

# I/O中断处理程序

- 位于操作系统底层，与硬件设备密切相关，与系统其余部分尽可能少地发生联系
- 进程请求I/O操作时，通常被挂起，直到数据传输结束后并产生I/O中断时，操作系统接管CPU后转向中断处理程序
- 当设备向CPU提出中断请求时，CPU响应请求并转入中断处理程序

# I/O中断处理程序的功能

- 检查设备状态寄存器内容，判断产生中断的原因，根据I/O操作的完成情况进行相应的处理
  - 如果数据传输有错，向上层软件报告设备的出错信息，实施重新执行
  - 如果正常结束，唤醒等待传输的进程，使其转换为就绪态
  - 如果有等待传输的I/O命令，通知相关软件启动下一个I/O请求

# 设备驱动程序



- 包括与设备密切相关的所有代码
- 从独立于设备的软件中接收并执行I/O请求
  - 把用户提交的逻辑I/O请求转化为物理I/O操作的启动和执行
  - 监督设备是否正确执行，管理数据缓冲区，进行必要的纠错处理

# 设备驱动程序的功能

- 设备初始化
  - 在系统初次启动或设备传输数据时，预置设备和控制器以及通道状态
- 执行设备驱动例程
  - 负责启动设备，进行数据传输
  - 对于具有通道方式，还负责生成通道指令和通道程序，启动通道工作
- 调用和执行中断处理程序
  - 负责处理设备 and 控制器及通道所发出的各种中断

# 设备驱动程序的层次

- 每个设备驱动程序只处理一种设备，或者一类紧密相关的设备
- 设备驱动程序分为整体驱动程序和分层驱动程序
  - 整体驱动程序直接向操作系统提供接口和控制硬件
    - 适用于功能简单的驱动程序，效率较高，但较难迁移
  - 分层驱动程序将驱动程序分成多层，放在栈中，系统接到I/O请求时先调用栈顶的驱动程序，栈顶的驱动程序可以直接处理请求或向下调用更低层的驱动程序，直至请求被处理
    - 适用于功能复杂、重用性要求较高的驱动程序，结构清晰且便于移植，但会增加一部分系统开销

# 独立于设备的I/O软件

- 执行适用于所有设备的常用I/O功能，并向用户层软件提供一致性接口
- 功能
  - 设备命名：通过路径名寻址设备
  - 设备保护：检查用户是否有权访问所申请设备
  - 提供与设备无关的数据单位：字符数量，块尺寸
  - 缓冲技术：传输速率，时间约束，不能直接送达目的地
  - 设备分配和状态跟踪：分配不同类型的设备
  - 错误处理和报告：驱动程序无法处理的错误

# 用户空间的I/O软件



- 库函数

- 一小部分I/O软件不在操作系统中，是与应用程序链接在一起的库函数，甚至完全由运行于用户态的程序组成
- 系统调用通常由库函数封装后供用户使用，封装函数只是将系统调用所用的参数放在合适位置，然后执行访管指令来陷入内核，再由内核函数实现真正的I/O操作

- SPOOLing软件

- 在内核外运行的系统I/O软件，采用预输入、缓输出和井管理技术，通过创建守护进程和特殊目录解决独占型设备的空占问题