# 《操作系统》期末速通教程

# 5. 设备管理

## 5.1 I/O 系统

**[I/O 系统]**

(1) 管理对象: I/O 设备.

(2) 必要性:

① 设备种类多, 物理特性多样, 控制复杂.

② 设备与 CPU 速度不匹配.

(3) 基本任务:

① 完成 I/O 请求.

② 提高 I/O 效率.

③ 提高 I/O 设备利用率.

(4) 主要功能:

① 缓冲区管理.

② 设备分配.

③ 设备处理.

④ 虚拟设备.

⑤ 实现设备独立性.

[**I/O 设备的分类**]

(1) 按传输效率分类:

① 低速设备: 如键盘、鼠标、语音输入输出等.

② 中速设备: 如行式打印机、激光打印机.

③ 高速设备: 磁带机、磁盘机、光盘机.

(2) 按信息交换单位分类:

① 块设备:

(i) 定义: 信息交换以数据块为单位.

(ii) 例: 磁盘、磁带.

(iii) 特点:

i) 传输效率高.

ii) 可寻址, 支持随机存取.

iii) I/O 常用 DMA 方式.

② 字符设备:

(i) 定义: 信息交换以字符为单位.

(ii) 例: 交互式终端机、打印机.

(iii) 特点:

i) 数据无结构.

ii) 传输效率低.

iii) 不可寻址.

iv) I/O 常用中断驱动方式.

(3) 按共享属性分类:

① 独占设备:

(i) 定义: 一个时间段内只允许一个用户使用的设备.

(ii) 例: 大部分低速设备, 如打印机.

② 共享设备:

(i) 定义: 一个时间段内允许多个用户并发使用.

(ii) 例: 磁盘.

③ 虚拟设备: 通过 SPOOLing 技术将独占设备改造为共享设备, 即将一个物理设备变为多个逻辑设备.

[**注 1**] 共享设备必须可寻址且支持随机访问, 否则无法保证数据的完整性和一致性.

[**注 2**] 设备管理需要考虑的因素:

① 设备的固有属性: 决定设备的使用方式.

② 设备独立性.

③ 安全性: 保证设备分配时不会被永久阻塞.

[**注 3**] 磁带机不是共享设备, 因为磁带机旋转至所需读写的位置耗时长, 若一个时间间隔内被多个进程访问, 则只能一直旋转, 无法读/写.

[**注 4**] 独占设备一般采用静态分配方式, 共享设备一般采用动态分配方式.

[**注 5**] 具备设备独立性的系统中, 设备可视为特殊的文件, 可用文件名访问设备.

[**注 6**] 具备设备独立性的系统中, 更换物理设备后无需修改访问该设备的应用程序, 只需更换设备的驱动程序.

[**I/O 接口**, **I/O 控制器**]

功能:

① 作为 CPU 与设备间的接口.

② 控制多个 I/O 设备.

③ 实现 I/O 设备与主机的数据交换.

④ 编址设备, 通过 I/O 地址识别不同设备.

[**I/O 通道**]

(1) 定义: **I/O 通道**是一种特殊的处理机, 可执行通道指令.

(2) 功能: 实现 CPU 、通道和 I/O 设备并行工作, 提高系统资源利用率.

(3) I/O 通道与一般处理机的区别:

① I/O 通道的指令类型单一, 只能执行 I/O 相关指令.

② I/O 通道无自己的内存, 通道程序放在主机的主存, 即通道与 CPU 共享内存.

(4) I/O 通道与 DMA 方式的区别:

① DMA 方式需 CPU 控制传输的数据块大小、传输的内存位置; I/O 通道中这些信息由 I/O 通道控制.

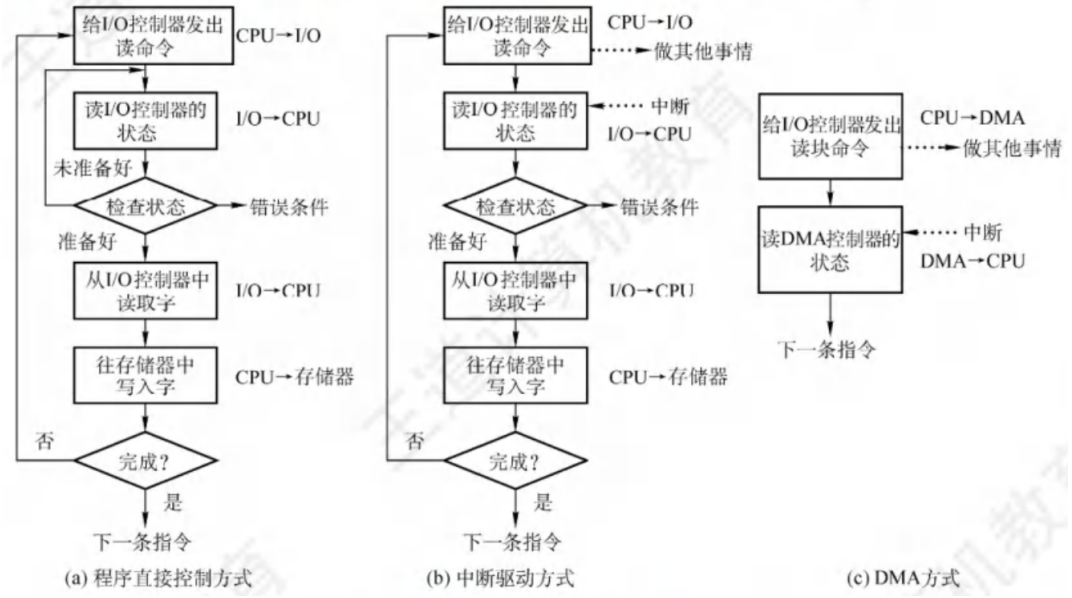
② 一个 DMA 控制器对应一台设备与内存传递数据; 一个 I/O 通道可控制多台设备与内存的数据交换.

[**注 1**] 通道是硬件, 而缓冲池、SPOOLing、内存覆盖都是在内存的基础上通过软件实现.

[**注 2**] 对于同一组指令, 设备控制器、通道和设备不会并行工作, 因为 CPU 要么给通道发命令, 要么给设备控制器发命令.

[**I/O 控制方式**]

(1) 分类:




① 程序直接控制方式.

② 中断驱动方式.

③ DMA 方式.

(2) **程序直接控制方式 (程序轮询方式)** :

① 策略: CPU 轮询 I/O 设备.

② 优点: 实现简单.

③ 缺点:

(i) CPU 绝大部分时间忙等.

(ii) CPU 与 I/O 设备只能串行工作, 资源利用率低.

(3) **中断驱动方式**:

① 策略:

(i) I/O 设备可主动打断 CPU 的运行并请求服务.

(ii) CPU 向设备控制器发出一条 I/O 指令后即可继续其它工作.

② 优点:

(i) CPU 无需轮询 I/O 设备, 不忙等.

(ii) CPU 与 I/O 设备可并行工作, 资源利用率高.

③ 缺点:

(i) 设备与内存的数据交换都需经过 CPU 的寄存器.

(ii) CPU 以字 (或字节) 为单位, 将该方式应用于块设备的 I/O 极其低效.

(4) **直接内存访问方式** (**Direct Memory Access**, **DMA**) :

① 策略:

(i) 在 I/O 设备和内存间开辟直接的数据交换通路.

(ii) 数据通路仅是逻辑上的, 并未建立实际的物理通路, 数据交换通常通过总线进行.

(iii) CPU 只在传输的开始时 (预处理) 和结束时 (后处理) 干预.

② 阶段:

(i) 预处理: CPU 初始化 DMA 控制器中的寄存器、设置传送方向、测试并启动设备等.

(ii) 数据传送: 完全由 DMA 控制, DMA 控制器接管系统总线.

(iii) 后处理: DMA 控制器向 CPU 发送中断请求, CPU 执行中断服务程序, 结束 DMA 处理.

③ 优点:

(i) 数据传输以块为单位, 效率高.

(ii) CPU 介入频率低.

(iii) 设备与内存的数据交换无需经过 CPU 的寄存器.

(iv) CPU 与 I/O 设备的并行性高.

[**注 1**] 块设备 (如磁盘) 的 I/O 控制主要采用 DMA 方式.

[**注 2**] I/O 控制方式中, 中断方式和 DMA 方式会导致用户进程进入阻塞态.

---

# 5.2 缓冲管理

[**缓冲管理**]

(1) 引入缓冲的目的:

① 缓和 CPU 与 I/O 设备间速度不匹配的矛盾.

② 减少 CPU 中断频率, 放宽 CPU 中断响应的时间限制.

③ 提高 CPU 与 I/O 设备的并行性.

④ 解决基本数据单元大小不匹配的问题.

(2) 分类:

① 单缓冲.

② 双缓冲.

③ 循环缓冲.

④ 缓冲池.

[**注 1**] 单缓冲、双缓冲、循环缓冲都是专用缓冲.

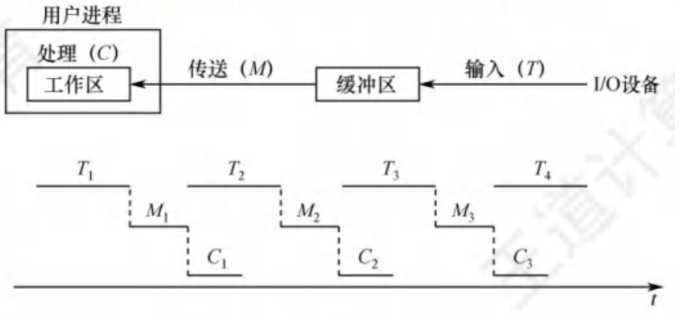
[**注 2**] 为使多个并发进程有效地输入输出, 缓冲技术应采用缓冲池.

[**注 3**] 若 I/O 耗时远低于 CPU 耗时, 则缓冲区几乎无效.

[**注 4**] 缓冲区是临界资源, 需实现进程访问缓冲区的同步.

[**单缓冲**]

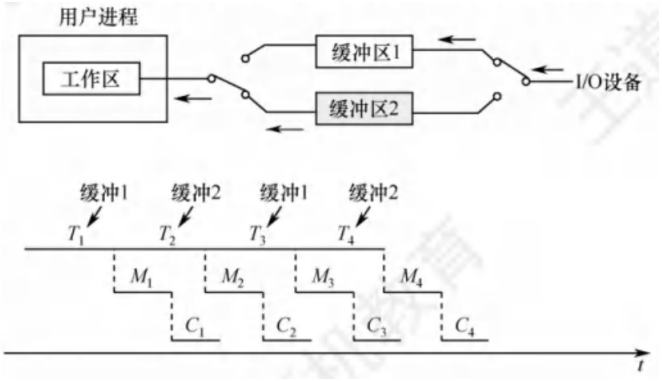
(1) 示意图:




(2) 处理每块数据的平均时间为 $\max\{C, T\} + M$ .

[**双缓冲**]

(1) 示意图:




(2) 处理每块数据的平均时间为 $\max\{C + M, T\}$ .

[**例**] 某文件占用 $10$ 个磁盘块. 现将该文件的磁盘块依次读入主存缓冲区, 并送到内存区分析. 一个缓冲区的大小等于一个盘块的大小, 将一个盘块读入缓冲区的时间 $T = 100\ \mu\text{s}$ , 将缓冲区中的数据传送到用户区的时间为 $M = 50\ \mu\text{s}$ , CPU 分析一块数据的时间为 $C = 50\ \mu\text{s}$ . 采用如下的缓冲结构时, 求读入并分析完该文件所需的时间:

[1] 单缓冲.

[2] 双缓冲.

[**解**]

[1] 处理一块的平均时间 $\max\{C, T\} + M = 150\ \mu\text{s}$ .

总时间 $=$ 处理 $10$ 块的平均时间 $+$ 处理最后一块的时间 $= 10 \times 150\ \mu\text{s} + 50\ \mu\text{s} = 1550\ \mu\text{s}$ .

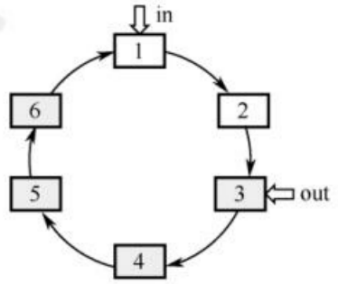
[2] 处理一块的平均时间 $\max\{C + M, T\} = 100\ \mu\text{s}$ .

总时间 $=$ 处理 $10$ 块的平均时间 $+$ 读入最后一块的时间 $+$ 处理最后一块的时间

$= 10 \times 100\ \mu\text{s} + 50\ \mu\text{s} + 50\ \mu\text{s} = 1100\ \mu\text{s}$ .

[**循环缓冲**]

(1) 示意图:



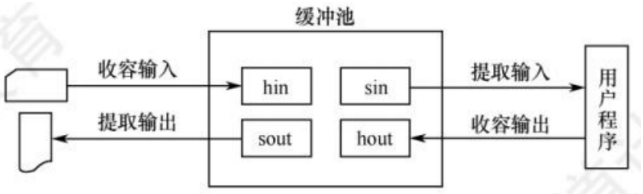

(2) 指针:

① in 指针指向首个可输入数据的空缓冲区.

② out 指针指向首个可提取数据的满缓冲区.

[**缓冲池**]

(1) 示意图:




(2) 三种缓冲区: 空闲、输入数据、输出数据.

(3) 三个队列: 空缓冲队列、输入队列、输出队列.

(4) 四种工作方式: 收容输入、提取输入、收容输出、提取输出.

[**例**] 分析单用户计算机上下列 I/O 操作是否需使用缓冲.

[1] 图形用户界面下使用鼠标.

[2] 多任务 OS 下的磁带驱动器 (假设无设备预分配) .

[3] 包含用户文件的磁盘驱动器.

[4] 使用存储器映射 I/O , 直接和总线相连的图形卡.

[**答**]

[1] 需要. 有更高优先级的操作时需记录鼠标活动情况.

[2] 需要. 磁盘驱动器和目标或源 I/O 设备的吞吐量不同, 需采用缓冲区.

[3] 需要. 处理数据在用户作业空间与磁盘间的传递.

[4] 需要. 交换帧缓冲 (双缓冲) .

# 5.3 I/O 软件

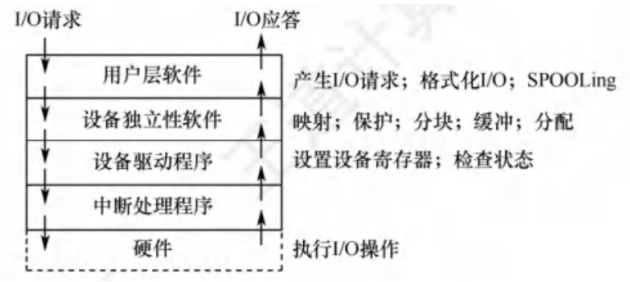
[**I/O 软件**]

(1) 目标:

① 保证 CPU 与 I/O 设备并发, 提高资源利用率.

② 提供简单、抽象、清晰、统一的接口, 统一标准, 规范操作.

(2) I/O 层次结构:




(3) **设备独立性软件** (又称**设备无关性软件**) 负责实现**设备无关性**.

① **设备无关性**:

(i) 定义: 用户编程时使用的设备与实际设备无关, 即程序中只说明使用的设备的类型.

(ii) 优点:

i) 方便编程.

ii) 使程序运行不受具体机器环境的限制.

iii) 便于程序移植.

② 设备独立性软件向上层提供系统调用的接口, 根据设备类型选择响应的驱动程序, 将系统调用的参数翻译为设备操作命令.

(4) **设备驱动程序**:

① 工作:

(i) 初始化设备: 设置数据传输方式和控制设备的工作状态.

(ii) 写设备寄存器.

(iii) 执行 I/O 命令.

(iv) 检查 I/O 过程是否出错.

(v) 处理与设备相关的中断.

(vi) 管理 I/O 设备.

(vii) 计算数据所在磁盘的柱面号、磁头号、扇区号.

② 不同 OS 有不同的驱动程序接口, 设备驱动程序需根据 OS 的要求定制.

[**注**] 分析和缓冲 I/O 设备传回的消息由进程或 OS 完成.

[**例**] 在接收和处理输入设备的中断的过程中, 下列哪些操作一定不由硬件完成: ① 判断产生中断的类型; ② CPU 从用户态切换到内核态; ③ 主机获取设备输入; ④ 保存用户程序断点.

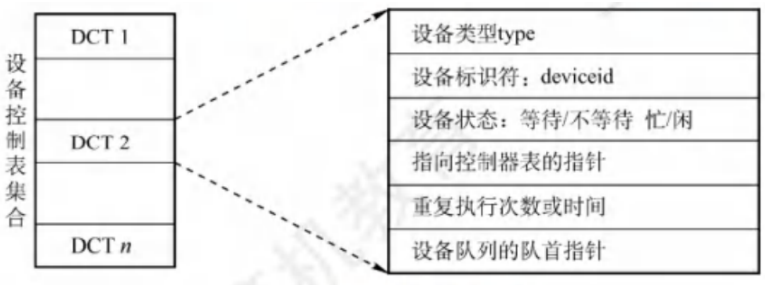
[**答**] ③ , 中断服务程序完成数据的输入输出.

① 可用硬件识别法. ② 和 ④ 在中断响应阶段由硬件完成.

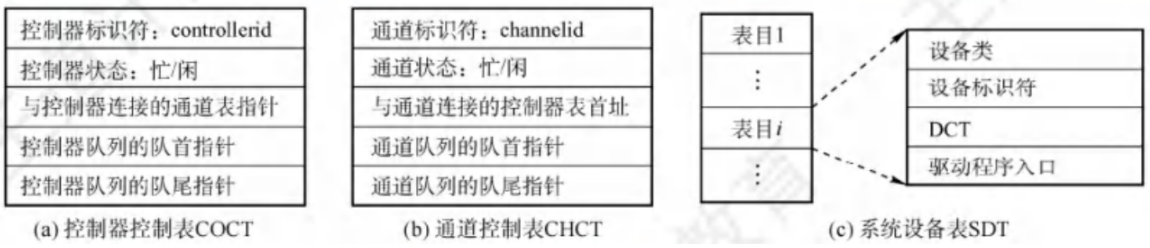
[**设备分配**]

(1) 设备分配的数据结构:

① **设备控制表** (**Device Control Table**, **DCT**) :




② **控制器控制表** (**Controller Control Table**, **COCT**) 、**通道表** (**Channel Control Table**, **CHCT**) 、**系统设备控制表** (**System Device Table**, **SDT**) :




(2) 分配设备过程:

① 分配设备.

② 分配设备控制器.

③ 分配通道.

[**假脱机技术**, **SPOOLing**]

(1) 主要目的: 提高独占设备利用率.

(2) 策略:

① 模拟脱机输入输出, 将独占设备改造为共享设备, 即将一个物理设备变为多个逻辑设备.

② 系统控制设备与输入井/输出井间的数据传送.

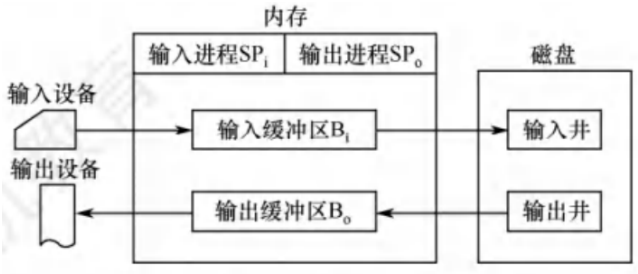
(3) 前提:

① 多道程序系统.

② 大容量的高速随机外存, 通常采用磁盘存储技术.

③ SPOOLing 软件.

(4) 示意图:




(5) 优点:

① 缓和 CPU 与 I/O 设备速度不一致的矛盾.

② 提高 I/O 速度: 将对 I/O 设备的操作转化为对磁盘缓冲区的操作.

③ 将独占设备改造为共享设备, 提高独占设备利用率.

④ 实现了虚拟设备.

[**注 1**] SPOOLing 系统中, 用户的打印结果现被送到磁盘的固定区域.

[**注 2**] SPOOLing 系统中, 用户进程实际分配到的是外存区, 即虚拟设备.

[**注 3**] SPOOLing 系统中, 进程不必等待 I/O , 只需将数据放入输入井或输出井后, 即可执行其它操作.

# 5.4 磁盘管理

## 5.4.1 磁盘的存取时间

**[磁盘的存取时间]**

(1) **寻道时间** $T_s$ :

① 定义: 磁头移动到目标磁道所需的时间.

② 计算: 设启动磁臂的时间 $s$ , 移动 $n$ 条磁道, 则 $T_s = m \times n + s$ ,

其中 $m$ 是与磁盘驱动器有关的常数.

(2) **旋转延迟时间** $T_r$ :

① 定义: 磁头定位到要读/写扇区的时间, 即扇区旋转到磁头下的时间.

② 计算: 设磁盘旋转速度为 $r$ , 则 $T_r = \dfrac{1}{2r}$ , 即平均旋转半周.

③ 影响因素: 文件的物理结构, 即物理地址是否连续.

(3) **传输时间** $T_t$ :

① 定义: 从磁盘读出数据或向磁盘写入数据的时间.

② 计算: 设每次读/写的字节数为 $b$ , 磁盘旋转速度为 $r$ , 一个磁道上的字节数为 $N$ ,

则 $T_t = \dfrac{b}{rN}$ , 即 $b$ B 平均占用 $\dfrac{b}{N}$ 个磁道.

③ 影响因素: 扇区处理时间.

(4) **总平均存取时间** $T_a = T_s + T_r + T_t = T_s + \dfrac{1}{2r} + \dfrac{b}{rN}$ .

(5) 磁盘的存取时间中:

① 寻道时间占大头, 与磁盘调度算法相关.

② 旋转延迟时间和传输时间由硬件决定.

(6) 提高磁盘 I/O 效率的方法:

① 提升磁盘的硬件性能.

② 采用更好的磁盘调度算法.

③ 设置磁盘高速缓冲.

④ 提前读: 由局部性原理, 读取一个盘块时将邻近的盘块提前读入内存, 如预调页策略.

⑤ 延迟写: 修改过的页不立即写回磁盘, 而是积累一定数量后一次性写回, 减少磁盘 I/O 次数.

⑥ 优化物理块分布, 使得同一文件的物理块尽量集中.

⑦ 虚拟盘: 利用内存或其它存储介质仿真磁盘, 如内存式硬盘和固态硬盘 (SSD) .

**[注 1]** 存储文件时, 若一个磁道存不下, 则最好存在同一柱面上的不同盘面, 这样无需移动磁道, 文件访问效率高.

[**注 2**] 改善磁盘设备 I/O 性能的方法:

① 重排 I/O 请求次序.

② 预读、延迟写.

③ 优化文件物理块分布.

注意在磁盘上设置多个分区不能提高磁盘 I/O 性能, 反而使得处理复杂, 降低 I/O 利用率.

[**磁盘高速缓存**, **Disk Cache**]

(1) 策略: 利用内存中的存储空间, 暂存从磁盘中读出的一系列盘块信息.

(2) 磁盘高速缓存逻辑上属于磁盘, 物理上驻留在内存中的盘块.

(3) 磁盘高速缓存在内存中的两种形式:

① 内存中单独开辟空间作为磁盘高速缓存, 其大小固定, 不受应用程序多少影响.

② 将所有未利用的内存空间变为缓冲池, 供请求分页系统和磁盘高速缓存共享.

[**例**] 某磁盘的转速为 $7200\ 转/分$, 每个磁道有 $160$ 个扇区, 每个扇区大小为 $512\ \mathrm{B}$. 求理想状态下的数据传输效率.

[**解**] $转速 = 7200\ 转/分 = 120\ 转/\mathrm{s}$, 转一圈经 $160$ 个扇区, 每个扇区 $512\ \mathrm{B}$,

则 $数据传输效率 = 120 \times 160 \times 512\ \mathrm{KB/s} = 9600\ \mathrm{KB/s}$.

---

## 5.4.2 磁盘调度算法

[**磁盘调度算法**]

(1) 原因: 磁盘是共享设备, 允许多个进程访问, 故需进行磁盘调度.

(2) 目标: 减少磁盘的平均寻道时间.

(3) 分类:

① **先来先服务算法** (FCFS) .

② **最短寻道时间优先算法** (SSTF) .

③ **扫描算法** (SCAN) .

④ **循环扫描算法** (C-SCAN) .

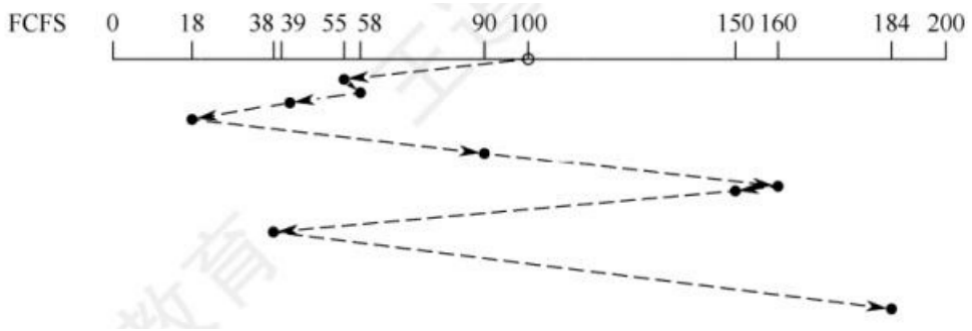
[**注**] 上述四个算法中, 只有 FCFS 算法无 "磁臂粘着" 现象.

**[先来先服务算法, First Come First Served, FCFS]**

(1) 策略: 按访问磁盘的先后调度.

(2) 例: 磁头初始时在 $100$ 号磁道.

磁道请求: $55$、$58$、$39$、$18$、$90$、$160$、$150$、$38$、$184$.




磁头移动了 $45+3+19+21+72+70+10+112+146=498$ 个磁道,

$$\text{平均寻道长度}=\frac{489}{9}\approx 55.3\ .$$

(3) 优点:

① 实现简单.

② 有公平性.

③ 访问磁盘的进程少且访问集中时, 性能好.

④ 无 "磁臂粘着" 现象.

(4) 缺点:

① 大量进程访问磁盘时, FCFS 的性能接近随机调度.

② 平均寻道时间较大.

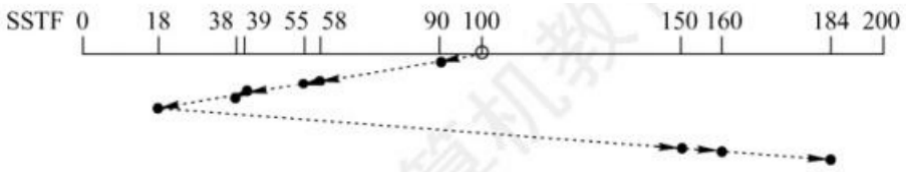
③ 只适用于磁盘 I/O 较少的场合.

**[最短寻道时间优先算法, Shortest Seek Time First, SSTF]**

(1) 策略: 每次访问离当前磁头位置最近的磁道.

(2) 例: 磁头初始时在 $100$ 号磁道.

磁道请求: $55$、$58$、$39$、$18$、$90$、$160$、$150$、$38$、$184$.




磁头移动了 $10+32+3+16+1+20+132+10+24=248$ 个磁道,

$$\text{平均寻道长度}=\frac{248}{9}\approx 27.5\ .$$

(3) 优点: 平均寻道时间较短.

(4) 缺点:

① 每次的寻道时间最短不能保证平均寻道时间最短.

② 可能饥饿, 即某些进程长时间得不到访问.

③ 有 "磁臂粘着" 现象, 即磁头长时间停留在同一磁道.

**[扫描算法, SCAN, 电梯调度算法]**

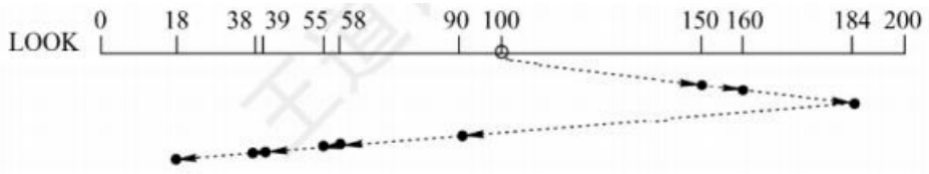
(1) 策略:

① 规定磁头只有到最外侧磁道才能向内侧移动, 只有到最内侧磁道才能向外侧移动.

② **LOOK 调度算法**: 磁头移动到最远端的请求即可转向, 无需到达磁盘端点.

(2) 例: 磁头初始时在 $100$ 号磁道, 且向磁道号增大的方向运动.

磁道请求: $55$、$58$、$39$、$18$、$90$、$160$、$150$、$38$、$184$.




(3) 优点:

① 不饥饿.

② 平均寻道时间较短.

(4) 缺点:

① 有 "磁臂粘着" 现象.

② 偏向于处理最内侧和最外侧的请求.

③ 对最近扫过的磁道不公平, 即对与当前磁头距离近但在磁头移动的反方向的磁道不公平.

④ 访问局部性方面不如 FCFS 算法和 SSTF 算法.

[**循环扫描算法, C-SCAN**]

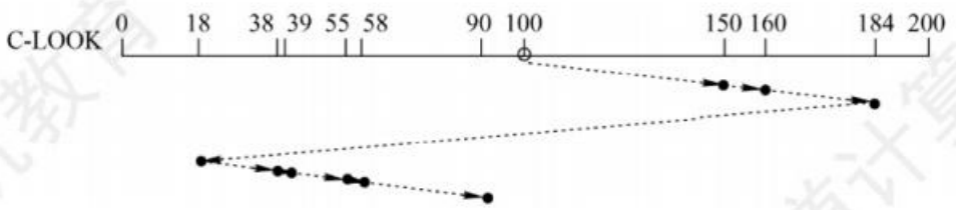
(1) 策略:

① 在 SCAN 调度算法的基础上, 规定磁头到达最外 (内) 侧的磁道后回到最内 (外) 侧的磁道, 途中不处理请求.

② **C-LOOK 调度算法**: 磁头处理完最接近内 (外) 侧的磁道的请求后回到最靠外 (内) 的请求的磁道, 途中不处理请求.

(2) 例: 磁头初始时在 $100$ 号磁道, 且向磁道号增大的方向运动.

磁道请求: $55$、$58$、$39$、$18$、$90$、$160$、$150$、$38$、$184$.




(3) 优点:

① 不饥饿.

② 平均寻道时间较短.

③ 最长等待时间比 SCAN 算法少一半.

④ 对各磁道公平.

(4) 缺点: 有 "磁臂粘着" 现象.