

I/O管理

IO核心子系统

- I/O子系统概述
- 组成 - 设备独立性软件、设备驱动程序、中断处理程序
  - 提供的服务功能 - I/O调度、缓冲与高速缓存、设备分配与回收、假脱机、设备保护和差错处理等
- I/O调度概念
- 引入目的 - 应用程序所发布的系统调用的顺序不一定总是最佳选择，所以需要I/O调度来改善系统整体性能，使得进程之间公平地共享设备访问，减少I/O完成所需要的平均等待时间
  - 定义 - I/O调度就是确定一个好的顺序来执行这些I/O请求
  - 细节补充 - 磁盘调度就是I/O调度的一部分
  - 用“请求队列”、调度算法来实现调度

操作系统内核、IO核心子系统

类似进程调度、磁盘调度

- 引入缓冲区的目的
- 1. 缓解CPU与设备的速度矛盾
  - 2. 减少CPU的中断频率
  - 3. 解决数据粒度(基本数据单元大小)不匹配的问题
  - 4. 提高CPU与I/O设备之间的并行性

缓冲区的实现 - 利用内存的一部分作为缓冲区

- 缓冲区的特点
- 当缓冲区的数据非空时，不能往缓冲区写入数据，只能从缓冲区传出数据
  - 当缓冲区为空时，可以往缓冲区冲入数据，但必须把缓冲区充满后，才能读出数据

类似“管道”，单向传递  
要实现双向，必须用两个管道

- 缓冲技术分类(根据缓冲器的个数)
- 单缓冲 - 设备——T——缓冲区——M——工作区——C处理  
处理一块数据的平均耗时为： $\text{Max}(C, T) + M$   
分析问题的初始状态：假设工作区满，缓冲区空
  - 双缓冲 - 处理一块数据的平均耗时为： $\text{Max}(T, C) + M$   
分析问题的初始状态：假设工作区空，一个缓冲区满，另一个缓冲区空
  - 循环缓冲 - 多个缓冲区链接成循环队列，in指针指向第一个空缓冲区，out指针指向第一个满缓冲区
  - 缓冲池 - 三个队列：空缓冲区队列、输入队列、输出队列
  - 四个工作缓冲区 - 用于收容输入数据的工作缓冲区、用于提取输入数据的工作缓冲区  
用于收容输出数据的工作缓冲区、用于提取输出数据的工作缓冲区

高速缓存是可以保存数据拷贝的高速存储器，访问高速缓存比访问原始数据更高效，速度更快。高速缓存和缓冲区的对比见表5.1。

表 5.1 高速缓存和缓冲区的对比

		高 速 缓 存	缓 冲 区
区 别	相 同 点	都介于高速设备和低速设备之间	
	存放数据	存放的是低速设备上的某些数据的复制数据，即高速缓存上有的，低速设备上必然有	存放的是低速设备传送给高速设备的数据（或相反），而这些数据在低速设备（或高速设备）上即不一定有备份，这些数据再从缓冲区传送到高速设备（或低速设备）
	目的	高速缓存存放的是高速设备经常需要访问的数据，若高速设备要访问的数据不在高速缓存中，则高速设备就需要访问低速设备	高速设备和低速设备的通信都要经过缓冲区，高速设备永远不会直接去访问低速设备

高速缓存与缓冲区的对比

- 设备分配与回收应考虑的因素
- 独占设备 - 一个进程申请使用设备后，不再允许其它进程申请，要一直等到该设备释放，如打印机。（会死锁）
  - 设备的固有属性 - （分时）共享设备 - 一段时间内允许许多设备分时共享。（不会死锁）
  - 虚拟设备(SPOOLing) - 以空间换时间，对I/O操作进行批处理
  - 设备分配算法 - 先来先服务、优先级、短任务优先等
  - 设备分配的安全性 - 安全分配方式  
不安全分配方式

- 设备分配的策略
- 静态分配 - 进程运行前为其分配全部所需资源，运行结束后才归还资源  
优缺点 - 不会死锁，效率高
  - 动态分配 - 进行运行的过程中动态申请需要的设备资源  
优缺点 - 分配算法使用不当导致分配欠性不当，可能造成进程死锁；动态申请和释放有利于提高设备的利用率
  - 补充 - 独占设备两者都可，但大多采用静态分配  
共享设备一般采用动态分配方式，但每个I/O传输的单位时间内只被一个进程占有，通常采用先来先分配、优先级高优先的分配算法

- 设备分配的数据结构
- 系统设备表(SDT) - 用于记录整个系统中所有设备的使用情况，每个设备对应一个表目  
一个系统设备表  
包含所有设备使用情况
  - 设备控制表(DCT) - 关键字段：设备类型/标识符/DCT/驱动程序  
每个设备对应一张DCT
  - 一个控制器控制多个设备  
控制器控制表(COCT) - 关键字段：类型/标识符/状态/指向COCT的指针/等待设备队列指针  
每个控制器对应一张COCT
  - 一个通道控制多个控制器  
通道控制表(CHCT) - 关键字段：状态/指向CHCT的指针/等待控制器的队列指针  
每个控制器对应一张CHCT
  - 关键字段：状态/与通道连接的控制器表集合的首项地址/等待通道的队列指针

- 设备分配的步骤
- 步骤 - 1. 根据进程请求的物理设备名查找系统表SDT  
2. 根据系统表SDT找到DCT该设备的控制表，并分配设备  
3. 根据DCT设备控制表，找到COCT并分配控制器  
4. 根据COCT找到CHCT并分配通道
  - 缺点 - 用户编程时必须用“物理设备名”，若除了一个物理设备，则程序无法运行，若进程请求的这个物理设备忙，则即使系统中还有同类型的设备，进程也必须阻塞等待当前请求的设备

- 补充：逻辑设备到物理设备的映射
- 引入目的 - 改进设备分配步骤；用户编程时使用逻辑设备名(设备类型)来申请设备，操作系统实现从逻辑设备到物理设备名的映射(通过LUT)
  - 逻辑设备表(LUT)的设置 - 整个系统只有一张LUT；各用户所用的逻辑设备名不允许重复  
每个用户一张LUT；各个用户的逻辑设备名可以重复

①根据进程请求表的物理设备名查找SDT (LUT：用户编程时使用的逻辑设备名其本质是“设备类型”) ②查SDT，找到用户进程用设备名，并且空闲的设备，将其分配给该进程。操作系统在逻辑设备表(LUT)中构造一个表目 ③根据DCT找到COCT，若进程忙，则将该进程PCB挂到控制阻塞队列中，不忙则将该控制表分配给该进程 ④根据COCT找到CHCT，若进程忙，则将该进程PCB挂到通道阻塞队列中，不忙则将该通道分配给该进程

物理设备表 (SDT)	逻辑设备表 (LUT)	逻辑设备名	物理设备名	驱动程序入口地址
表目1	表目1	设备类型	设备标识符	设备标识符
表目2	表目2	设备类型	DCT (设备控制表)	驱动程序入口
...	...	...	...	...

改进后的设备分配步骤

- SPOOLing假脱机技术
- 假脱机技术 - 使用控制机，更高速的设备完成。如：磁带
  - 作用 - 缓解设备与CPU的速度矛盾，实现预输入、缓输出
  - 用软件的方式模拟假脱机技术
  - 组成 - 输入井和输出井 - 模拟假脱机输入/输出时的磁带  
输入进程和输出进程 - 模拟假脱机输入/输出时的外围控制机  
输入缓冲区和输出缓冲区 - 内存中的缓冲区，输入、输出时的暂存“中转站”
  - 提高了I/O速度
  - 特点 - 将独占式设备改造造成共享设备  
实现了虚拟设备的功能(缓冲区暂存数据)
  - 例子 - 共享打印机。使用SPOOLing技术将独占式打印机虚拟成共享打印机，同一时间允许许多个进程访问(缓冲区暂存数据)，但同一时刻还是只允许一个