Ch 10 I/O 管理

1. I/O 管理概述

1. 1 I/O 的特点

- I/O 的性能经常是系统性能的瓶颈
- 操作系统庞大复杂的原因：资源多、杂，并发，均来自 I/O
- 与其他功能联系密切，特别是文件系统

设备控制器：

1. CPU 和 I/O 设备间的接口
2. 向 CPU 提供特殊指令和寄存器

1. 2 设备的分类

按数据特征分

- 字符设备：
    - 以字节为单位存储、传输信息
    - 传输速率低，不可寻址
- 块设备
    - 以数据块为单位存储、传输信息
    - 传输速率高，可寻址（随机读写）
- 网络设备
    - 格式化报文交换

从资源分配角度

- 独占设备
    - 在一段时间内只能有一个进程使用的设备，一般为低速 I/O 设备

- 共享设备
    - 在一段时间内可有多个进程共同使用的设备，多个进程以交叉的方式来使用设备，**资源利用率高**。
- 虚设备
    - 在一类设备上模拟另一类设备，常用共享设备模拟独占设备，用高速设备模拟低速设备，被模拟的设备称为虚设备
    - 目的：将慢速的独占设备改造成多个用户可共享的设备，**提高设备的利用率**。

1.4 I/O 管理的目标和任务

1. 按照用户的请求，控制设备的各种操作，完成 I/O 设备与内存之间的数据交换，最终完成用户的 I/O 请求

- 设备分配与回收
    - 记录设备的状态
    - 根据用户的请求和设备的类型，采用一定的分配算法，选择一条数据通路
- 执行设备驱动程序，实现真正的 I/O 操作
- 设备中断处理：处理外部设备的中断
- 缓冲区管理：管理 I/O 缓冲区

2. 建立方便、统一的独立于设备的接口

- **方便性**：向用户提供使用外部设备的方便接口，使用户编程时不考虑设备的复杂物理特性
- **统一性**：对不同的设备采取统一的操作方式，在用户程序中使用的是逻辑设备
- 逻辑设备与物理设备、屏蔽硬件细节

3. 充分利用各种技术提高 CPU 与设备、设备与设备之间的并行工作能力，充分利用资源，提高资源利用率
4. 保护
    - 设备传送或管理的数据应该是安全的、不被破坏的、保密的

2. I/O 硬件组成

2. 1 I/O 设备组成

I/O 设备一般由**机械**和**电子**两部分组成

1. 机械部分是设备本身（物理装置）
2. 电子部分又称设备控制器（适配器）

2. 2 设备接口——控制器的作用

操作系统将命令写入控制器的接口寄存器中，实现输入/输出，并从接口寄存器读取状态信息或结果信息

控制器的任务：把串行的位流转换为字节块，并进行必要的错误修正。首先，控制器按位进行组装，然后存入控制器内部的缓冲区中形成以字节为单位的块；在对块验证检查和并证明无错误时，再将它复制到内存中

2. 3 I/O 端口地址

I/O 端口地址：接口电路中每个寄存器具有的、唯一的地址，是个整数，亦称为 I/O 地址

所有 I/O 端口地址形成 I/O 端口空间

I/O 指令形式与 I/O 地址是相互关联的，拥有两种形式

- 内存映射编址（内存映射 I/O 模式）
- I/O 独立编址（I/O 专用指令）

I/O 独立编址

分配给系统中所有端口的地址空间完全独立，与内存地址空间无关

使用专门的 I/O 指令对端口进行操作

优点

- 外部设备不占用内存的地址空间
- 编程时，易于区分是对内存操作还是对 I/O 端口操作

缺点：

- I/O 端口操作的指令类型少，操作不灵活

内存映射编址

- 分配给系统中所有端口的地址空间与内存的地址空间统一编址
- 把 I/O 端口看作一个存储单元，对 I/O 的读写操作等同于对内存的操作
- 优点
    - 凡是可对内存操作的指令都可对 I/O 端口操作
    - 不需要专门的 I/O 指令
    - I/O 端口可占有较大的地址空间
- 缺点
    - 占用内存空间

内存映射 I/O 的优点

- 不需要特殊的保护机制来阻止用户进程执行 I/O 操作

  操作系统要做的事情：避免把包含控制寄存器的那部分地址空间放入任何用户的虚拟地址空间中
- 可以引用内存的每一条指令也可以引用控制寄存器

内存映射 I/O 的缺点

- 对一个设备控制寄存器不能进行高速缓存
- 硬件需要针对每个页面具备选择性禁用高速缓存的能力，操作系统必须管理选择性高速缓存，所以这一特性为硬件和操作系统两者增添了额外的复杂性

2.4 I/O 控制方式

1. 可编程 I/O（轮询、查询）

   由 CPU 代表进程给 I/O 模块发 I/O 命令，进程进入忙等待，直到操作完成才继续执行

2. 中断驱动 I/O

   - 目的：减少设备驱动程序不断询问控制器状态寄存器的开销
   - 实现：I/O 操作结束后，由设备控制器主动通知设备驱动管理程序

3. DMA 技术——直接存储器访问

3. I/O 软件组成

I/O 软件设计

分层的设计思想

- 把 I/O 软件组织成多个层次
- 每一层都执行操作系统所需要的功能的一个相关子集，它依赖于更低一层所执行的更原始的功能，从而可以隐藏这些功能的细节，同时，又给更高一层提供服务
- 较低层考虑硬件的特性，并向较高层软件提供接口
- 较高层不依赖于硬件，并向用户提供一个友好的、清晰的、简单的、功能更强的接口

设备独立性

用户编写的程序可以访问**任意** I/O 设备，无需事先指定设备

- 从用户角度：用户在编制程序时，使用逻辑设备名，由系统实现从逻辑设备到物理设备（实际设备）的转换，实施 I/O 操作。
- 从系统角度：设计并实现 I/O 软件时，除了直接与设备打交道的底层软件之外，其他部分的软件不依赖于硬件

好处：

- **设备分配时非常灵活**
- **容易实现 I/O 重定向**

4. I/O 管理相关技术

4. 1 缓冲技术

操作系统中最早引入的技术

- 解决 CPU 与 I/O 设备之间速度不匹配的问题

  凡是数据到达和离去速度不匹配的地方均可采用缓冲技术
- 提高 CPU 与 I/O 设备之间的并行性
- 减少了 I/O 设备对 CPU 的中断请求次数，放宽 CPU 对中断响应时间的要求

缓冲区分类

- **硬缓冲**：由硬件寄存器实现（设备中设置的缓冲区）
- **软缓冲**：在内存中开辟一个空间，用作缓冲区

缓冲区管理

- 单缓冲
- 双缓冲
- **缓冲池**：多缓冲，循环缓冲：管理多个缓冲区，使用有界缓冲区的生产者/消费者模型对缓冲池中的缓冲区进行循环利用

缓冲池可以平滑和加快信息在内存和磁盘之间的传输

缓冲区结合提前读和延迟写技术对具有重复性及阵发性 I/O 进程、提高 I/O 速度有很多帮助

可以充分利用之前从磁盘读入、虽已传入用户区但仍在缓冲区的数据（减少 I/O 速度，提高系统运行速度）

4.2 设备管理有关的数据结构

- 描述设备、控制器等部件的表格
- 建立同类资源的队列
- 面向进程 I/O 请求的动态数据结构
- 建立 I/O 队列

独立设备的分配

- 静态分配
    - 在进程运行前，完成设备分配，运行结束后，收回设备
    - 缺点：设备利用率低
- 动态分配
    - 在进程运行过程中，当用户提出设备要求时，进行分配，一旦停止使用立即收回
    - 优点：效率好；缺点：分配策略不好时，产生死锁

分时式共享设备的分配

以一次 I/O 为单位分时使用设备，不同进程的 I/O 操作请求以排队方式分时地占用设备进行 I/O

由于同时有多个进程同时访问，且访问频繁，就会影响整个设备使用效率，影响系统效率。因此要考虑多个访问请求到达时服务的顺序，使平均服务时间越短越好

4.3 设备驱动程序

- 与设备密切相关的代码放在设备驱动程序中，每个设备驱动程序处理一种设备类型
- 设备驱动程序的任务是接收来自与设备无关的上层软件的抽象请求，并执行这个请求
- 每一个控制器都设有一个或多个设备寄存器，用来存放向设备发送的命令和参数；设备驱动程序负责释放这些命令，并监督它们正确执行
- 设备驱动程序与外界的接口
    - 与操作系统的接口
    - 与系统引导的接口

- 与设备的接口

4.4 I/O 进程

I/O 进程：专门处理系统中的 I/O 请求和 I/O 中断工作

I/O 进程

- 是系统进程，一般赋予**最高优先级**，一旦被唤醒，可以很快抢占投入运行
- 开始运行时，首先关闭中断，然后用 receive 接收消息
- 两种情形：
    - 没有消息：开中断，将自己阻塞
    - 有消息：判断消息类型

5. I/O 性能问题

- 使 CPU 利用率尽可能不被 I/O 降低
- 使 CPU 尽可能摆脱 I/O

采取的措施

- 减少或缓解速度差距 $\rightarrow$ 缓冲技术
- 使 CPU 不等待 I/O $\rightarrow$ 异步 I/O
- 让 CPU 摆脱 I/O 操作 $\rightarrow$ DMA、通道

同步 I/O：

- 在 I/O 处理过程中，CPU 处于空闲等待状态
- 在处理数据过程中，不能同时进行 I/O 操作

异步传输

异步模式——用于优化应用程序的性能

- 通过异步 I/O，应用程序可以启动一个 I/O 操作，然后在 I/O 请求执行的同时继续处理
- 基本思想：填充 I/O 操作间等待的 CPU 时间

同步 I/O

- 应用程序被阻塞直到 I/O 操作完成