

IO设备管理

IO是输入输出，IO设备可以将数据输入到计算机、或从计算机输出数据到IO设备

IO设备分类

- 按传输速率分类
 - 低速设备：几字节到几百字节每秒，鼠标、键盘、语音输入输出等
 - 中速设备：千个到数万个字节每秒，打印机等
 - 高速设备：数百千到数十兆字节每秒，光盘、磁带等
- 按信息交换单位分类
 - 块设备：用于存储信息且信息存取总以数据块为单位，典型的块设备是磁盘
 - 字符设备：用于数据输入输出，基本单位是字符
- 按使用方式分类
 - 独占设备：一段时间内只允许一个用户/进程访问的设备，例如打印机这一临界资源必须互斥访问的设备
 - 共享设备：一段时间内允许多个进程同时访问，例如磁盘为可寻址、可随机访问的设备
 - 虚拟设备：通过虚拟技术会将一台独占设备变为若干台逻辑设备

IO控制器

分为机械部件和电子部件。其中电子部件就是常说的IO控制器（接口、控制接口、模块）

IO控制器功能

- 数据缓冲：主存和CPU寄存器的速度很快，外设速度较低。在设备控制器汇总引入数据缓冲器用于缓存数据
- 错误和就绪检测：提供错误和就绪时需检测逻辑，并将结果保存在状态寄存器供CPU查用
- 控制和定时：接收和识别CPU或通道发来的控制信息和定时信号，根据相应信息和信号向外设发送控制信号
- 数据格式的转换：提供数据格式转换部件，通过外部接口得到的数据转换为内部接口需要的格式
- 当链接多台设备时，设别地址识别

IO端口

CPU能够访问的各类寄存器称为IO端口，驱动程序通过IO端口控制外设进行IO。对IO端口读写就是向IO设备送出命令或从设备读状态或读写数据。

- 控制端口（命令端口）：存放CPU通过out指令向外发送的控制命令
- 状态端口：CPU通过in指令读取状态寄存器了解外设和设备控制器的状态
- 数据端口：访问数据缓冲寄存器进行数据的IO

一个IO控制器会有多个端口地址，IO端口编号后才能被CPU访问，IO设备的寻址方式就是IO端口的编号方式：

1. 独立编址方式（IO映射方式）：外设与主存单元的地址分别单独编址，外设端口不占用主存空间。但CPU要专门设置输入IO指令访问端口，增加了控制复杂性
2. 同一编址方式：外设端口和主存单元的地址统一编址，区分靠不同的地址码，不需要给IO端口额外的编址。但占用了存储器地址空间，执行速度较慢

IO控制方式

CPU与IO设别实现数据交换的方式。CPU和外设之间交换数据，实质上是通过IO端口进行。

- 程序查询方式：完全通过CPU执行IO程序控制主机和外围设备间的信息交换
 - 需要查询指令（查询设备状态）、传送指令（设备就绪时执行数据交换）、转移指令（数据未就绪时转向查询指令继续查询）
 - 不断查询IO设备情况，终止源程序执行，完全处在串行工作状态
- 程序中断方式：允许IO设备主动报告自己的状态，不需要CPU不断查询
- DMA方式（直接存储器存取）：在外围设备和主存间开辟直接的数据传送通路
 - 需要寄存器：命令/状态寄存器CR、内存地址寄存器MAR、数据寄存器DR、数据计数器DC
 - 需要DMA中断机构，当一个数据块传输完成后触发中断机构向CPU提出中断请求
 - 由于是异步方式工作，数据块传输结束的时机必须由DMA中断机构以中断方式向CPU告知
- IO通道控制方式：一种特殊的具有执行IO指令能力并通过执行通道IO程序来控制IO操作的处理机
 - 指令类型单一，通道硬件比较简单，其执行的命令主要局限于与IO相关的操作操作有关的指令
 - 通道与CPU共享内存
 - 主要是工作站等专业设备使用
- IO处理机：可以不通过CPU来独立控制IO设备

IO软件

IO软件的设计目的和原则：与具体设备无关，统一命名，对错误处理，缓冲技术等

IO软件层次结构

（驱动程序直接将请求信息发给硬件，硬件发送应答给中断处理程序，再发给驱动程序）：

- 用户进程
- ↓IO请求 || ↑IO应答
- ↓用户层软件↑ ← 交互接口，调用函数库
- ↓设备独立性软件↑ ← 设备的命名、保护、分配、回收
- ↓设备驱动程序↑（可重入）← IO命令转为具体要求，DC和软件的接口
- | | 中断处理程序↑ ← IO操作结束被唤醒，处理中断
- 硬件↑ ← 执行IO操作

层次	作用
用户层软件	实现与用户交互的接口，用户可直接调用再用户层提供的与IO相关操作的库函数，对设备进行操作
设备独立性软件	（设备无关软件）实现设备独立性，让应用程序独立于具体物理设备，使用逻辑设备名来请求某类设别（再系统实际执行时还需使用跟物理设备名称）
设备驱动程序	IO进程和设别控制器之间的通信程序。接收上层软件发来的抽象IO要求，转换为具体要求后发给设备控制器启动设备并执行。为每一类设备设置一个进程，专门用于执行这类设备的IO操作

层次	作用
中断处理程序	进行进程上下文切换，对处理中断信号源进行测试，读取设备状态和修改进程状态

IO缓冲处理

CPU实现与IO设备进行数据交换的方式。CPU和外设实质上通过IO端口进行交换数据：CPU <----> IO接口/端口 <----> IO设备。

IO缓冲的目的：减少CPU被中断的频率，放宽CPU的中断响应时间限制，提高CPU和IO设备间的并行性，缓解CPU和IO设备的速度矛盾：
CPU <----> 缓冲区 <----> IO。

一般多使用内存作为缓冲区。

单缓冲：只设置一个缓冲区，用户之间发数据只能以“半双工”形式；
双缓冲：设置两个缓冲区，可以同时输入或输出，也可以使得用户间可以“全双工”形式。

T：IO设备将数据输入单缓冲的时间；
M：OS将缓冲数据传至内存用户区的时间；
C：CPU对数据的处理时间；
总处理时间 = M + max(C+M,T)。//对于多块设备需要额外加上C；
若C+M = T，则双缓冲区构成二集流水线结构

设备的分配、回收

- 分配策略以及考虑因素
 - 静态分配：一次性分配进程所需全部设备
 - 动态分配：进程执行过程中根据执行需要按策略分配
 - 设备分配算法：
 - 先请求先分配
 - 优先级高者优先
 - 设备固有属性：不同属性采取不同的分配策略
 - 独占设备：得到设备的进程独占，直到完成或其他原因释放设备
 - 共享设备：分配给多个进程，并合理分配各进程的访问顺序
 - 虚拟设备：共享设备，可将其同时分配多个进程
 - 设备回收：释放设备、设备控制器、通道；修改相应数据结构
 - 安全性：
 - 安全分配：进程发出IO请求后进入阻塞，串行工作
 - 不安全分配：进程发出IO请求后仍可运行，并行工作
- 逻辑设备名到物理设备名的映射

设备分配中的数据结构：

- 设备控制表DCT：系统为每个设备配一个DCT，记录状态
- 控制器控制表COCT：系统为每个控制器配一个COCT
- 通道控制表CHCT：系统为每个通道配一个CHCT
- 系统设备表SDT：记录系统全部设备情况的表，每个设备占一个表目

- 逻辑设备表LUT：存放逻辑设备名、物理设备名、设备驱动程序入口地址的表，用于完成逻辑设备到物理设备的转换

假脱机SPOOLing

缓和CPU和IO设备的高低速矛盾而引入。当系统引入多道程序技术后，完全可以利用其中两道程序模拟脱机输入和输出的外围控制机功能，实现低速设备和高速设备的脱机输入、输出。

SPOOLing技术把一台物理设备虚拟成逻辑上的多台设备，将独占式设备改为共享设备。提高了IO速度，将独占设备改造为共享设备，实现了虚拟设备功能。

补充

- 区分硬件和识别设备的代号称为设备的绝对号（硬件角度）
- 对多个同一类硬件（例如多台打印机）只需提供一个设备驱动程序
- 将占10个硬盘块的问题逐个读入主存缓冲区并送入用户区分析，磁盘块读入缓冲区耗时100us，缓冲区数据传至用户区50us，CPU分析数据50us
 - 单缓冲区结构： $10 \times (100 + 50) + 50 = 1500$
 - 双缓冲区结构： $10 \times (100) + 100 = 1100$
- SPOOLing提高了单机资源利用率