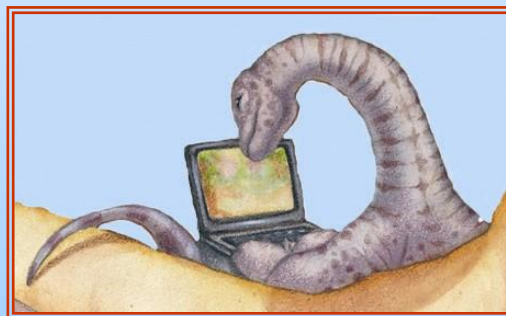


# 第十三章 I/O系统



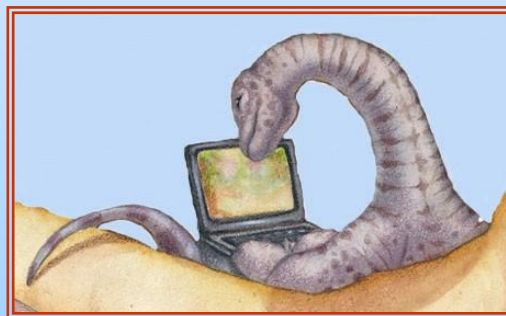


# 内容

1. I/O概述
2. I/O控制方式
3. I/O硬件
4. I/O设备接口



# 1、I/O概述





# 概述

- n 计算机2个主要任务：
  - | I/O操作-更加频繁
  - | 计算
- n I/O设备特点
  - | 种类繁多，不断有新设备出现
  - | 硬件与软件接口标准化：如USB
- n 如何适应I/O设备的特点是操作系统的一个挑战





# I/O系统的对象和任务

- n I/O系统的主要对象
  - | I/O设备和对应的设备控制器
  
- n I/O系统的主要任务
  - | 完成用户提出的I/O请求
  - | 提高I/O速率
  - | 改善I/O设备的利用率





# I/O系统的基本功能

1. 隐藏物理设备的细节
2. 与设备的无关性
3. 提高处理机和I/O设备的利用率
4. 对I/O设备进行控制
5. 确保对设备的正确共享
6. 错误处理





# 设备独立性

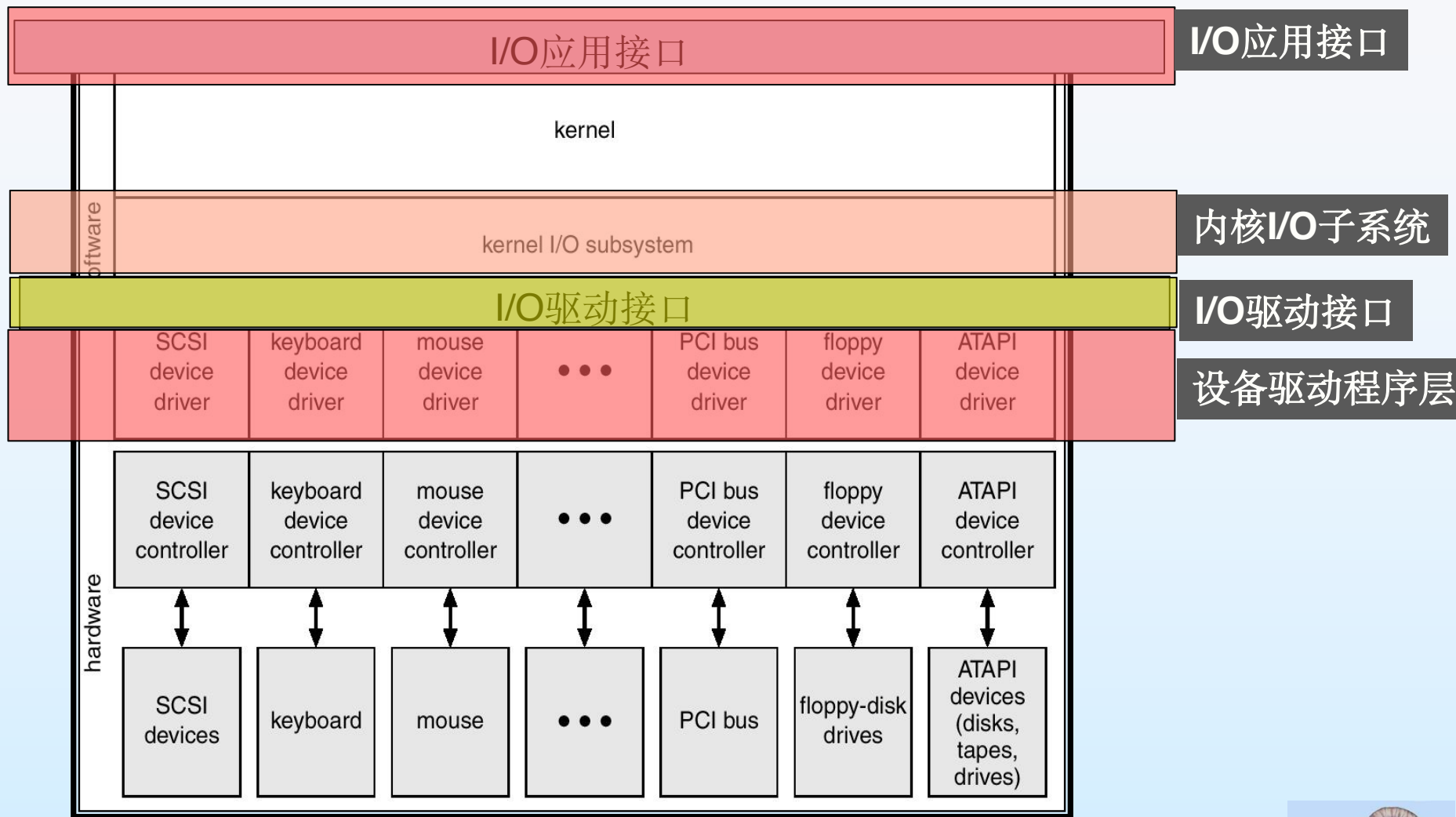
- n 为了提高操作系统的可适应性和可扩展性
  - | 现代操作系统中实现了**设备独立性**，也称为设备无关性
  - | 应用程序独立于具体使用的物理设备
- n 实现方法
  - | 在系统中引入**逻辑设备**和**物理设备**
  - | 在应用程序中 使用**逻辑设备名**来请求使用某类设备
  - | 而系统在实际执行时使用**物理设备名**
  - | 系统必须具有将**逻辑设备名转换为物理设备名**的功能
  - | 类似于存储管理中的逻辑地址和物理地址的概念





# 内核I/O结构

应用程序



I/O系统的核心是I/O管理和控制

硬件设备



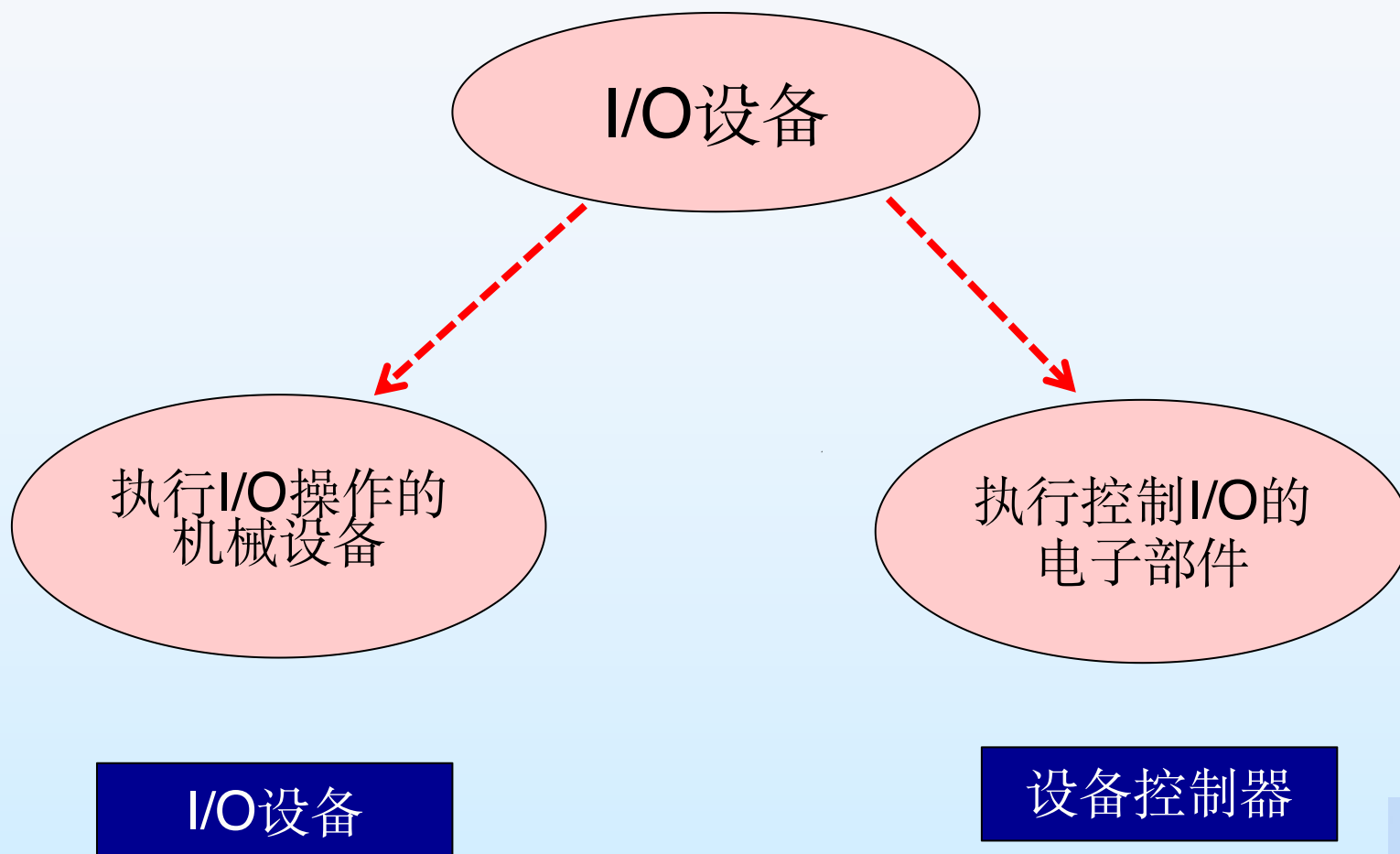




# I/O应用接口

- n I/O系统调用将设备的行为封装成一些通用类型，每个通用类型都可以通过一组标准函数（即接口）来访问，具体的差别被设备驱动程序所封装。
  - 。
- n 设备驱动层对内核隐藏了I/O（设备）控制器的不同细节
- n 将I/O系统与硬件分离，简化了操作系统开发人员的工作量；也有利于硬件制造商，可以设计新的设备，并使其与现有的控制器接口兼容，这样新的外设可以与原有计算机相连。无需等待操作系统开发商提供支持代码。
  - 。
- n 设备在许多方面有很大差异
  - ▶ 字符流或块
  - ▶ 顺序或随机访问
  - ▶ 共享或独占
  - ▶ 操作速度
  - ▶ 读写、只读、只写





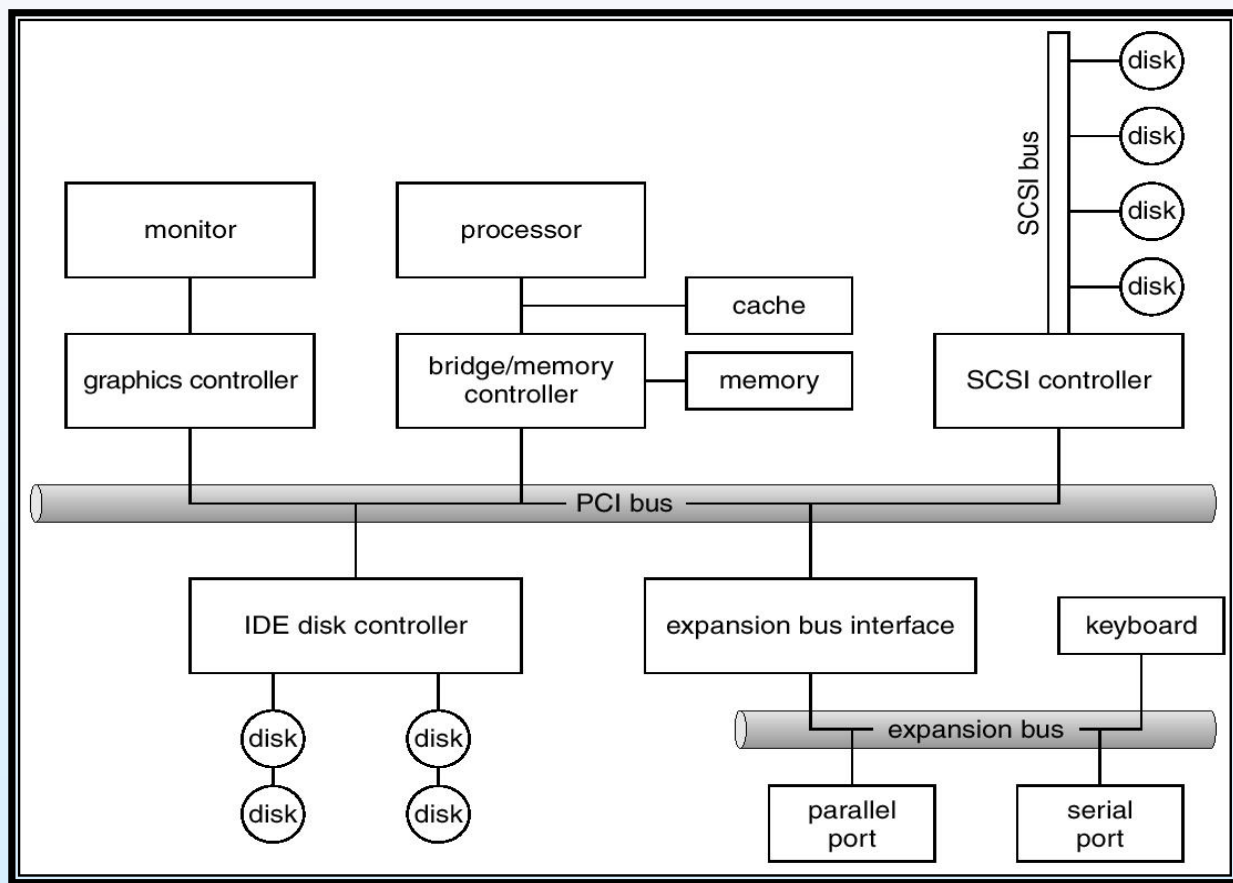


# I/O硬件

## n 概念

- | 端口
- | 总线（链环或共享总线）
- | 控制器（主机适配器）

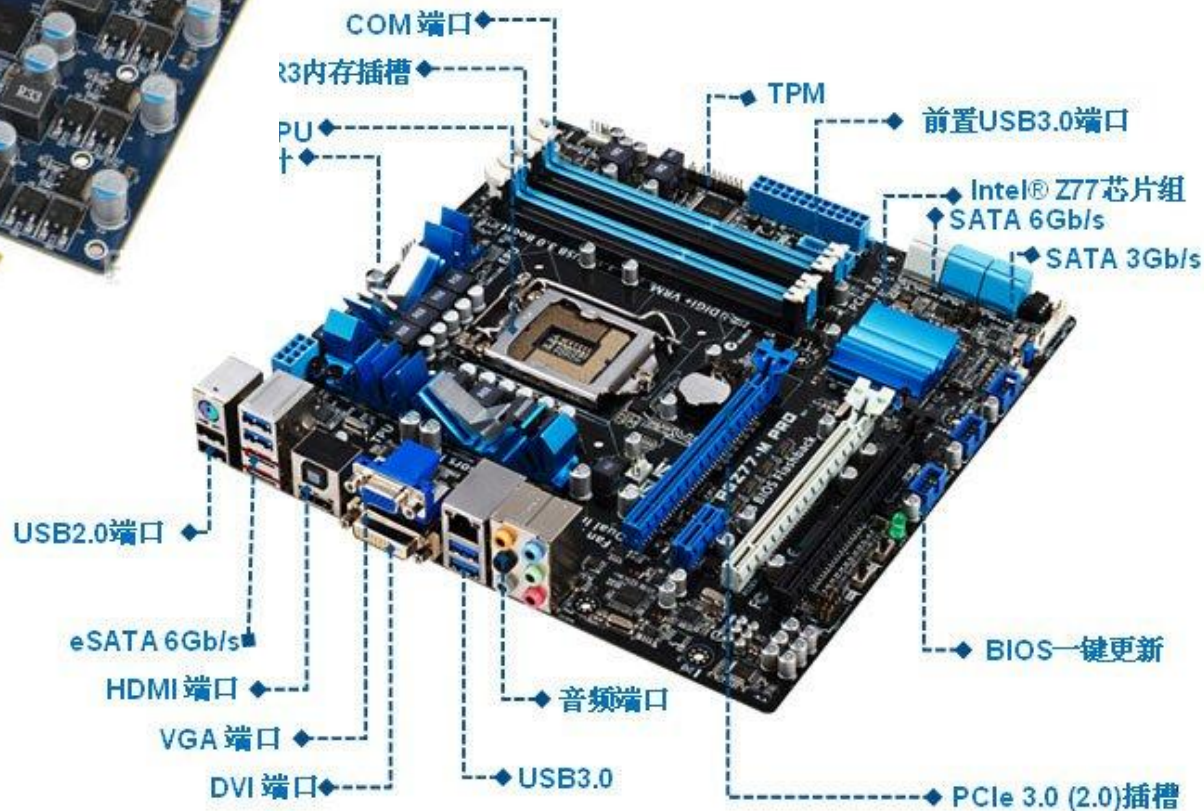
## n I/O指令控制设备





# 设备控制器

控制器 (Controller) 是用于操作端口、总线或设备的一组电子器件。





# 设备控制器

## n 设备控制器主要功能

- | 控制一个或多个I/O设备，以实现I/O设备和计算机之间的数据交换；它也是CPU与I/O设备之间的接口，接受从CPU发出来的命令，并控制I/O设备工作，以使CPU从繁杂的设备控制事务中解脱出来。

## n 驱动程序

- | 将抽象I/O命令转换成具体的命令和参数等转入设备控制器的相应寄存器，由控制器执行这些命令，具体实施对I/O设备的控制

。

## n I/O寄存器

- 状态寄存器
- 控制寄存器
- 数据输入寄存器
- 数据输出寄存器





# I/O硬件控制

n 控制设备方法（通过设备地址）：

- | 通过使用特殊的I/O指令来向指定的I/O端口地址传输字节，这使得访问内存和设备需要两种不同的指令
  - ▶ IN AL,60H
  - ▶ OUT 61H,AL
- | 内存映射I/O，设备控制器寄存器的地址映射为内存地址，处理器通过标准的内存传输指令来完成对I/O的操作，这种方法统一了访问方法，简化了I/O编程
  - ▶ 设备地址和内存统一编址







# 个人计算机中的设备I/O端口位置 (部分)

I/O address range (hexadecimal)	device
000-00F	DMA controller
020-021	interrupt controller
040-043	timer
200-20F	game controller
2F8-2FF	serial port (secondary)
320-32F	hard-disk controller
378-37F	parallel port
3D0-3DF	graphics controller
3F0-3F7	diskette-drive controller
3F8-3FF	serial port (primary)





# I/O设备的特点

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read&write	CD-ROM graphics controller disk





# I/O设备的类型

## n 按信息交换的单位分类

- l 块设备（以块为单位传输，如磁盘）、字符设备（一个字节一个字节地传输，如终端）

## n 按访问方法分类

- l 顺序访问（按其指定顺序传输数据，如调制解调器）、直接（随机）访问（可寻找任意数据存储位置，如CD-ROM）

## n 按传输速率分类

- l 低速设备：键盘、鼠标器、语音的输入和输出等
- l 中速设备：行式打印机、激光打印机等
- l 高速设备：磁带机、磁盘机、光盘机等





# I/O设备的类型

## n 按设备的共享属性分类

- l 独占设备（一个时间上只能由一个进程使用，如打印机、扫描仪等）、共享设备（可以被多个进程并发使用，如磁盘）

## n I/O方向分类

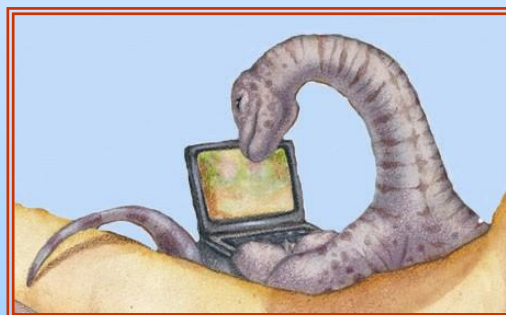
- l 只读设备（CD-ROM）、只写设备（图像控制器）、读写设备（磁盘）

## n 按使用特性分类

- l 存储设备、读写设备



## 2、I/O控制方式





# 对I/O设备的控制方式

- n 使用轮询的可编程I/O方式(现在基本不用)
- n 使用中断的可编程I/O方式(广泛采用)
  - | 可使CPU与I/O设备并行工作
- n 直接存储器访问(DMA)方式
  - | 进一步提高了CPU与I/O设备的并行操作程度
- n I/O通道控制方式 (特殊处理机、指令单一、没有自己的内容)





# 轮询 Polling

- n 又称“可编程I/O” (Programmed I/O)
- n 流程:
  1. 由CPU代表进程给I/O模块发I/O命令，询问设备是否忙
  2. 不忙即进行I/O，否则重复前面过程，直到操作完成才继续执行
- n 设备的状态
  - | 就绪、忙、出错
- n 轮询就是CPU不断读取设备状态寄存器，直到忙位被清除。
- n 实现容易，但效率偏低。
  - | CPU会长期处于忙等待





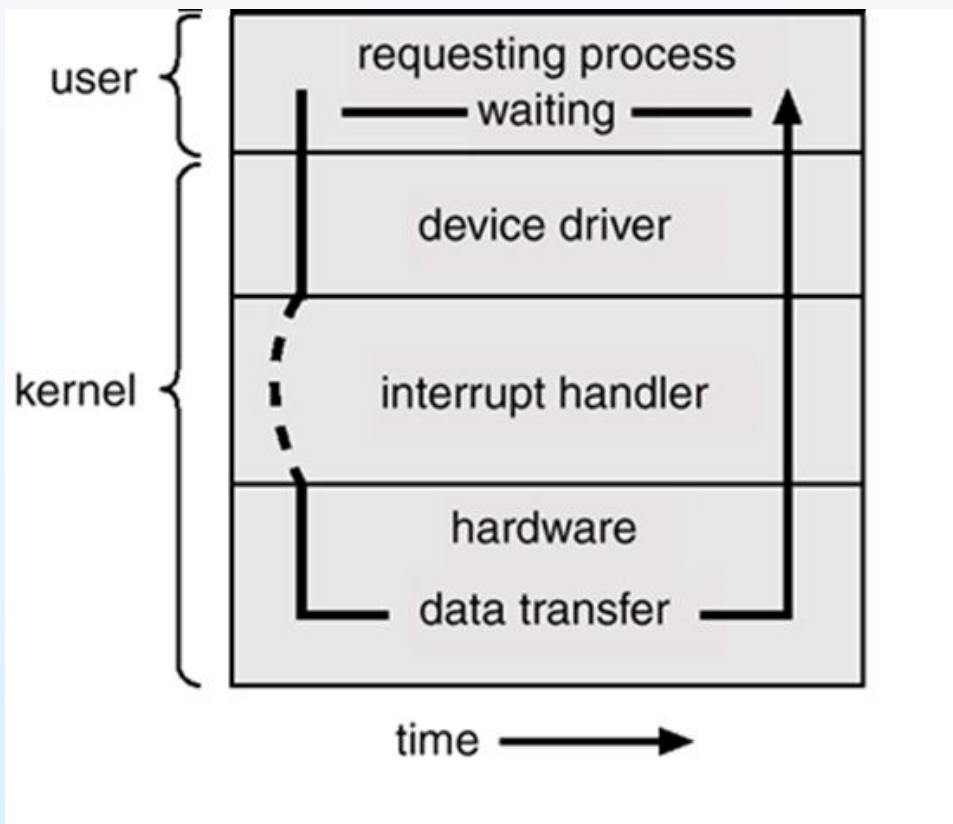
# 中断机制

- n CPU硬件有一根中断请求线，IRL
- n 基本中断工作机制
  - ① CPU执行完每条指令后，检测IRL
  - ② 如检测到信号，CPU保存当前状态，并跳转到中断处理程序。
  - ③ 执行中断处理程序
  - ④ 执行完后，清除中断，返回





# 中断驱动I/O方式



用户进程发出I/O请求

通过设备驱动程序

转到中断处理程序

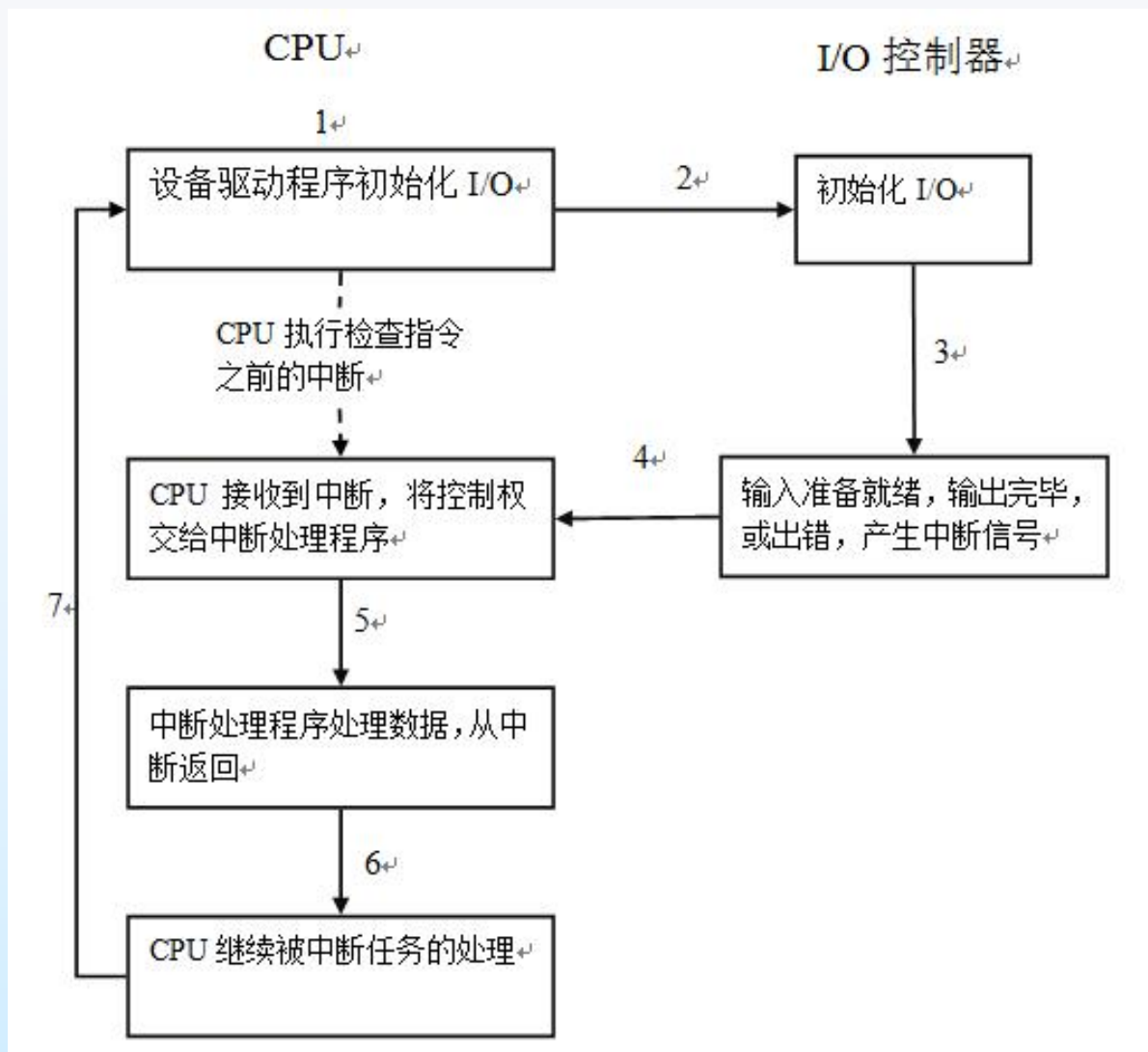
I/O设备完成数据传输

最终返回用户进程





# 中断驱动的I/O循环周期







# 中断

## n 中断和陷入

- | 中断(Interrupt)是指CPU对I/O设备发来的中断信号的一种响应
- | 陷入(Trap)是指CPU内部事件所引起的中断

## n 中断向量和中断向量表

- | 中断向量表存放每个设备的中断处理程序的入口地址，并将每个设备的中断请求作为一个中断号，对应于中断向量表中的一个表项。
- | 中断向量就是中断向量表中的一项内容，即中断处理程序的内存地址。

## n 中断优先级

- | 系统为每个中断规定了不同的优先级，这样可以使CPU延迟处理低优先级中断，也可以让高优先级中断抢占低优先级中断处理。如陷入所赋予的中断优先级低于硬件中断优先级。





# 对多中断源的处理方式

- n 中断源：引起中断的事件
- n 当处理机正在处理一个中断时，又来了一个新的中断请求，有两种处理方式
  - l 屏蔽(禁止)中断
  - l 嵌套中断





# Intel Pentium处理器的事件向量表

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts





# 中断处理程序主要的处理步骤

- ① 检测是否有未响应的中断信号
- ② 保护被中断进程的CPU环境
- ③ 转入相应的设备处理程序
- ④ 中断处理
- ⑤ 恢复CPU的现场并退出中断

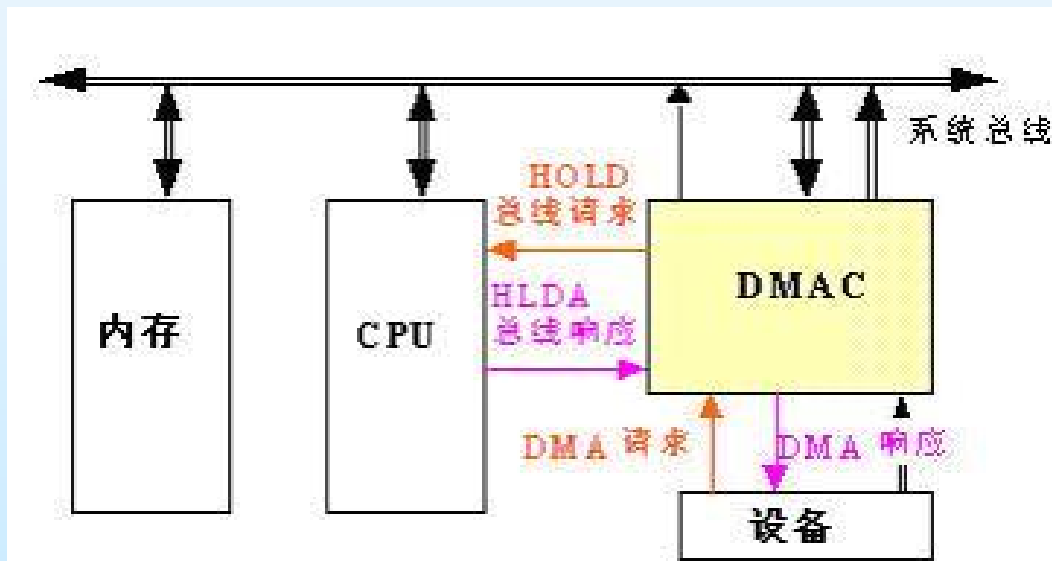
- n 中断是现代操作系统中用来处理异步事件和设置陷阱进入内核模式的管理程序。
- n 为了能使最紧迫的工作先做，现代计算机都使用中断优先级。
- n 设备控制器、硬件错误、系统调用，都可以引起中断，并触发内核程序。由于中断大量的用于时间敏感的处理，所以高性能系统需要高效中断处理。





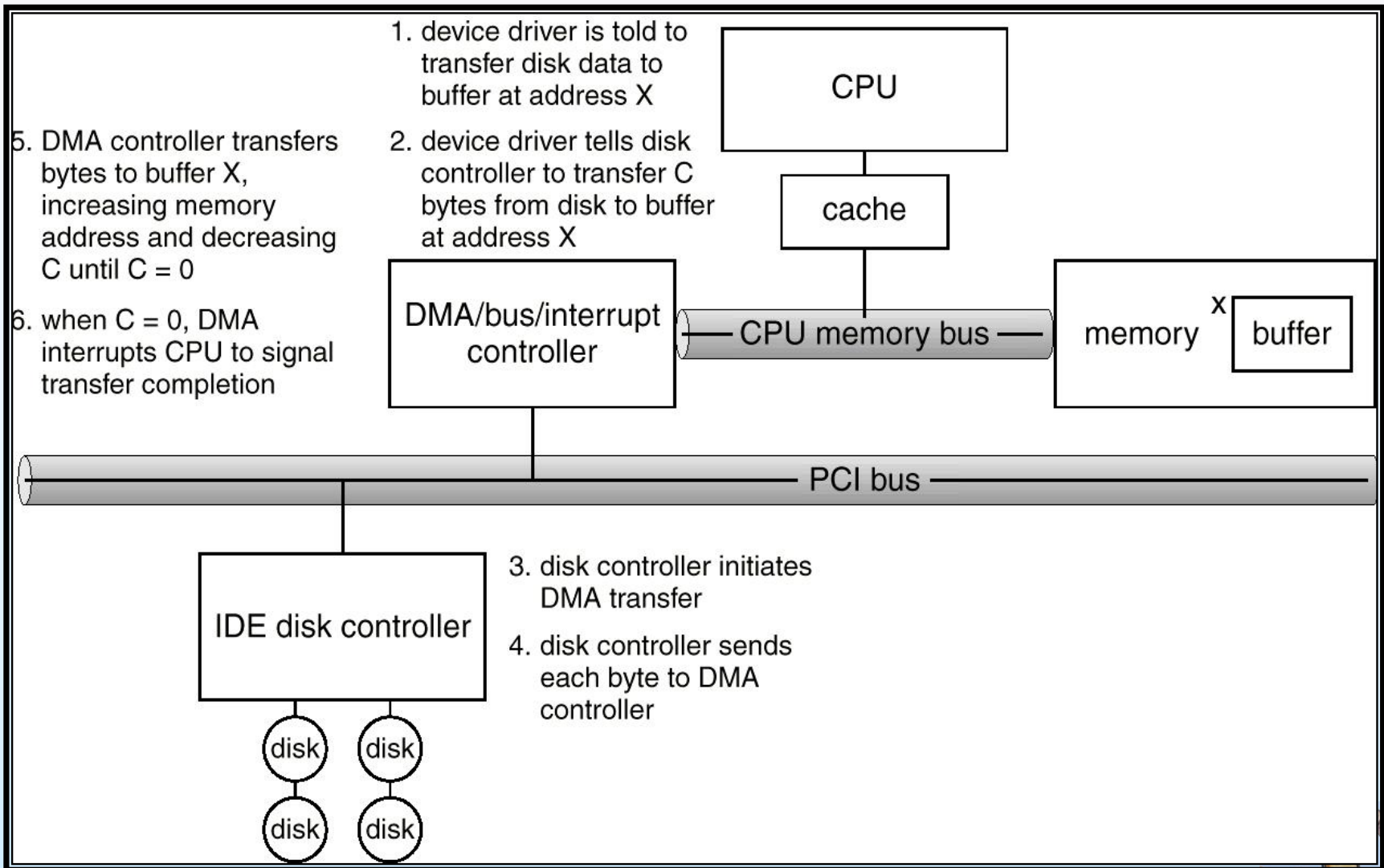
# 直接内存访问DMA

- n 为了避免用程序控制I/O来传输大量数据，需要专用的处理器，称之为直接内存访问控制器，简称DMA控制器
- n 绕过CPU，直接在I/O设备和内存之间传输数据
- ① 在开始DMA传输时，主机向内存写入DMA命令块，该块包括传输的源地址指针、目标地址指针、传输的字节数。
- ② CPU将该命令块的地址写入DMA控制器后，就继续其他工作。
- ③ DMA控制器继续工作，直接操作内存总线。无需主CPU帮助，可以将地址放到总线开始传输。





# DMA传输中的步骤

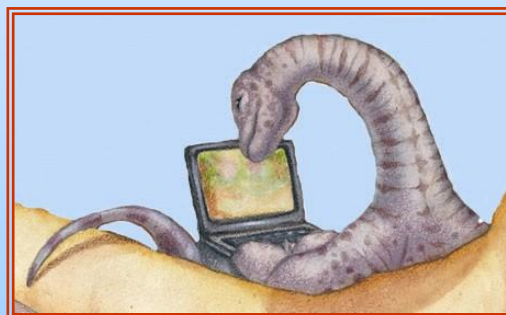




- 有的计算机体系结构的**DMA**使用物理内存地址
- 有的使用直接虚拟内存访问，称为**DVMA**，所使用的虚拟内存地址需要经过虚拟地址到物理地址转换
- **DVMA**可以直接实现两个内存映射设备之间的直接传输，而无需**CPU**的干涉或使用主内存。



### 3、I/O内核子系统







# I/O内核子系统

- n 内核提供了与I/O有关的服务。
  - | I/O调度
  - | 缓冲
  - | 高速缓存
  - | 假脱机
  - | 设备预留
  - | 错误处理
  - | I/O保护
- n 建立在硬件和设备驱动程序结构之上，还负责保护自己免受错误进程和恶意用户的危害

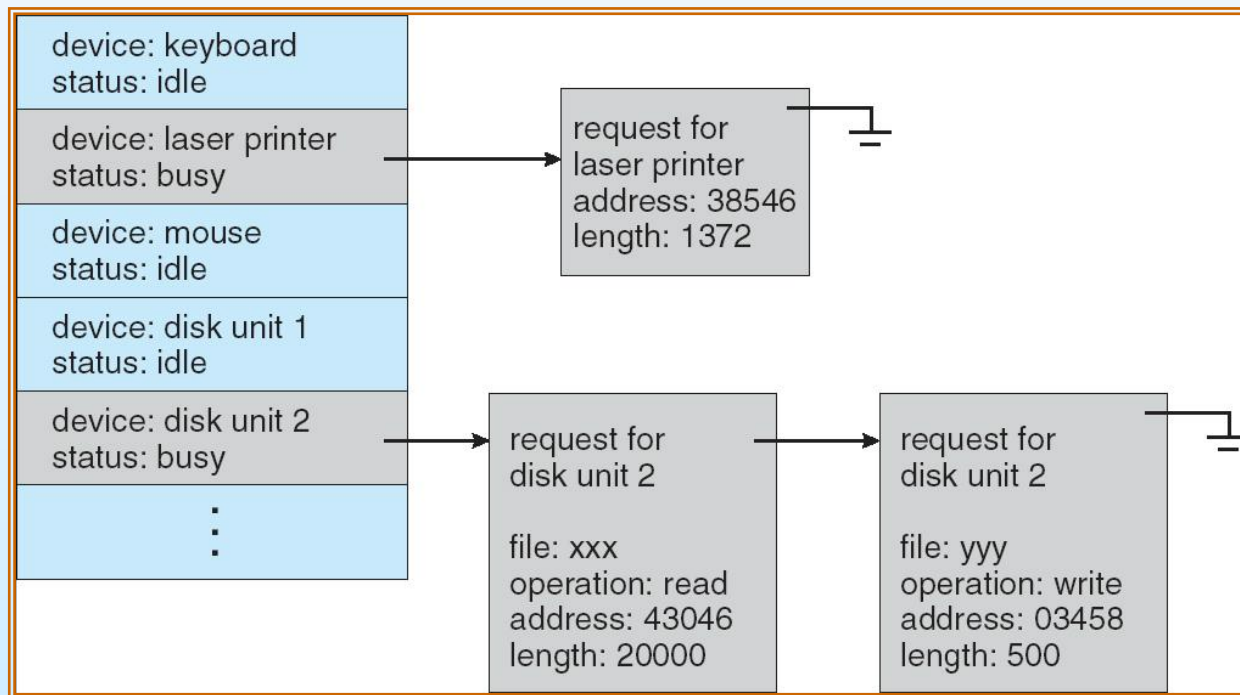




# I/O调度

- n 调度一组I/O请求就是确定一个合适的顺序来执行这些请求。调度能够改善系统整体性能，能在进程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。
- n 操作系统通过为每个设备维护一个请求队列来实现调度。同时为设备状态表配置等待队列。

- n 因此，I/O子系统改善计算机效率的一个方法是进行I/O调度
- n 操作系统试图维持公平
- n 绝大多数先来先服务算法



设备状态表



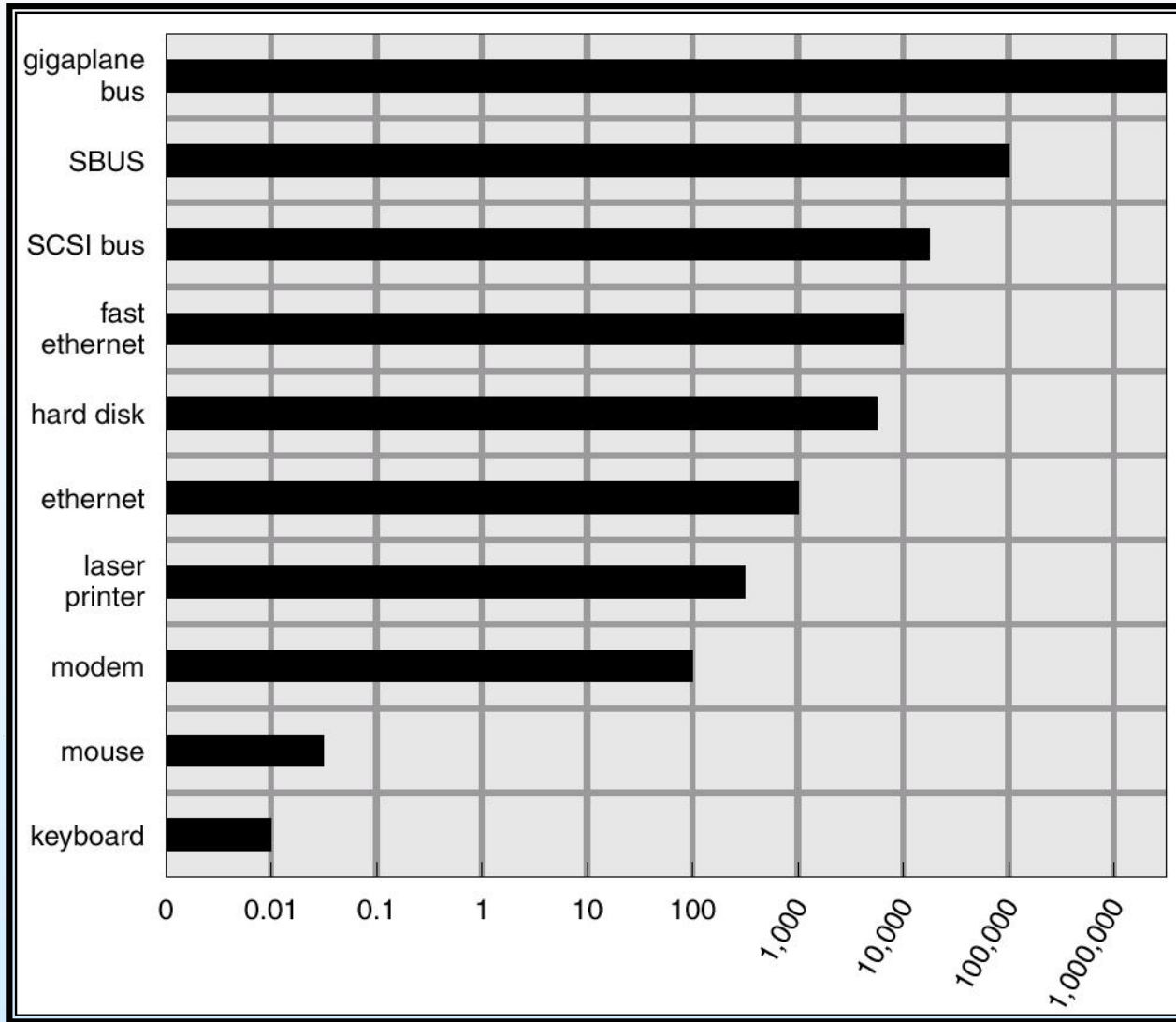
# 缓冲(Buffering)

- n 在现代操作系统中，几乎所有的I/O设备在与CPU交换数据时，都用了缓冲区。
- n 缓冲区是一个存储区域，用来保存两个设备之间，或设备 and 应用程序之间所传输数据的一个存储区域，可以由专门的硬件组成；更多的是利用内存。
- n 引入缓冲的理由
  - | 解决设备之间的速度差异
  - | 协调传输数据大小不一致
  - | 支持应用程序I/O的“复制语义”
- n 缓冲提高了CPU和I/O设备之间的并行性





# Sun Enterprise 6000的设备传输率





# 高速缓存(Cache)

- n 高速缓存Cache – 保留数据副本的高速存储器
  - | 仅仅是一个拷贝
  - | 关键是性能，高速缓冲区副本的访问比原来数据访问更为高效
- n 与缓冲的区别
  - | 缓冲可能是数据项的唯一一个副本
  - | 高速缓存只是提供了一个驻留在其他地方的数据在高速存储上的一个副本





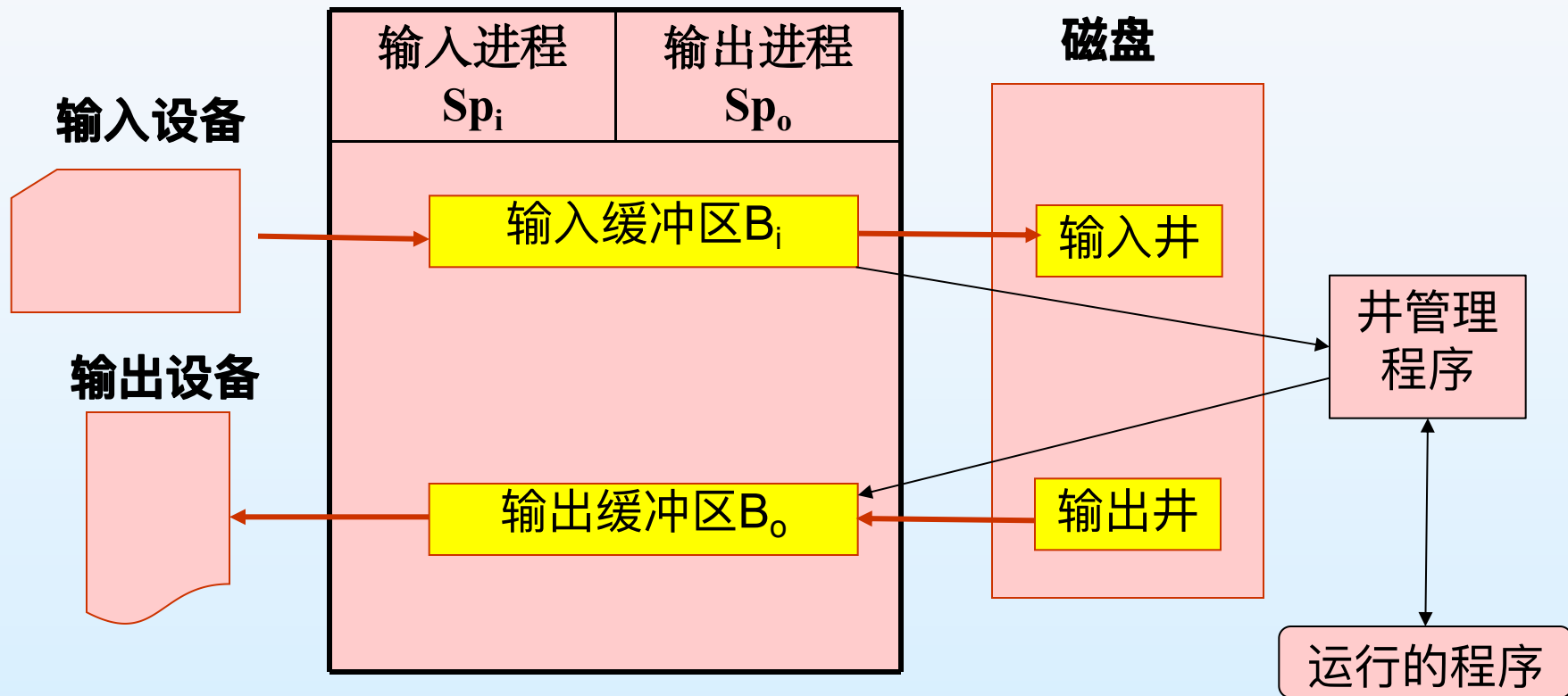
# 假脱机SPOOLing

- n 为了缓和CPU的高速性与I/O设备的低速性间的矛盾而引入了脱机输入、脱机输出技术。
  - | 利用一道程序，模拟脱机输入时的外围控制机的功能，把低速I/O设备上的数据传送到高速磁盘上
  - | 用另一道程序，模拟脱机输出时的外围控制机的功能，把数据从磁盘传送到低速输出设备
  - | 这样就可以在主机的直接控制下，实现脱机输入、输出功能。
  - | 此时的外围操作与CPU对数据的处理同时进行，这种在联机情况下实现的同时外围操作称为假脱机技术SPOOLing。





# SPOOLing系统的组成





# SPOOLing系统的特点

- n 提高了I/O的速度
- n 将独占设备改造为共享设备，因此实现了虚拟设备功能
- n 虚拟设备是指由独占设备改造为共享设备的那个设备，这个设备只是逻辑上的设备







# 假脱机打印系统

- n 打印机属于独占设备，利用SPOOLing技术，可将之改造为可供多个用户共享的设备，从而提高设备的利用率
- n 共享打印机技术已被广泛用于多用户系统和局域网中
- n 假脱机打印系统
  - | 磁盘缓冲区：是一块磁盘空间，用来暂存用户程序的输出数据，也就是输出井
  - | 打印缓冲区：设在内存，暂存从磁盘缓冲区送来的数据，也就是输出缓冲区
  - | 假脱机管理进程和假脱机打印进程
    - 假脱机管理进程为每个要求打印的用户数据建立一个假脱机文件，并放入文件队列中
    - 假脱机打印进程依次对队列中的文件进行打印





# 假脱机打印系统

- n 当用户进程发出打印请求时，假脱机打印系统并不是立即把打印机分配给该用户进程，而是由假脱机管理进程完成两项工作：
  - l 1) 在磁盘缓冲区为之申请一个空闲盘块，并将要打印的数据送入其中暂存；
  - l 2) 为用户进程申请一张空白的用户请求打印表，填入用户的打印请求，再将该表挂到假脱机文件队列上。
- n 真正的打印输出由假脱机打印进程负责：
  - l 当打印机空闲时，进程从请求打印队列的队首取出一张请求打印表，根据表中的请求将要打印的数据从输出并传送到内存缓冲区，再由打印机进行打印。
  - l 打印完，进程再次查看请求打印列表，若非空，重复上述工作，直到队列为空。此后进程才将自己阻塞起来，仅当下次再有打印请求时，进程才被唤醒。





# I/O内核子系统

## n 错误处理

- | 操作系统可以从磁盘读、设备无效、暂时写失败等错误中恢复
- | 当出现I/O请求失败时，多数情况返回一个错误号
- | 系统出错日志记录了错误报告

## n I/O保护 – 防止非法I/O指令的执行

- | 定义I/O指令为特权指令
- | 应用程序I/O操作必须通过系统调用实现

## n 内核数据结构

- | 内核保存了I/O组件的状态信息，包括文件打开表、网络连接、字符设备状态等



