

# I/O管理和磁盘调度

## I/O设备的分类

---

- 交互方式
  - 人可读的设备
  - 机器可读的设备
  - 通信设备
- 设备管理
  - 字符设备：以字符流的形式传递数据
  - 块设备：磁盘
  - 网络设备：根据通信协议，以数据块为单位交换
- 处理器与I/O设备的连接
  - **设备控制器（硬件）**
  - **驱动程序（软件）**

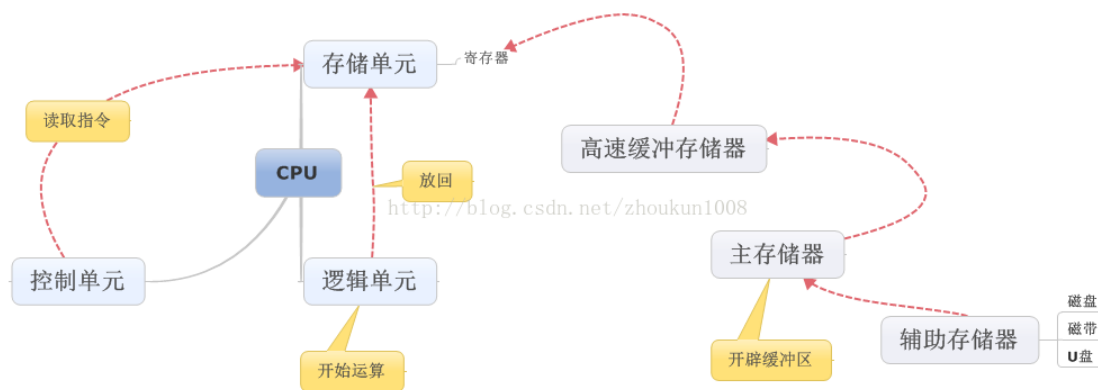
## 驱动程序

---

- 概念：驱使硬件设备工作的**软件**
- 提供了操作硬件设备的接口（API/函数）
- 与设备控制器的关系
  - 设备控制器是一个硬件设备，相当于一个小型处理器，而驱动程序相当于编译器，将用户I/O指令转换为设备控制器可以读懂的指令，也就是I/O进程与设备控制器之间的通信程序
  - 也就是设备控制器+驱动程序合起来操作了I/O设备
  - [设备驱动程序参考讲解](#)
  - [设备控制器参考讲解](#)
- 设备驱动程序的处理过程
  - 接到抽象的要求转换为具体要求
  - 检查用户I/O请求的合法性：是否有权限等
  - 读出和检查设备的状态
  - 传送必要的参数（在读写前，传递参数到设备控制器的寄存器中）
  - 完成I/O操作
- 设备驱动与软硬件的关系
  - 无操作系统：应用软件直接调用驱动程序提供的接口，操纵硬件，例：单片机，STM32等
  - 有操作系统：操作系统进一步封装驱动程序提供的接口，向应用程序提供统一的接口操纵硬件

## 前引

---



- 从上面可以看到，I/O操作就是不停地数据读取，辅存读到主存，(主存读到cache)，cache读到处理器的寄存器中（主存读到处理器的寄存器中）
- 那么我们可以考虑几个问题
  - 如何在辅存中更快地找到数据（辅存的速度是上述涉及到存储单元中最慢的）---》磁盘调度
  - 每一次的I/O都会等很长时间（相比于处理器的速度），能不能利用这段时间？
    - 边读边用？ ---》I/O缓冲

## I/O缓冲

- 依赖于DMA
- 思想
  - 以往通常是进程因为I/O阻塞，等到I/O操作全部完成（辅存中的数据全部读入主存中）之后再唤醒进程
  - DMA 进行I/O不需要CPU的参与，那么读取数据与数据计算同时进行，也就是处理器在进行计算的同时，并行地进行DMA I/O数据读取
  - 在主存中开辟缓冲，当处理器在进行上一部分数据计算的同时，DMA往缓冲中进行下一部分数据的读写
- 单缓冲：主存中只设置一个缓冲区，这样当处理器在从缓冲中读数据时，不能再向缓冲区中读取数据
- 双缓冲：主存中设置两个缓冲区，这样当处理器在从一个缓冲中读取数据时，可以向另一个缓冲区中读取数据
- 多缓冲：主存中设置多个缓冲区

## 磁盘调度策略

- 设计磁盘调度策略的目的是减少寻道时间，更快地找到数据
- 效率最低的磁盘调度是随机调度（每次寻道之间没有关联）
- 接近于随机调度的磁盘调度策略
  - 先进先出：顺序处理进程的磁盘请求
  - 后进先出
  - 优先级

- 上述三种策略的特点是：没有考虑每次寻道之间位置的关系，性能接近于随机调度，仅仅是从用户角度，**考虑进程的执行顺序**
- 考虑减少寻道时间的磁盘调度策略
  - **最短服务时间优先**
    - 在进程队列中寻找最短寻道时间的进程
  - **扫描 (SCAN)**
    - 沿一个方向扫描，当该方向没有其他进程的I/O请求或者已经到了最后一个磁道
    - 接着从现在的位置反向扫描
  - **C-SCAN**
    - 沿一个方向扫描，当该方向没有其他进程的I/O请求或者已经到了最后一个磁道
    - 再回到磁盘的相反方向磁道的末端重沿该方向新扫描
  - **N-step-SCAN**
    - 磁盘请求进程队列分为长度为N的多个子队列
    - 采用SCAN，一次处理一个队列
    - 新磁盘请求进程进入未处理的队列
  - **FSCAN**
    - 磁盘请求队列分为两个子队列，一个存放已经到达的队列（旧队列），一个为空（新队列）
    - 现在执行的进程存放于旧队列中，旧队列执行完后，新队列转为旧队列
    - 新磁盘请求到达后进入新队列

## RAID

---

- 磁盘条带化
  - 将数据按照一定大小分成多个数据块，可分别存放于多个不同的物理盘上
  - 系统在特定的物理盘读取数据时，可以提前通知下一个目标盘准备数据
  -
- 磁盘镜像
  - 将相同的数据存入多个物理盘中
  - 冗余备份---》提升可靠性
  -
- [RAID wiki](#)