I/O管理

I/O管理要完成哪些功能?

状态跟踪	要能实时掌握外部设备的状态
设备存取	要实现对设备的存取操作
设备分配	在多用户环境下，负责设备的分配与回收
设备控制	包括设备的驱动，完成和故障的中断处理

I/O软件层次结构

四种结构（常考点）

	定义	举例
1.用户层 I/O软件	• 用户通过统一的接口发送命令	• 如发送read命令 • 如讲二进制整数转换为ascii码的格式打印
2.设备独立性软件	• 对用户的命令进行解析，把逻辑信息转换为物理信息，例如查FAT表或者inode信息，但不是更具体的物理信息	• 如解析read命令 • 如检查用户是否有权使用设备（即设备保护） • 如进行缓冲管理
3.设备驱动程序	• 负责执行OS发出的I/O命令 • 针对不同的硬件将命令解析为指令 • 若更换物理设备，只需要修改设备驱动程序，不需要修改应用程序	• 如将解析好的read命令转换为指令 • 如计算磁盘的柱面号，磁头号，扇区号 • 如设备寄存器写命令 • 如将逻辑块号转换为物理地址
4.中断处理程序	• 中断正在运行的进程，转而执行用户命令	• 如中断当前进程，执行相关指令

层次结构的特点

- 每层都是利用其下层提供的服务，完成输入/输出功能中的某些子功能
- 屏蔽子功能实现的细节，向高层提供服务
- 仅最底层才涉及硬件的具体特性

应用程序I/O接口分类

字符设备接口	• 字符设备是指数据的存取和传输都是以字节为单位的设备 • 字符设备都属于独占设备
块设备接口	• 块设备是指数据的存取和传输都是以数据块为单位的设备，如磁盘 • 磁盘设备的I/O常使用DMA方式
网络设备接口	• 许多OS提供的网络I/O接口为网络套接字接口
阻塞/非阻塞 I/O	• 大多数OS提供的 I/O接口都是采用阻塞 I/O

设备的分类

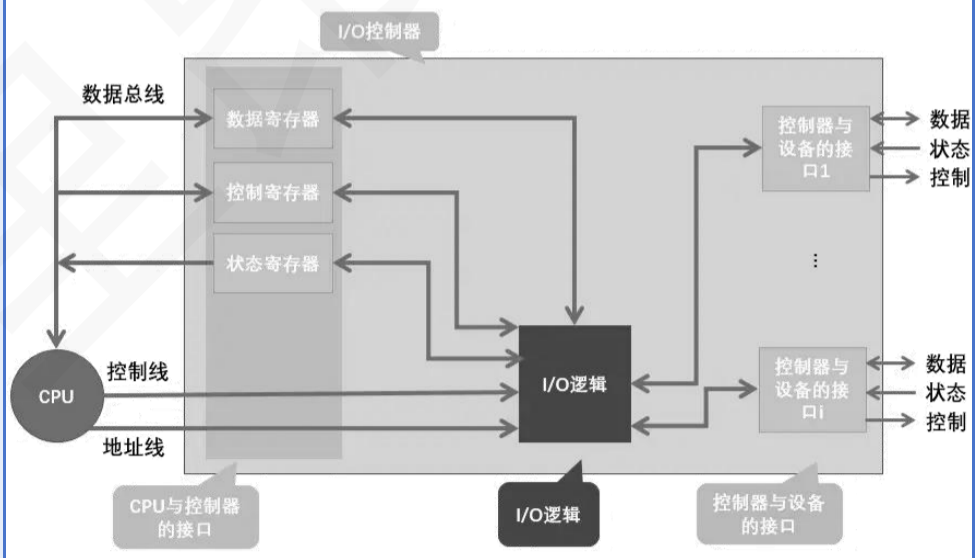
按信息交换的单位分类	**块设备**	• 传输速度较高+可寻址+可随机读/写任一块 • 属于有结构设备；如磁盘，硬盘，USB
	字符设备	• 传输速率低+不可寻址+时常采用中断I/O方式 • 属于无结构类型；如交互式终端机，打印机，鼠标
按传输速度分类	**低速设备**	键盘，鼠标
	中速设备	激光打印机
	高速设备	磁盘机，光盘机
按设备特性分类	**独占设备**	• 一个时段只能分配给一个进程 • 所有字符设备都是独占设备 • 速度慢，利用率低 • 如输入机、打印机、磁带机等
	共享设备	• 一段时间内允许多个进程同时访问的设备 • 共享设备必须是可寻址和可随机访问的设备 • 软硬盘、磁盘、光盘等块设备都是共享设备
	虚拟设备	• 通过虚拟软件技术将独占设备改造成共享设备 • 如通过 SPOOLing技术将一台打印机虚拟成多台打印机 • 其实质上还是独占设备
按访问顺序	**可以顺序访问的**	磁带
	可以随机访问的（直接访问）	光盘，磁盘，U盘
	两者都可以的	光盘，磁盘，磁盘

设备控制器(I/O接口)

1.设备控制器主要功能

- 接受和识别CPU发来的命令
- 数据交换（设备和控制器的数据交换+控制器和主存数据交换）
- 标识和报告设备的状态，以供CPU处理
- 地址识别；数据缓冲；差错控制
- 设备控制器不属于操作系统范畴，它是属于硬件

2.设备控制器的组成

1.设备控制器与CPU的接口	• 用于实现CPU与控制器之间的通信 • CPU通过控制线发出命令 • 通过地址线指明要操作的设备 • 通过数据线来输入输出数据
2.设备控制器与设备的接口	• 用于实现控制器和设备之间的通信
3.I/O逻辑	• I/O逻辑用于负责接收和识别CPU的各种命令（如地址译码） • 根据CPU的命令对相应的设备发送命令 • 用于实现设备控制功能
组成图	

3.设备控制器有三种寄存器
(设备控制器中可被CPU直接访问的寄存器，也叫I/O端口)

数据寄存器	CPU 向 I/O 设备写入需要传输的数据
命令/控制寄存器	• 比如要打印的内容是「Hello」，CPU 就要先发送一个 H 字符给到对应的 I/O 设备 • CPU 发送一个命令，告诉 I/O 设备，要进行输入/输出操作，于是就会交给 I/O 设备去工作 • 任务完成后，会把状态寄存器里面的状态标记为完成
状态寄存器	• 目的是告诉 CPU 现在已经在工作或工作已经完成 • 如果已在工作状态，CPU 再发送数据或者命令过来，都是没有用的 • 直到前面的工作已经完成，状态寄存标记成已完成，CPU 才能发送下一个字符和命令

4.CPU与I/O端口的通信方式

端口 I/O (独立编制)	• 每个控制寄存器被分配一个 I/O 端口 • 可以通过特殊的汇编指令操作这些寄存器 • 比如 in/out 类似的指令
内存映射 I/O (统一编制)	• 将所有控制寄存器映射到内存空间中 • 这样就可以像读写内存一样读写数据缓冲区

设备控制方式（I/O控制方式）

- 外围设备和内存之间的控制方式
- 用于控制设备和内存或CPU之间的数据传送

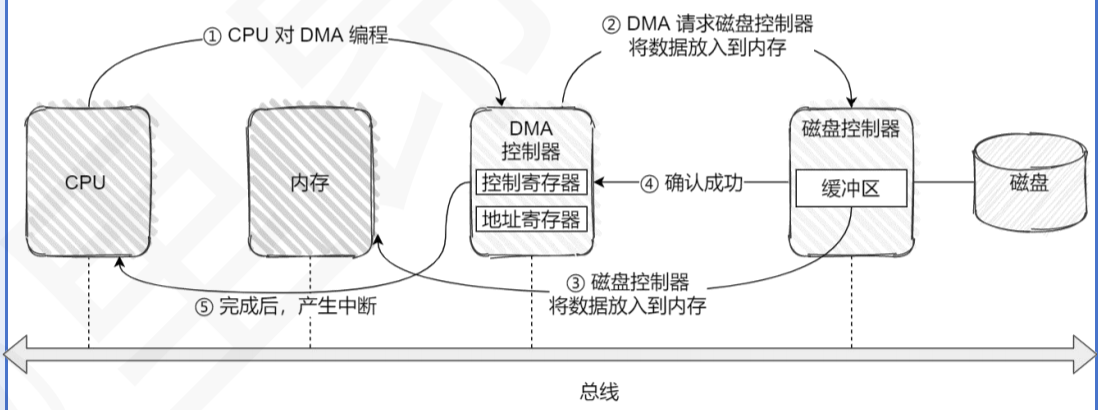
1.程序直接控制方式

- 一种轮待询等的方法，让 CPU 一直查寄存器的状态，直到状态标记为完成

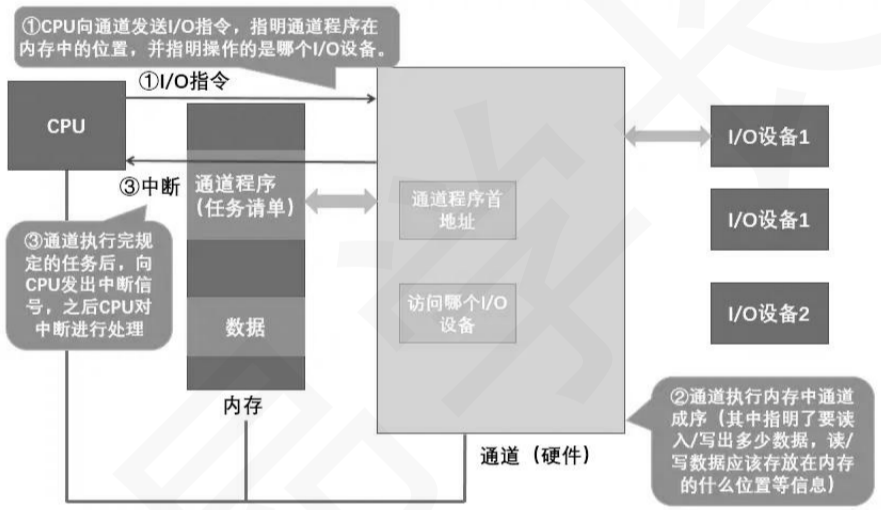
2.中断驱动程序

- 有一个硬件的中断控制器，当设备完成任务后触发中断到中断控制器，中断控制器就通知 CPU，例如键盘
- 一个中断产生了，CPU 需要停下当前手里的事情来处理中断
  - 软中断，例如代码调用 INT 指令触发
  - 硬件中断，就是硬件通过中断控制器触发的
- 中断的方式对于频繁读写数据的磁盘，并不友好，这样 CPU 容易经常被打断，会占用 CPU 大量的时间，使用DMA解决

3.DMA方式（Direct Memory Access）

概念	• 可以使得设备在 CPU 不参与的情况下，能够自行完成把设备 I/O 数据放入到内存 • 要实现 DMA 功能要有 DMA 控制器硬件的支持
工作方式	• CPU 需对 DMA 控制器下发指令，告诉它想读取多少数据，读完的数据放在内存的某个地方就可以了； • 接下来，DMA 控制器会向磁盘控制器发出指令，通知它从磁盘读数据到其内部的缓冲区中，接着磁盘控制器将缓冲区的数据传输到内存； • 当磁盘控制器把数据传输到内存的操作完成后，磁盘控制器在总线上发出一个确认成功的信号到 DMA 控制器； • DMA 控制器收到信号后，DMA 控制器发中断通知 CPU 指令完成，CPU 就可以直接用内存里面现成的数据了
工作流程	1. 初始化DMA控制器并启动磁盘 2. 从磁盘传输一块数据到内存缓冲区 3. DMA控制器发送中断请求 4. 执行DMA结束中断服务程序
特性	• 整个过程仅仅在开始和结束时需要CPU干预 • 每个DMA控制器对应一台设备与内存传递数据 • DMA方式主要用于块设备，磁盘是典型的块设备
示例图	

4.通道控制方式

通道方式过程	• 设置通道后，CPU只需向通道发送一条I/O指令，通道在收到该指令后，便从内存中取出本次要执行的通道程序，然后执行该通道程序 • 仅当通道完成规定的I/O任务后，才向CPU发出中断信号
特性	• CPU、通道、I/O设备可并行工作，资源利用率很高 • 实现复杂，需要专门的硬件支持 • 一个通道可以控制多台设备与内存的数据交换
分类	• 字节多路通道：通常包含许多非分配型字通道，数量可以达到几十到几百个，每个通道连接一台IO设备，并控制该设备的IO操作 • 字节多路通道常用于连接大量的低速或中速I/O设备
示例图	

四种方式的比较

I/O控制方式	过程	CPU干预频率	每次传输数据的单位	数据流向	优缺点
程序直接控制方式	CPU发出指令后需要不断轮询。	极高	字	设备-CPU-内存 内存-CPU-设备	每一个阶段的优点都是解决上一个阶段的最大缺点。总体来说就是尽量减少CPU对I/O过程的干预，把CPU从繁杂的I/O控制事务中解脱出来，以便更多地去完成其他任务。
中断驱动方式	CPU发出I/O指令后可以做其他事，本次I/O完成后设备控制器发出中断信号。	高	字	设备-CPU-内存 内存-CPU-设备	
DMA方式	CPU发出I/O命令后可以做其他事，本次I/O完成后DMA控制器发出中断信号。	中	块	设备-内存 内存-设备	
通道控制方式	CPU发出I/O指令后可以做其他事。通道会执行通道程序以完成I/O，完成后通道向CPU发出中断信号。	低	一组块	设备-内存 内存-设备	

设备分配

- 设备分配指根据用户的I/O请求分配所需的设备
- 计算机系统为每台设备确定一个设备的绝对号以便区分和识别设备

设备分配的策略

分配的原则	• 设备固有属性决定了设备的使用方式（充分发挥设备的使用效率，尽可能让设备忙碌） • 设备独立性可以提高设备分配的灵活性和设备的利用率（设备独立性是指用户使用设备的透明性，即用户程序与实际使用的物理设备无关） • 设备安全性可以保证分配设备时不会导致永久阻塞（要避免造成进程死锁）
设备分配的方式	• 静态分配→对独占设备的分配→在作业执行前分配→不会出现死锁→设备的使用率低 • 动态分配→在进程执行中分配，通过系统调用发送请求，根据具体策略分配设备→分配算法不好时会出现死锁→设备利用率高
设备分配算法	先请求先分配，优先级高者分配（类比调度算法）

设备分配的安全性

定义	• 设备分配安全性是指设备分配中应防止发生进程死锁
安全分配方式	• 为进程分配一个设备后就将该进程阻塞，本次I/O完成后才将进程唤醒 • 安全分配方式在一个时间段内每个进程只能使用一个设备。 • 优点：破坏了请求等待条件，不会死锁。 • 缺点：对于一个进程来说，CPU和I/O设备只能串行工作，系统资源利用率低。
不安全分配方式	• 进程发出I/O请求后，系统为其分配I/O设备，进程可继续执行，之后还可以发出新的I/O请求，只有某个I/O请求得不到满足时才将进程阻塞 • 不安全分配方式一个进程可以同时使用多个设备 • 优点：进程的计算任务和I/O任务可以并行处理，使进程推进。 • 缺点：有可能发生死锁。

逻辑设备名到物理设备名的映射

目的	• 为了提高设备分配的灵活性和利用率→分别实现I/O重定向→所有引入了设备独立性
定义	• 设备独立性：指应用程序独立于具体使用的物理设备/用户在编程序时使用的设备与实际设备无关 • 逻辑设备表LUT：将逻辑设备名映射为物理设备名 • LUT表项包括：逻辑设备名，物理设备名，设备驱动程序入口地址
分类	• 两种方式设置逻辑设备表 • 1.整个系统只设置一张LUT→适用于单用户系统 • 2.每个用户设置一张LUT→用户登陆时，系统为用户建立一个进程，同时建立一张LUT
好处	• 方便用户编程 • 使程序运行不受具体机器环境的限制 • 便于程序移植