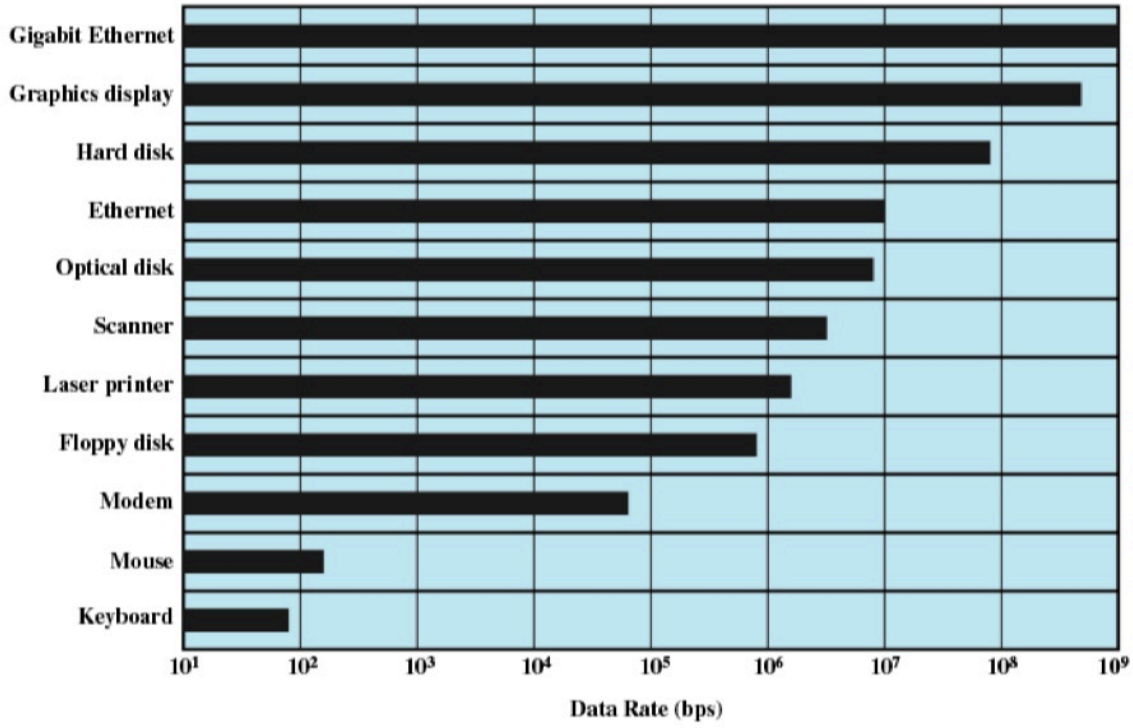


Figure 11.1 Typical I/O Device Data Rates

- 控制复杂性
- **传输单元**
  - 数据可以按照字**节流或者字符流**的形式传送（比如键盘），也可以按照更大的**块**来传送（比如磁盘）
- 数据表示
  - 编码方式
- 错误条件
  - 产生错误的原因、报告错误的方式、错误造成的后果以及影响的有效范围

6. I/O组织

- **可编程 I/O**
  - 处理器代表进程给 I/O 模块发送一个命令，该进程进入忙等待，等待操作的完成，然后才可以继续执行
- **中断驱动 I/O** **重点**
  - 处理器代表进程给 I/O 模块发送 I/O 命令
  - 然后，处理器执行后续指令
  - 模块完成工作后，向处理器发送中断信号
- **直接存储器访问 Direct Memory Access**

        - **DMA 模块控制主存和 I/O 模块之间的数据交换**
        - 只有当需要传送的数据传送完毕，才向处理器发送中断信号
    - 各种技术之间的关系

Table 11.1 I/O Techniques

	No Interrupts	Use of Interrupts
I/O-to-memory transfer through processor	Programmed I/O	Interrupt-driven I/O
Direct I/O-to-memory transfer		Direct memory access (DMA)

7. I/O功能的发展

    - 处理器直接控制外围设备
    - 增加了处理器或者 I/O 模块
        - Processor uses programmed I/O without interrupts
        - 处理器不需要处理外设的细节， Processor does not need to handle details of external devices
    - **中断方式的控制器或者 I/O 模块**
        - **处理器无需花费时间等待一个 I/O 操作**
    - **Direct Memory Access**
        - 在没有处理器参与的情况下，**从主存中移出或者往主存中移入一块数据**
        - **仅在传送开始和结束时需要用到处理器**
    - I/O 模块被增强为一个单独的处理器
    - **I/O 模块不仅拥有处理器还拥有自己的局部存储器**
        - I/O module has its own local memory
        - Its a computer in its own right
    - 后三种都指向一种技术 —— **DMA**

8. **Direct Memory Access**

> 只在开始和结束与 CPU 通信

    - （开始）处理器把 I/O 操作委托给 DMA 模块
    - DMA module transfers data directly to or from memory

- When complete DMA module sends an **interrupt signal to the processor**（结束）

DMA

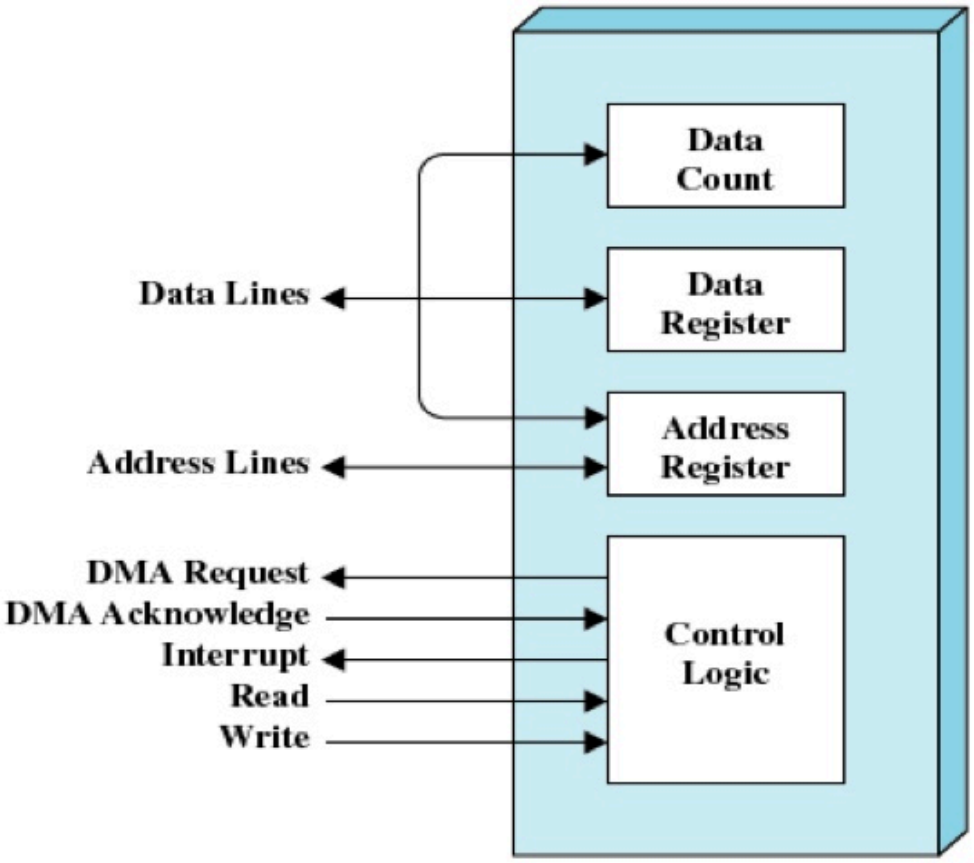
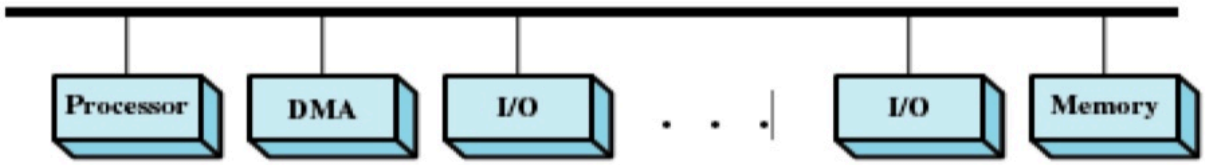
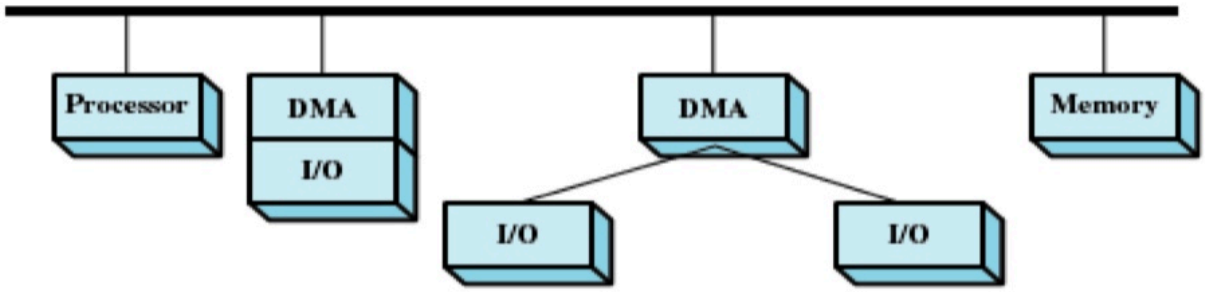


Figure 11.2 Typical DMA Block Diagram

DMA Configurations

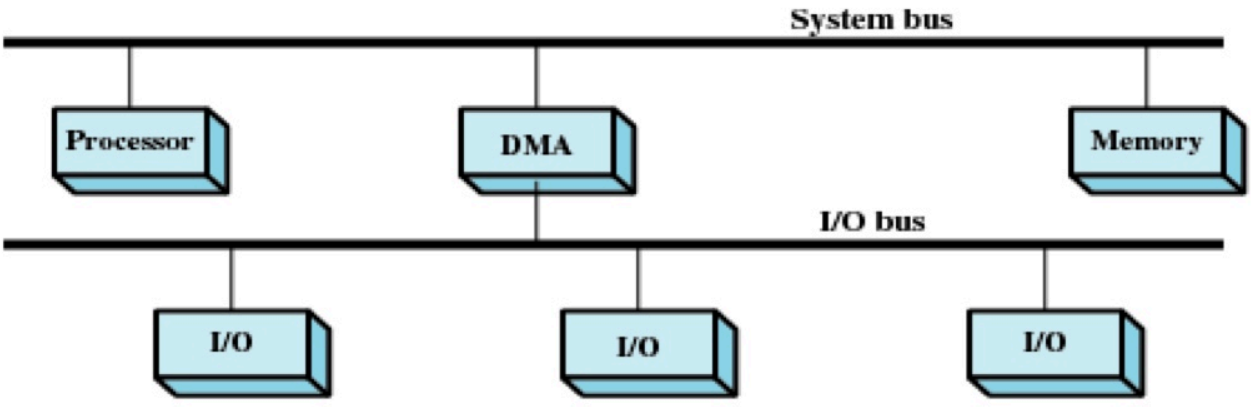


(a) Single-bus, detached DMA



(b) Single-bus, Integrated DMA-I/O

DMA Configurations



(c) I/O bus

Figure 11.3 Alternative DMA Configurations

- **兼顾效率和通用性**
- 效率
  - 大多数I/O设备要远远慢于主存
    - 多道程序设计允许在一个进程运行的同时，其他一些进程等待 I/O
  - 矛盾依然存在 ——I/O跟不上处理器的速度
    - 交换技术用于将额外的就绪进程加载到主存，从而保持处理器的工作状态，但是就交换技术本身而言，它也是一个 I/O 操作
- 通用性
  - 期望使用一种方式处理所有的 I/O 设备
  - Hide most of the details of device I/O in lower-level routines so that processes and upper levels see devices in general terms such as read, write, open, close, lock, unlock
- 硬件抽象 —— 逻辑I/O

10. **I/O缓冲**

- 简单方法 —— **忙等待或挂起**
  - 执行一个 I/O 命令，并等待数据传输完毕
- 缓冲的原因
  - **进程在等待较慢的 I/O 时会被挂起**
  - 可能会干扰操作系统的决策： I/O 过程中某些页面仍然要保存在内存中，否则某些数据就可能丢失

- 面向块

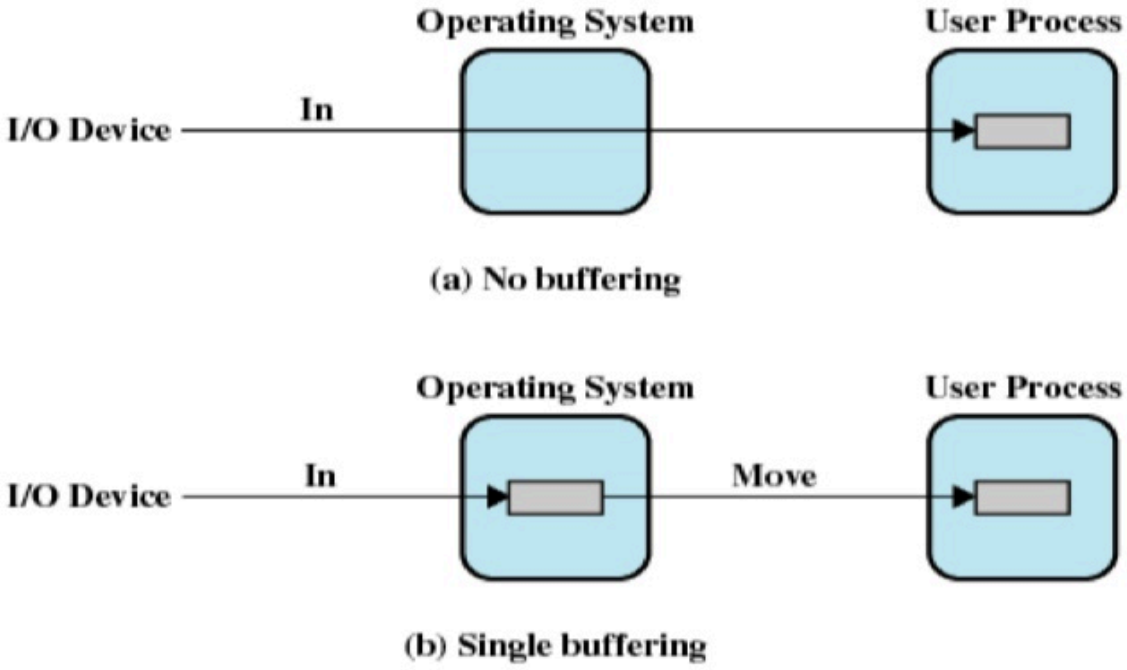
  - Information is stored in fixed sized blocks
  - Transfers are made a block at a time
  - Used for **disks and tapes**

- 面向流

  - Transfer information as **a stream of bytes**
  - Used for terminals, printers, communication ports, mouse and other pointing devices, and most other devices that are not secondary storage

11. **Single Buffer**

Single Buffer

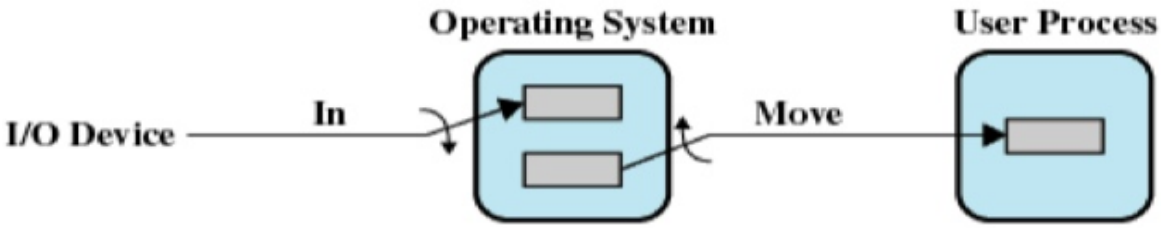


- Operating system assigns **a buffer in main memory** for an I/O request
- Block-oriented
  - Input transfers made to buffer
  - Block moved to user space when needed
  - Another block is moved into the buffer
    - Read ahead
- Block-oriented
  - User process can process one block of data while next block is read in
  - Swapping can occur since input is taking place in system memory, not user memory
  - Operating system keeps track of assignment of system buffers to user processes —— 增加了系统的逻辑复杂度
- Stream-oriented
  - Used a line at time
  - User input from a terminal is one line at a time with carriage return signaling the end of the line

- Output to the terminal is one line at a time

12. Double Buffer

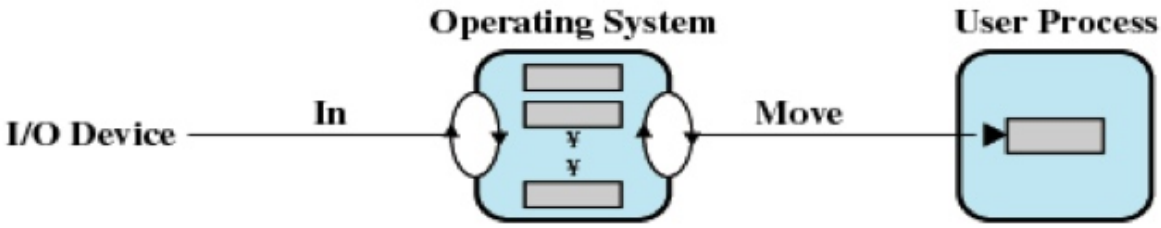
    - Use two system buffers instead of one
    - A process can transfer data to or from one buffer while the operating system empties or fills the other buffer



(c) Double buffering

13. Circular Buffer

    - More than two buffers are used
    - Each individual buffer is one unit in a circular buffer
    - Used when I/O operation must keep up with process



(d) Circular buffering

14. 磁盘调度

    - 磁盘性能参数
    - 磁盘调度策略
    - RAID， 独立磁盘冗余阵列
    - 磁盘缓冲

15. 磁盘性能参数

- 为了读和写，磁头必须定位于指定的磁道和该磁道中的指定的扇区的开始处
- **Seek time 寻道时间***
  - Time it takes to position the head at the desired track
- Rotational delay or rotational latency
  - **旋转延迟**
  - Time its takes for the beginning of the sector to reach the head
- 磁盘I/O传送的时序

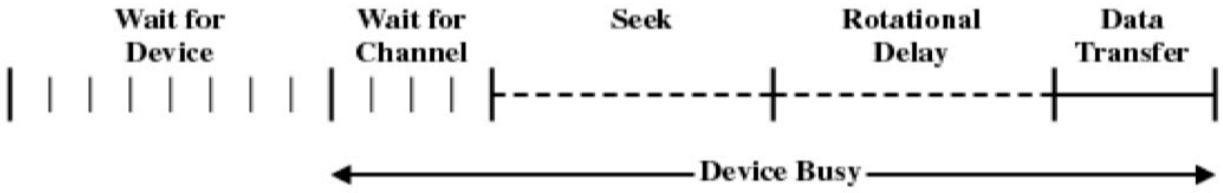
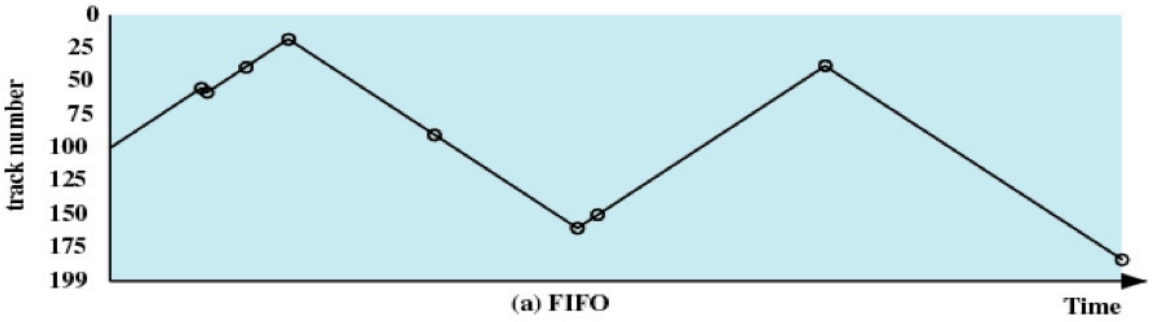


Figure 11.6 Timing of a Disk I/O Transfer

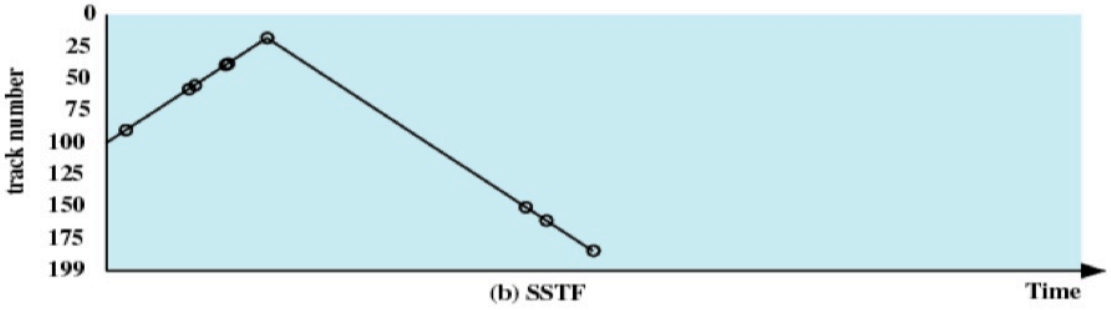
- Access time 存取时间
  - **Sum of seek time and rotational delay**
  - The time it takes to get in position to read or write
- Data transfer occurs as the sector moves under the head
- Transfer Time 传送时间

16. 磁盘调度策略

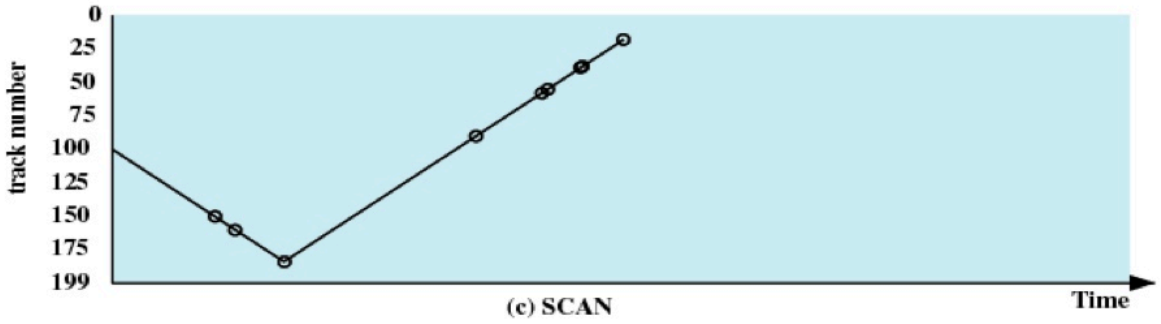
- Seek time is the reason for differences in performance
- For a single disk there will be a number of I/O requests
- If requests are selected randomly, we will poor performance
- 假设磁盘有 200 个磁道，磁头当前在 100 处， 请求队列中是一些随机请求，队列为：55,58,39,18,90,160,150,38,184
- 以下为算法
- **First-in, first-out (FIFO)**
  - Process request sequentially
  - Fair to all processes
  - Approaches random scheduling in performance if there are many processes



- **Priority**
  - Goal is not to optimize disk use but to meet other objectives
  - Short batch jobs may have higher priority
  - Provide good interactive response time
- **Last-in, first-out**
  - Good for transaction processing systems
    - The device is given to the most recent user so there should be little arm movement
  - Possibility of starvation since a job may never regain the head of the line
- **Shortest Service Time First**
  - Select the disk I/O request that requires the least movement of the disk arm from its current position
  - Always choose the minimum Seek time

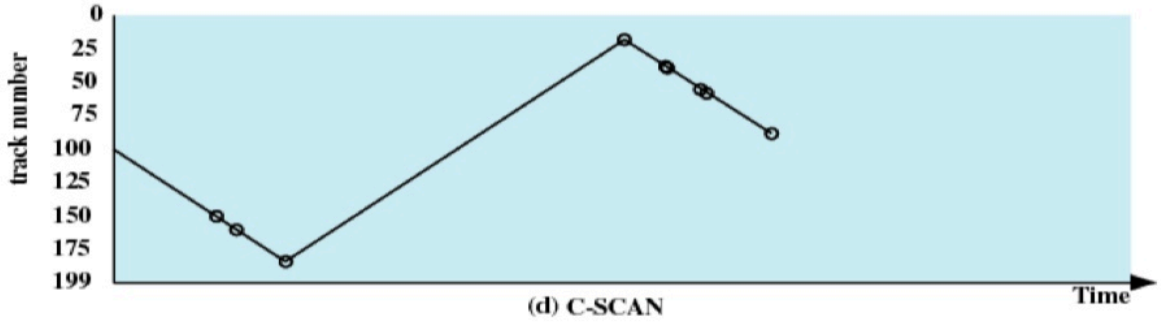


- SCAN
  - **Arm moves in one direction only**, satisfying all outstanding requests until it reaches the last track in that direction
  - Direction is reversed



(c) SCAN

- C-SCAN
  - Restricts scanning to one direction only
  - When the last track has been visited in one direction, **the arm is returned to the opposite end of the disk and the scan begins again**



(d) C-SCAN

- 时间对比

(a) FIFO (starting at track 100)		(b) SSTF (starting at track 100)		(c) SCAN (starting at track 100, in the direction of increasing track number)		(d) C-SCAN (starting at track 100, in the direction of increasing track number)	
Next track accessed	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**	**Next track accessed**	**Number of tracks traversed**
55	45	90	10	150	50	150	50
58	3	58	32	160	10	160	10
39	19	55	3	184	24	184	24
18	21	39	16	90	94	18	166
90	72	38	1	58	32	38	20
160	70	18	20	55	3	39	1
150	10	150	132	39	16	55	16
38	112	160	10	38	1	58	3
184	146	184	24	18	20	90	32
Average seek length	55.3	**Average seek length**	27.5	**Average seek length**	27.8	**Average seek length**	35.8

17. **RAID 独立硬盘冗余阵列**

- Redundant Array of Independent Disks

- 阵列有一组物理磁盘，但操作系统把它看作一个单个的逻辑驱动器
- 数据分布在物理驱动器中
- 使用冗余的磁盘容量保存奇偶校验信息， 从而保证当一个磁盘失败时，数据具有可恢复性
- 已经形成标准，有 7 层
- 具体7层见书或wiki

18. 磁盘缓冲区Disk Cache

- Buffer in main memory for disk sectors
- Contains a copy of some of the sectors on the disk