



第六章 输入输出系统

- 6.1 I/O设备(6.2.1+6.4.2)
- 6.2 用户层的I/O软件(6.6)
- 6.3 缓冲区管理(6.7)
- 6.4 磁盘存储器的性能与调度(6.8)



6.1 I/O设备

1、按使用特性分

人机交互类外设，如打印机、显示器、键盘、鼠标等

存储设备，如磁盘、光盘等

网络通信设备，如各种网络接口、调制解调器

2、按传输速率分

低速设备，如键盘、鼠标。每秒几个到数百个字节

中速设备，如打印机。每秒数千个字节至数万个字节

高速设备，如磁盘。每秒数百个千字节到千兆字节

3、按信息交换单位分

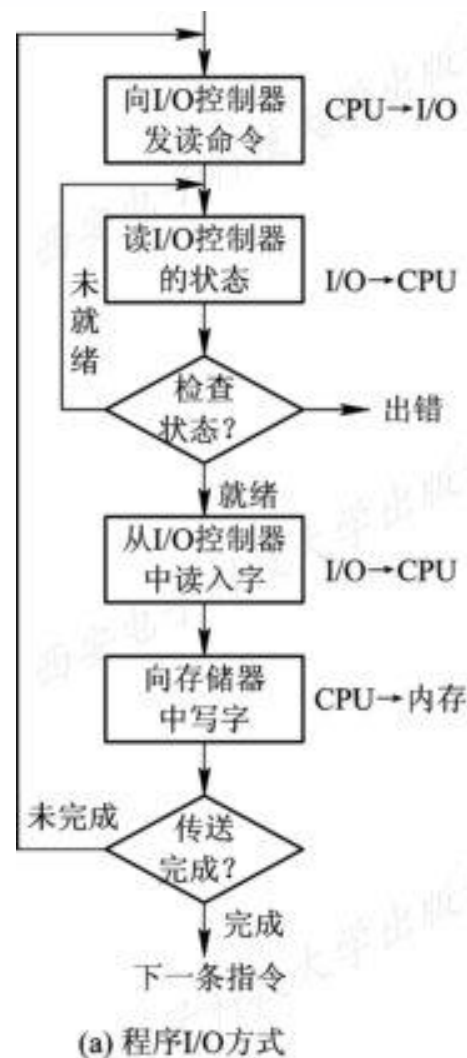
块设备，属于有结构设备，如磁盘

字符设备，属于无结构设备，如打印机

6.2 I/O控制方式

1、程序直接控制方式

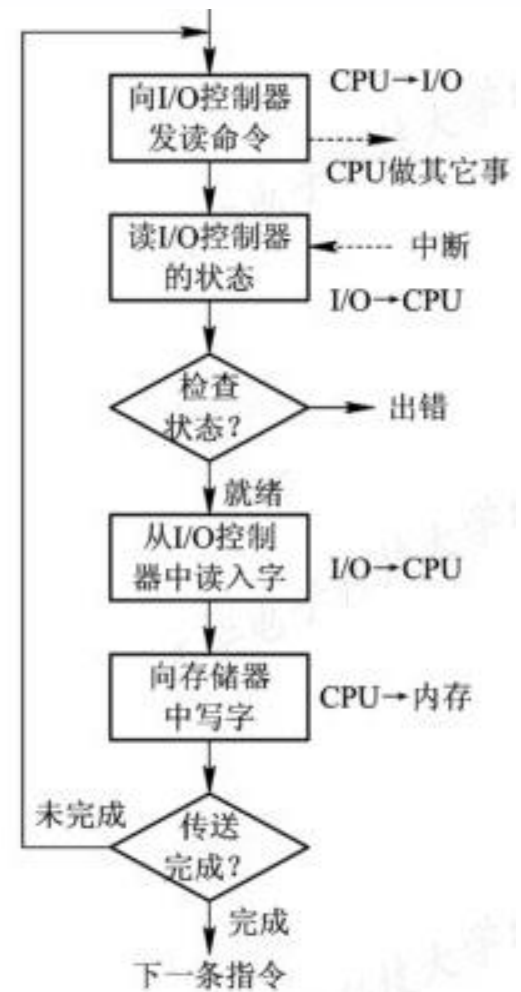
对读入的每个字，CPU需要对外设状态进行循环检查，直到确定该字已经在I/O控制器的数据寄存器中。由于CPU的高速性和I/O设备的低速性，CPU大部分时间都处于等待中。



6.2 I/O控制方式

2、中断驱动方式

允许I/O设备主动打断CPU的运行并请求服务，从而解放CPU，使得其向I/O控制器发送读命令后可以继续做其他有用工作。



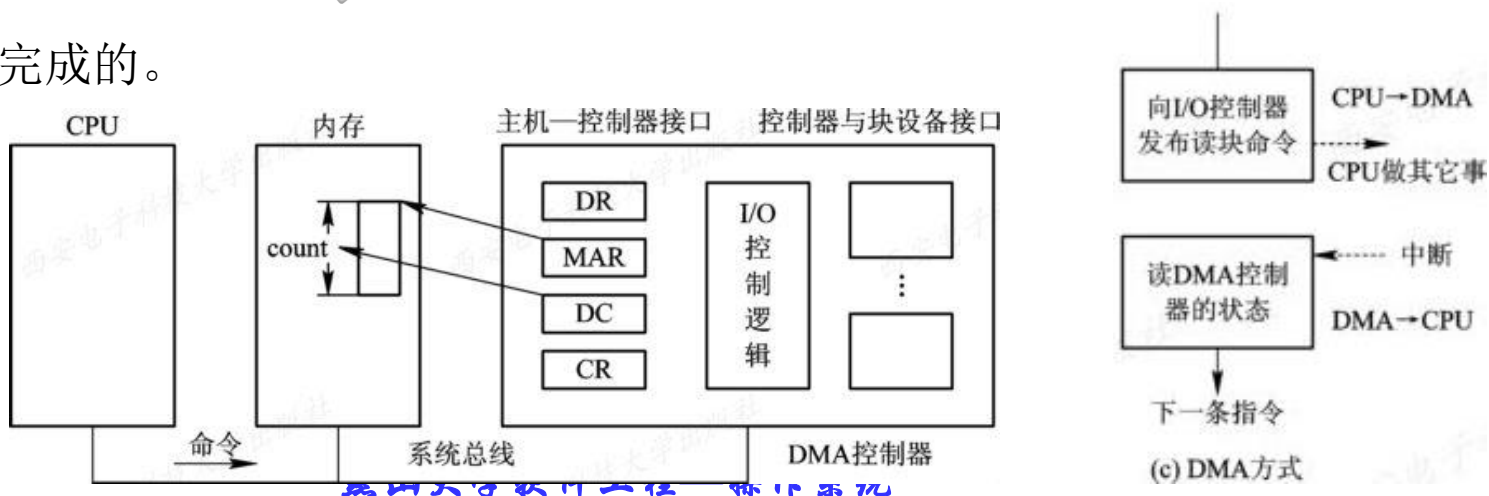
(b) 中断驱动I/O方式


6.2 I/O控制方式

3、DMA方式

中断方式中，I/O 设备与内存之间的数据交换必须要经过CPU中的寄存器，所以速度受限。DMA（直接存储器存取）方式是在I/O设备和内存之间开辟直接的数据交换通路。

中断驱动方式在每个数据需要传输时中断CPU，而DMA控制方式则是在所要求传送的一批数据全部传送结束时才中断CPU；此外，中断驱动方式数据传送是在中断处理时由CPU控制完成的，而DMA控制方式则是在DMA控制器的控制下完成的。






6.2 I/O控制方式

4、通道控制方式

I/O通道指专门负责输入/输出的处理机，是DMA方式的发展。

虽然DMA方式比起中断方式来已经显著地减少了CPU的干预，即已由以字(节)为单位的干预减少到以数据块为单位的干预，但CPU每发出一条I/O指令，也只能去读(或写)一个连续的数据块。而当我们一次去读多个数据块且将它们分别传送到不同的内存区域，或者相反时，则须由CPU分别发出多条I/O指令及进行多次中断处理才能完成。

通道是通过执行通道程序并与设备控制器共同实现对I/O设备的控制的。通道程序是由一系列通道指令(或称为通道命令)所构成的



6.3 用户层的I/O软件

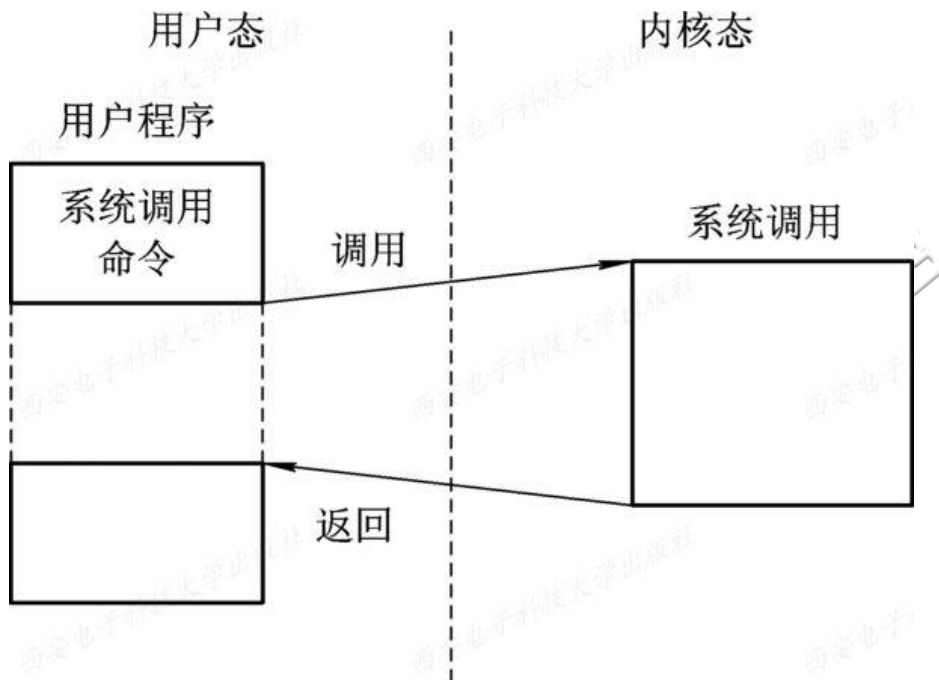
6.3.1 系统调用与库函数

1. 系统调用

一方面，不允许运行在用户态的应用进程去直接调用运行在核心态（系统态）的OS过程。

另一方面，应用进程在运行时，又必须取得OS所提供的服务，否则，应用程序几乎无法运行。

所以，OS在用户层中引入了一个中介过程——系统调用，应用程序可以通过它间接调用OS中的I/O过程，对I/O设备进行操作。




系统调用是应用程序取得OS所有服务的唯一途径。

早期，在汇编语言编写程序时，可以直接使用系统调用。

后来，C语言中，首先提供了与系统调用相对应的库函数。

图6-20 系统调用的执行过程




2. 库函数

在C语言以及UNIX系统中，系统调用(如read)与各系统调用所使用的库函数(如read)之间几乎是一一对应的。

微软定义了一套过程，称为Win32 API的应用程序接口(Application Program Interface)，程序员利用它们取得OS服务，该接口与实际的系统调用并不一一对应。

用户程序通过调用对应的库函数使用系统调用，这些库函数与调用程序连接在一起，被嵌入在运行时装入内存的二进制程序中。



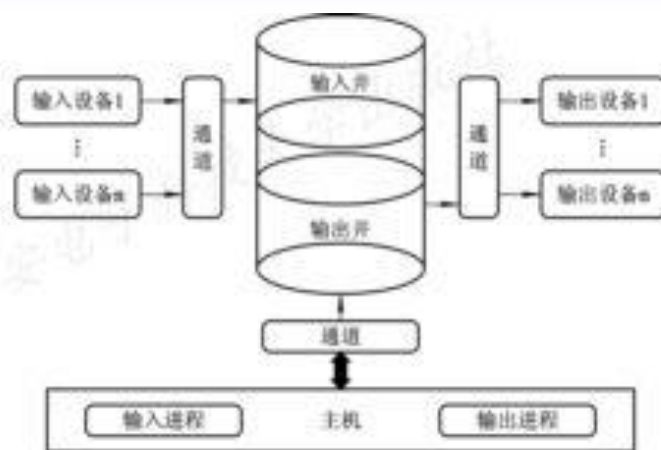
6.3.2 假脱机(Spooling)系统

1. 假脱机技术

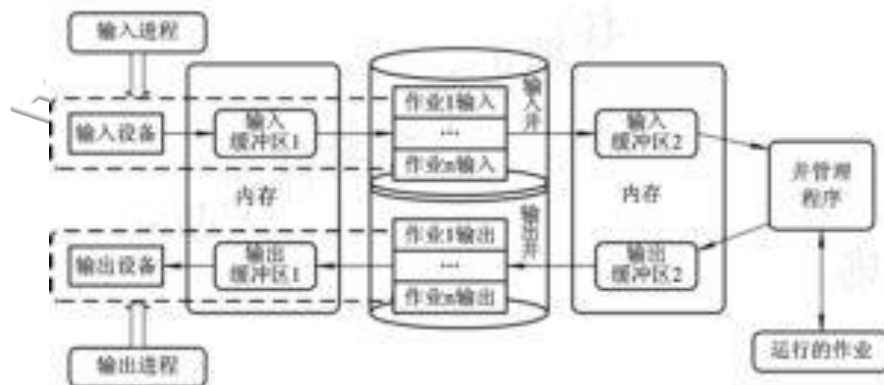
脱机输入、输出：利用专门的外围控制机，先将低速I/O设备上的数据传送到高速磁盘上，或者相反。这样当处理机需要输入数据时，便可以直接从磁盘中读取数据，极大地提高了输入速度。

假脱机技术：利用多道程序中的一道程序模拟脱机输入时的外围机功能，再利用另外一道程序模拟脱机输出时的外围控制机功能。这样可以在主机的直接控制下，实现以前的脱机输入、输出。外围操作与CPU对数据的处理同时进行，把这种在联机情况下实现的同时外围操作技术称为SPooling技术，或假脱机技术。

6.3.2 假脱机(Spooling)系统



(a) SPOOLing系统的组成

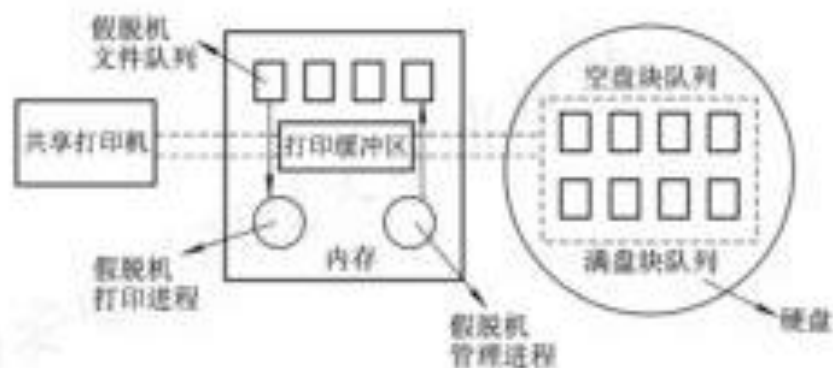


(b) SPOOLing的工作原理

2. 假脱机打印系统

打印机是经常用到的输出设备，属于独占设备。利用假脱机技术可将它改造为一台可供多个用户共享的打印设备，从而提高设备的利用率，也方便了用户。共享打印机技术已被广泛地用于多用户系统和局域网络中。假脱机打印系统主要有以下三部分：

- (1) 磁盘缓冲区。
- (2) 打印缓冲区。
- (3) 假脱机管理进程和假脱机打印进程。



(c) 假脱机打印系统的组成



6.4 缓冲区管理

6.4.1 缓冲的引入

引入缓冲区的原因有很多，可归结为以下几点：

- (1) 缓和CPU与I/O设备间速度不匹配的矛盾。
- (2) 减少对CPU的中断频率，放宽对CPU中断响应时间的限制。
- (3) 解决数据粒度不匹配的问题。
- (4) 提高CPU和I/O设备之间的并行性。

6.4.2 各种缓冲区

1. 单缓冲区(Single Buffer)

在设备和处理机之间设置一个缓冲区，交换数据时，先把被交换的数据写入缓冲区，再由需要的设备或处理机将数据从缓冲区取走。**存完才可以取，取光才可以再存。**

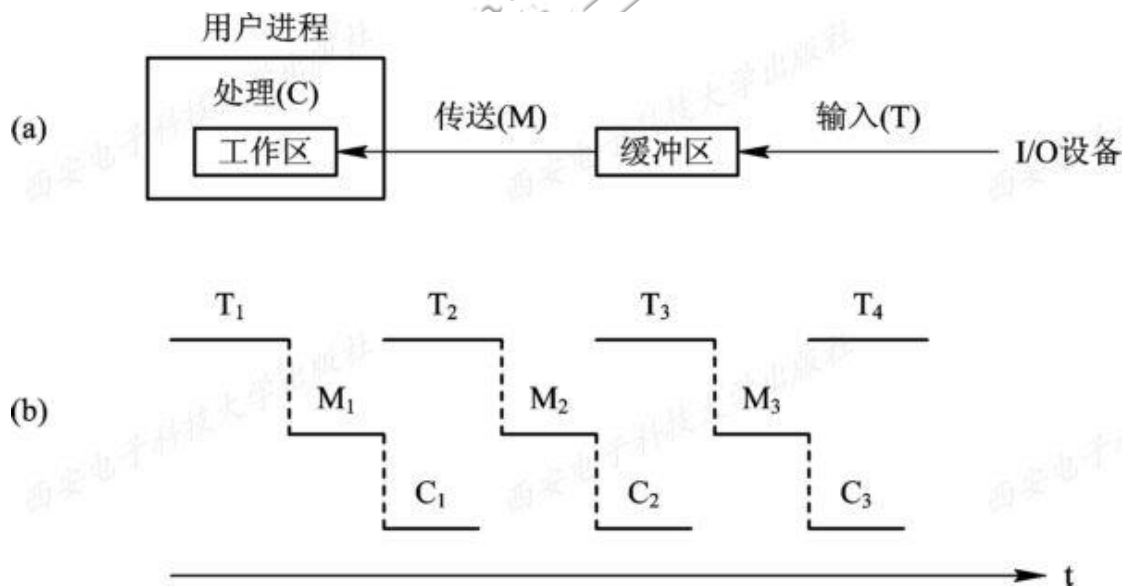


图6-23 单缓冲工作示意图



2. 双缓冲区(Double Buffer)

根据单缓冲的特点，CPU在传送时间内处于空闲状态，由此引入双缓冲。I/O设备输入数据时先装填到缓冲区1，在缓冲区1满后才开始装填缓冲区2，与此同时，处理机可以从缓冲区1中取出数据放入用户进程处理。当缓冲区1中的数据处理完后，如果缓冲区2已经填满，则处理机又从缓冲区2中取出数据放入用户进程处理。

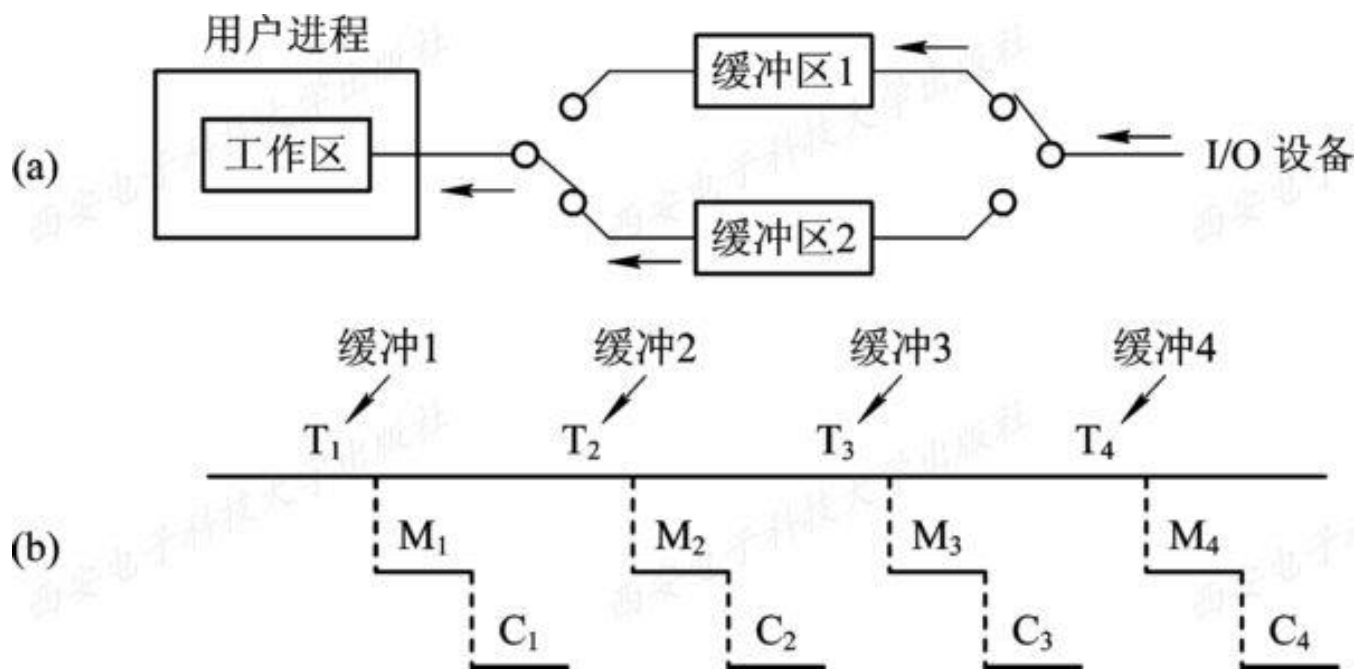



图6-24 双缓冲工作示意图



3、 环形缓冲区

包括多个缓冲区，其每个缓冲区的大小相同，每个缓冲区中有一个链接指针指向下一个缓冲区，最后一个缓冲区指针指向第一个缓冲区，多个缓冲区构成一个环形。

4、 缓冲池

由多个系统公用的缓冲区组成，缓冲区按其使用状况可以形成三个队列：空缓冲队列，装满输入数据的缓冲队列、装满输出数据的缓冲队列。