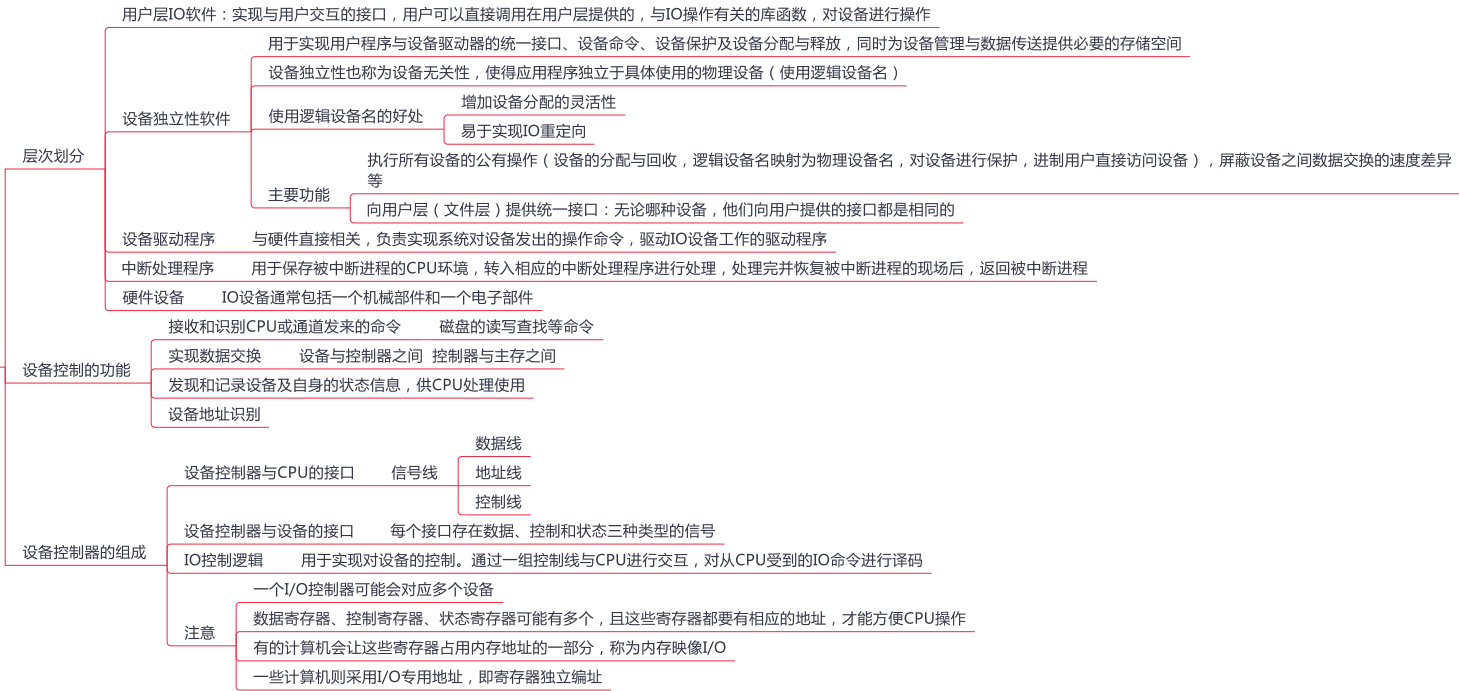


5.1 IO管理概述（上）



5.1 IO管理概述（下）

I/O子系统的层次结构



5.2 IO核心子系统（上）



5.2 IO核心子系统（下）

IO调度

- 设备分类
 - 独占设备——一个时段只能分配给一个进程（如打印机）
 - 共享设备——可同时分配给多个进程使用（如磁盘），各进程往往是宏观上同时共享使用设备，而微观上交替使用
 - 虚拟设备——采用 SPOOLing 技术将独占设备改造成虚拟的共享设备，可同时分配给多个进程使用（如采用 SPOOLing 技术实现的共享打印机）

- 设备分配的安全性
 - 安全分配方式
 - 进程发出IO请求后便进入阻塞态，直到IO结束才被唤醒
 - 优点：设备分配安全
 - 缺点：CPU和IO设备串行工作
 - 不安全分配方式
 - 进程发出IO请求后继续运行，需要时发出第二个，第三个IO请求
 - 优点：进程推进迅速
 - 缺点：可能产生死锁

- 逻辑设备名到物理设备名的映射
 - 目的
 - 提高设备分配的灵活性和利用率
 - 实现IO重定向
 - 引入设备独立性
 - 实现方法：引入逻辑设备表(LUT)，用来将逻辑设备名映射为物理设备名
 - 建立方式
 - 整个系统设置一张LUT，所有设备分配情况都记录在这张表上（单用户设备）
 - 每个用户建立一张LUT，分别记录设备的分配情况

SPOOLING技术（假脱机技术）

- 目的
 - 缓解CPU 与IO 的速度差异矛盾
- 要实现SPOOLing 技术，必须要有多个道程序技术的支持
- 输入井和输出井
 - 输入井用来收容IO设备的数据
 - 输出井用来模拟输出时的磁盘
- 输入缓冲区和输出缓冲区
 - 输入缓冲区：暂存由输入设备送来的数据
 - 输出缓冲区：暂存从输出井送来的设备
- 输入进程和输出进程
 - 输入进程：模拟脱机输入时的外围控制机，将用户要求的数据从输入机通过输入缓冲区送到输入井中，当CPU需要数据，直接将输出井中的数据送入内存
 - 输出进程：模拟脱机输出时的外围控制机，把用户要求输出的数据先从内存送到输出井中，待输出设备空闲时，再将输出井中的数据经过输出缓冲区送到输出设备
- 特点
 - 提高了IO速度
 - 独占设备变成了共享设备
 - 实现了虚拟设备功能
- 通俗一点就是，如果设备被占用，我们就先把数据暂存一下，等到设备空闲了就把这些数据输送到设备中

高速缓存与缓冲区对比

相同点

都介于高速设备和低速设备之间

不同

存放数据

高速缓存：存放的是低速设备上的某些数据的复制数据

缓冲区：存放的是低速设备传递给高速设备的数据，这些数据在低速设备上不一定有备份，这些数据再从缓冲区传送到高速设备

目的

高速缓存：高速缓存存放的是高速设备经常要访问的数据，如高速缓存中数据不在，高速设备就要访问低速设备

高速设备和低速设备的通信都要经过缓冲区，高速设备永远不会去直接访问低速设备

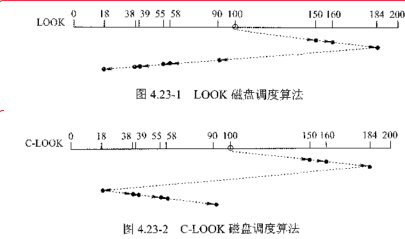
4.3 磁盘组织与管理

磁盘结构

- 表面涂有磁性物质的金属或塑料构成的圆形盘片，通过一个称为磁头的导体线圈从磁盘读取数据。
- 磁盘盘面上的数据存储在在一组同心圆中，称为磁道
- 一个盘面上有上千个磁道，磁道又划分为几百个扇区，每个扇区固定存储大小，一份扇区称为一个盘块
- 磁盘地址用 柱面号-盘面号-扇区号(块号) 表示
- 磁盘分类
 - 固定头磁盘：磁头相对于盘片的径向方向固定
 - 活动头磁盘：每个磁道一个磁头，磁头可以移动
 - 固定盘磁盘：磁头臂可以来回伸缩定位磁道，磁盘永久固定在磁盘驱动器内
 - 可换盘磁盘：可以移动和替换

磁盘调度算法

- 读写时间组成
 - 寻找时间：活动头磁盘在读写信息前，将磁头移动到指定磁道所需要的时间（跨越n条磁道+启动磁臂）： $T_s = m \cdot n + s$
 - 延迟时间：磁头定位到某一磁道扇区所需要的时间， $T_r = 1/2r$ （转速为r）
 - 传输时间：从磁盘读出或向磁盘写入数据经过时间， $T_r = b / rN$
- 磁盘调度算法
 - 先来先服务算法（FCFS）
 - 按照进程请求访问磁盘的先后顺序进行调度
 - 优点：公平 实现简单
 - 缺点：适用于少量进程访问，如果进程过多算法更倾向于随机调度
 - 最短寻找时间有限算法（SSTF）
 - 选择调度处理的磁道是与当前磁头所在磁道距离最近的磁道
 - 优点：性能强于先来先服务算法
 - 缺点：容易产生饥饿现象
 - 扫描(SCAN)算法
 - 在磁头当前移动方向上选择与当前磁头所在的磁道距离最近的请求作为下一次服务对象
 - 优点：寻道性能好，可以避免饥饿现象
 - 缺点：对最近扫描过的区域不公平，访问局部性方面不如FCFS和SSTF好
 - 循环扫描算法（C-SCAN）
 - 磁头单向移动，回返时直接回到起始端，而不服任何请求
 - LOOK与C-LOOK算法
 - 在SCAN与C-SCAN算法的基础上规定了查看移动方向上是否有请求，如果没有就不会继续向前移动，而是直接改变方向（LOOK）或者直接回到第一个请求处（C-LOOK）
- 对盘面交替编号 原因：磁头在读/写一个物理块后，需要经过短暂的处理时间才能开始处理下一块



磁盘的管理

- 磁盘初始化
 - 低级格式化：磁盘分扇区，为每个扇区采用特别的数据结构（头、数据区域、尾部组成），头部含有一些磁盘控制器所使用的信息
 - 进一步格式化处理：磁盘分区，对物理分区进行逻辑格式化（创建文件管理系统），包括空闲和已分配的空间及一个初始为空的目录
- 引导块
 - 计算机启动时运行自举程序，初始化CPU寄存器、设备控制器和内存等，然后启动操作系统
 - 组局程序通常保存在ROM中，在ROM中保留很小的自举块，完整的自举程序保存在启动块上
 - 拥有启动分区的磁盘称为启动磁盘或系统磁盘
- 坏块
 - 无法使用的扇区
 - 对于简单的磁盘，可以在逻辑格式化时（建立文件系统时）对整个磁盘进行坏块检查，标明哪些扇区是坏扇区，比如：在 FAT 表上标明
- 处理方式
 - 简单磁盘：手动处理，对坏块进行标记，程序不会使用
 - 复杂磁盘：控制器维护一个磁盘坏块链表，同时将一些块作为备用，用于替代坏块（扇区备用）