



11

马佳曼

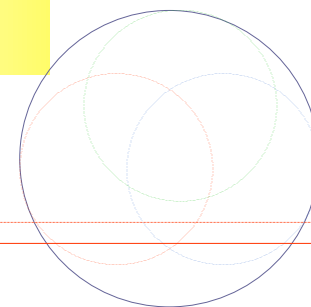
Jiaman.ma@nwpu.edu.cn

Review

概述

I/O系统硬件特点

I/O控制方式

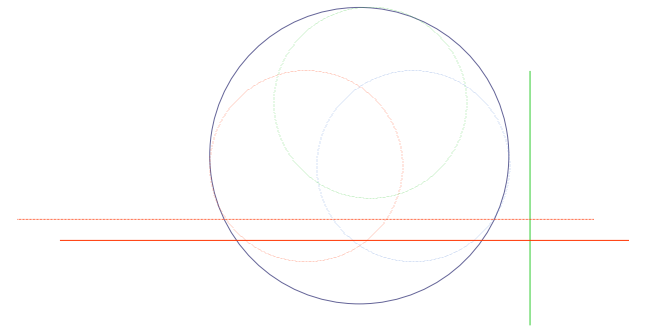


本次内容

I/O软件的组成

缓冲技术

其他技术

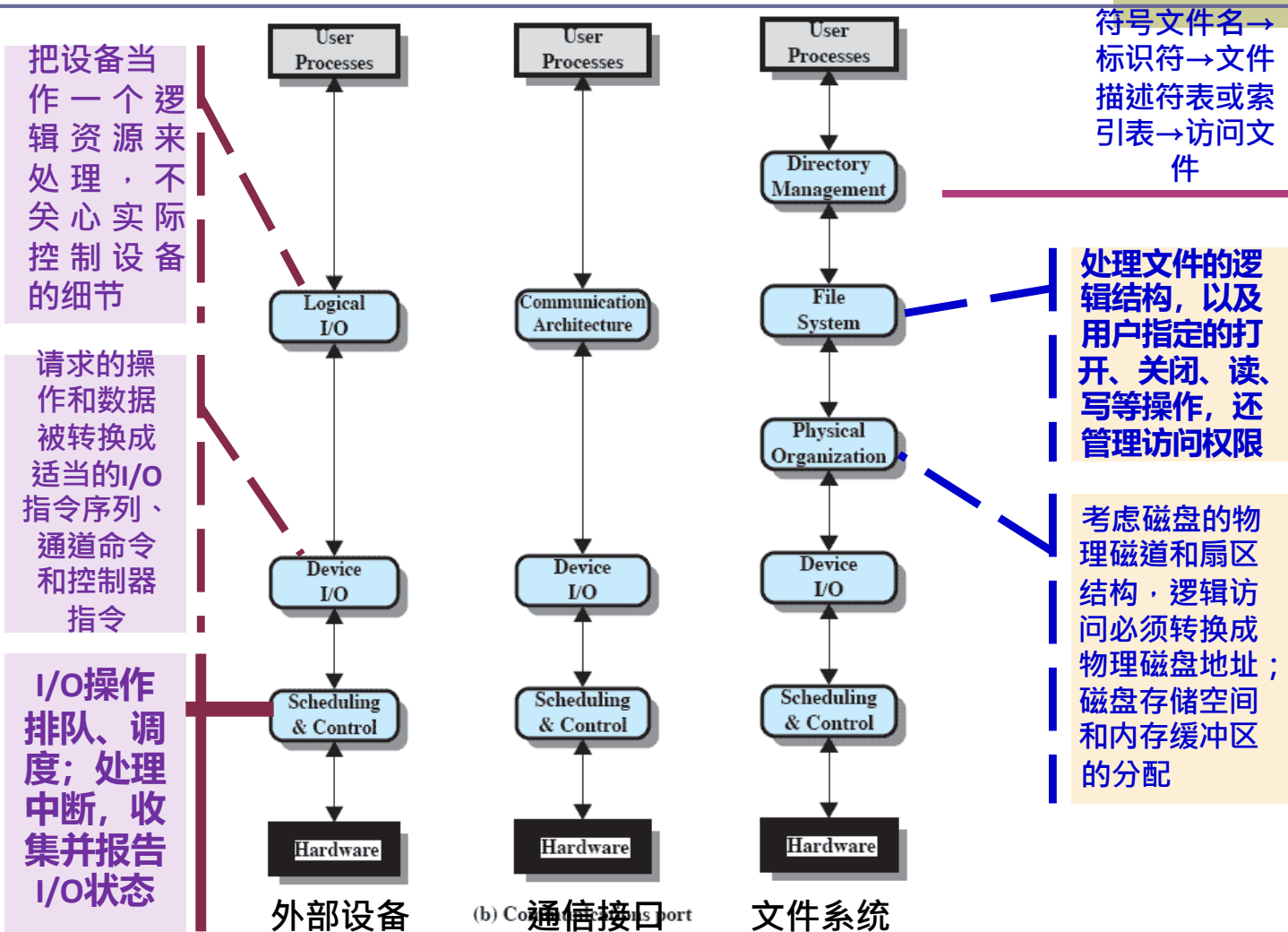


6.4 I/O系统的软件组成

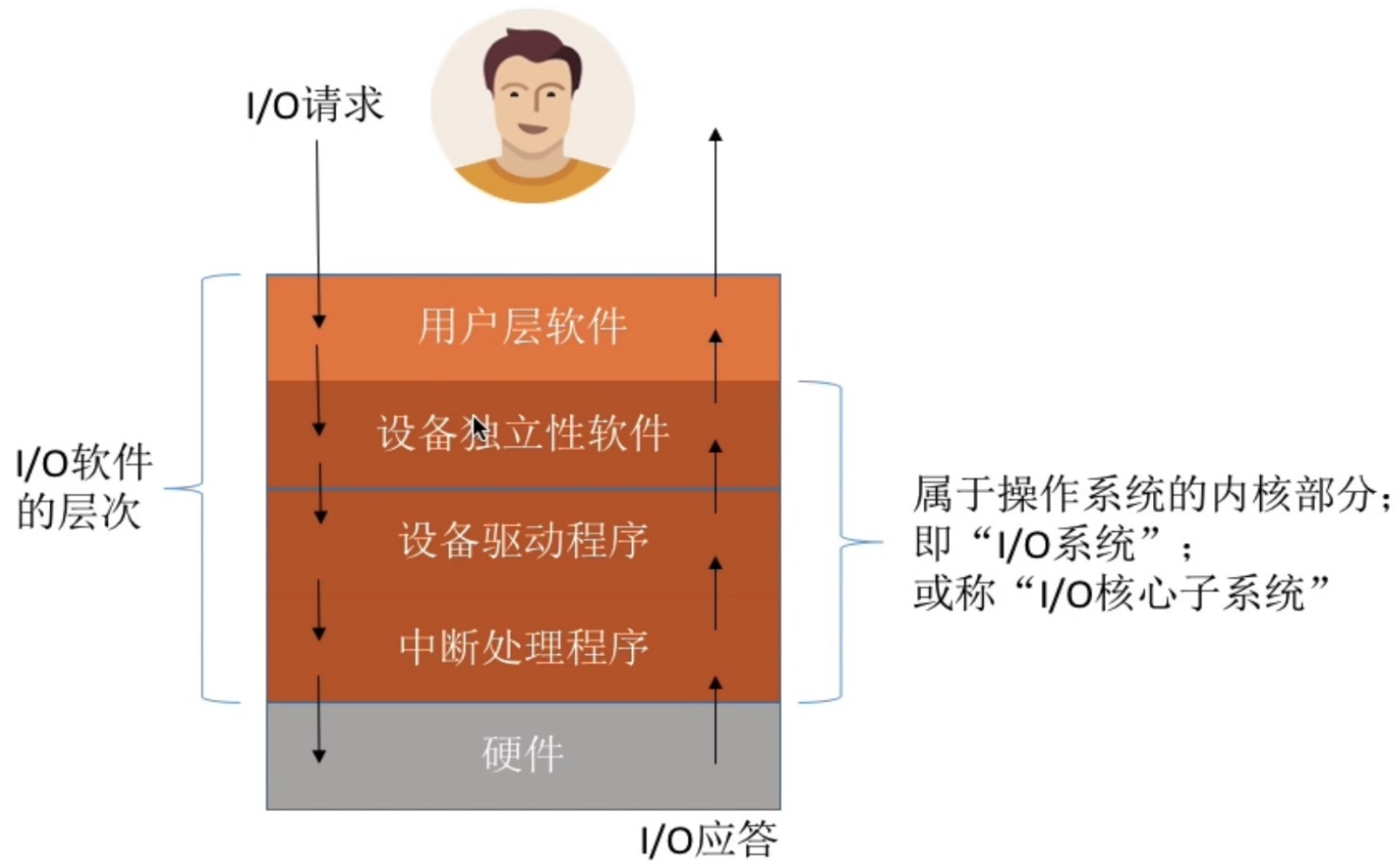
I/O软件设计 分层的设计思想

- 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，它又给高一层提供服务
- 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

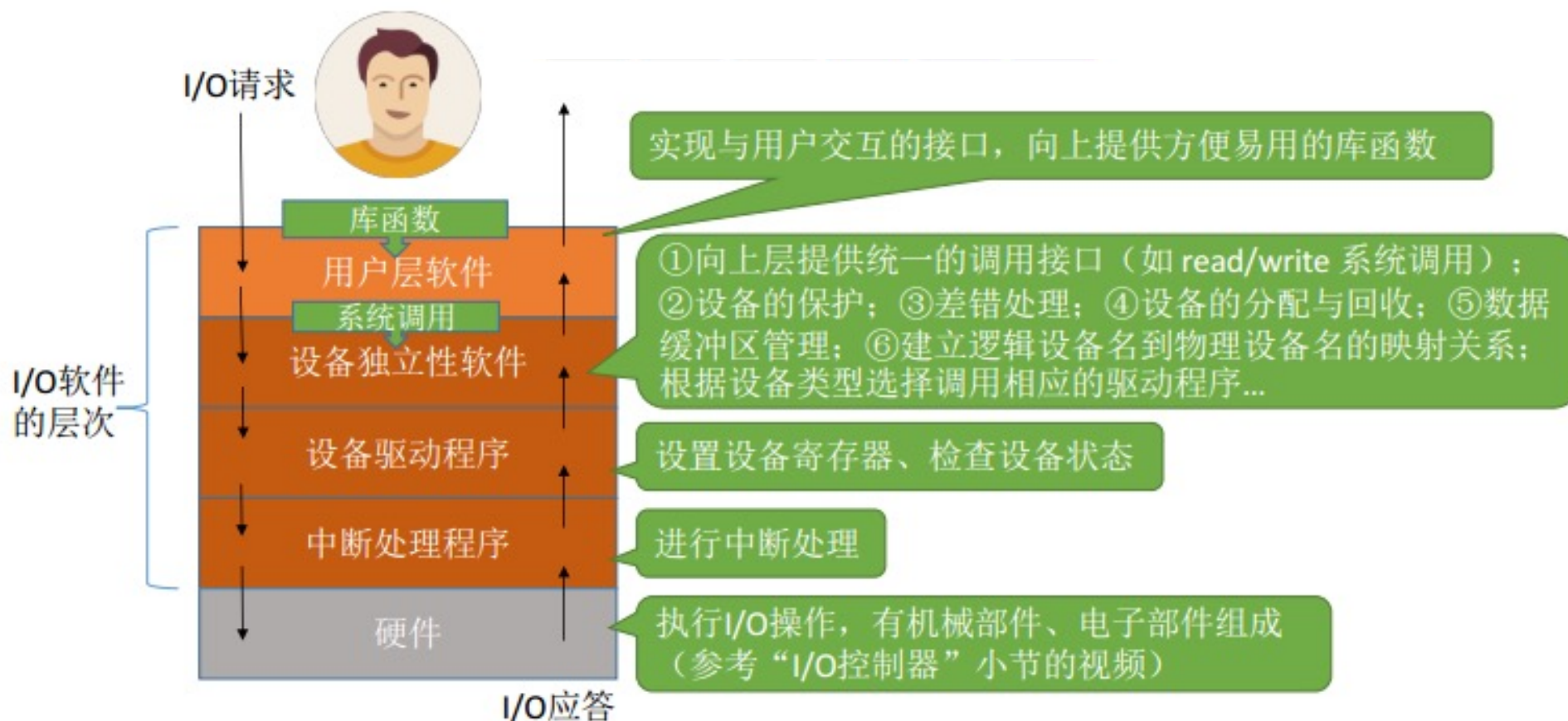
6.4 I/O系统的软件组成



I/O 软件层次划分

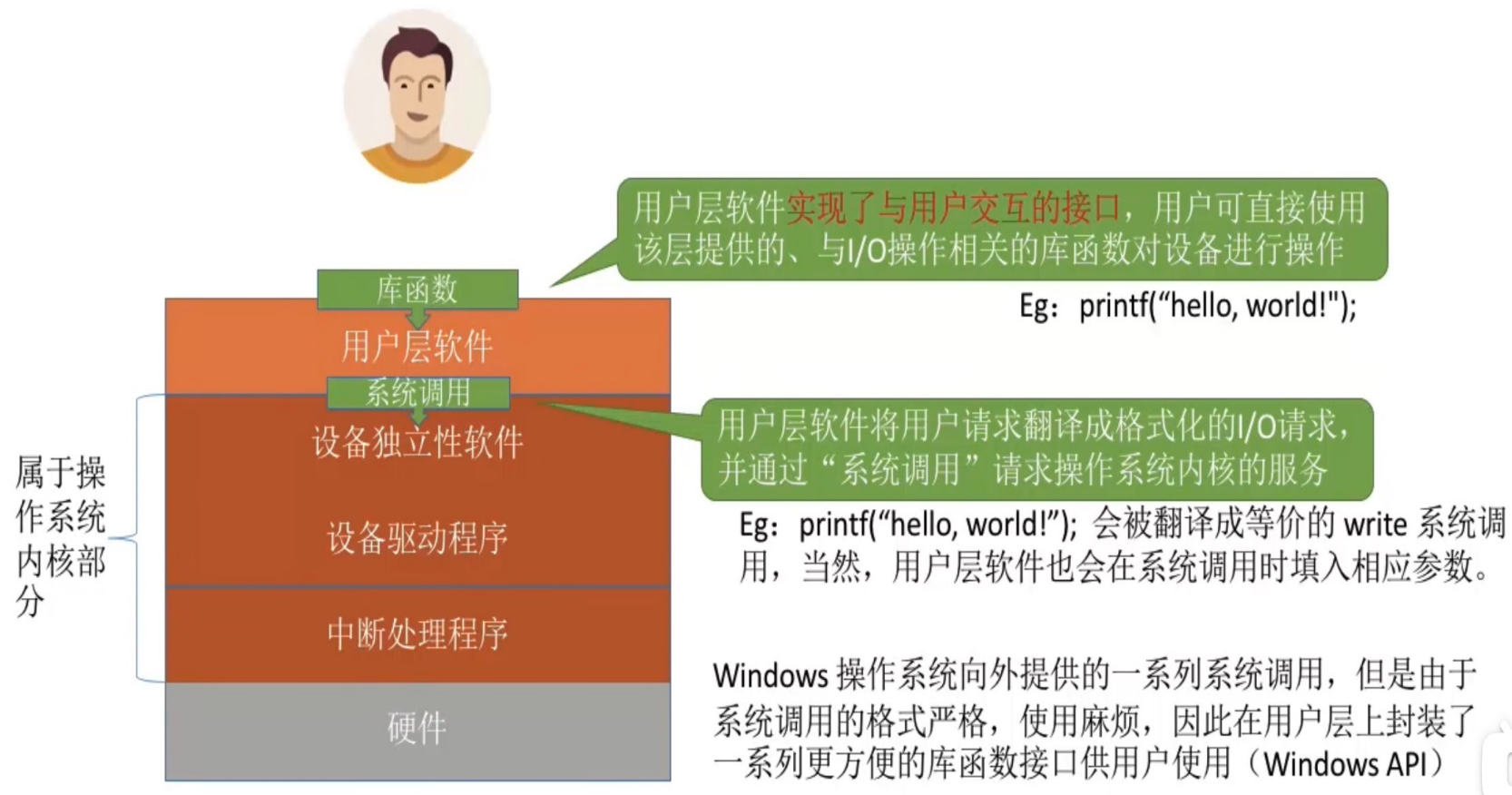


I/O 软件层次划分



理解并记住I/O软件各个层次之间的顺序，要能够推理判断某个处理应该是在哪个层次完成的（最常考的是设备独立性软件、设备驱动程序这两层。只需理解一个特点即可：直接涉及到硬件具体细节、且与中断无关的操作肯定是在设备驱动程序层完成的；没有涉及硬件的、对各种设备都需要进行的管理工作都

用户层软件实例



设备独立的软件

与设备驱动程序的统一接口

设备命名

设备保护

提供与设备无关的块尺寸

缓冲技术

设备分配

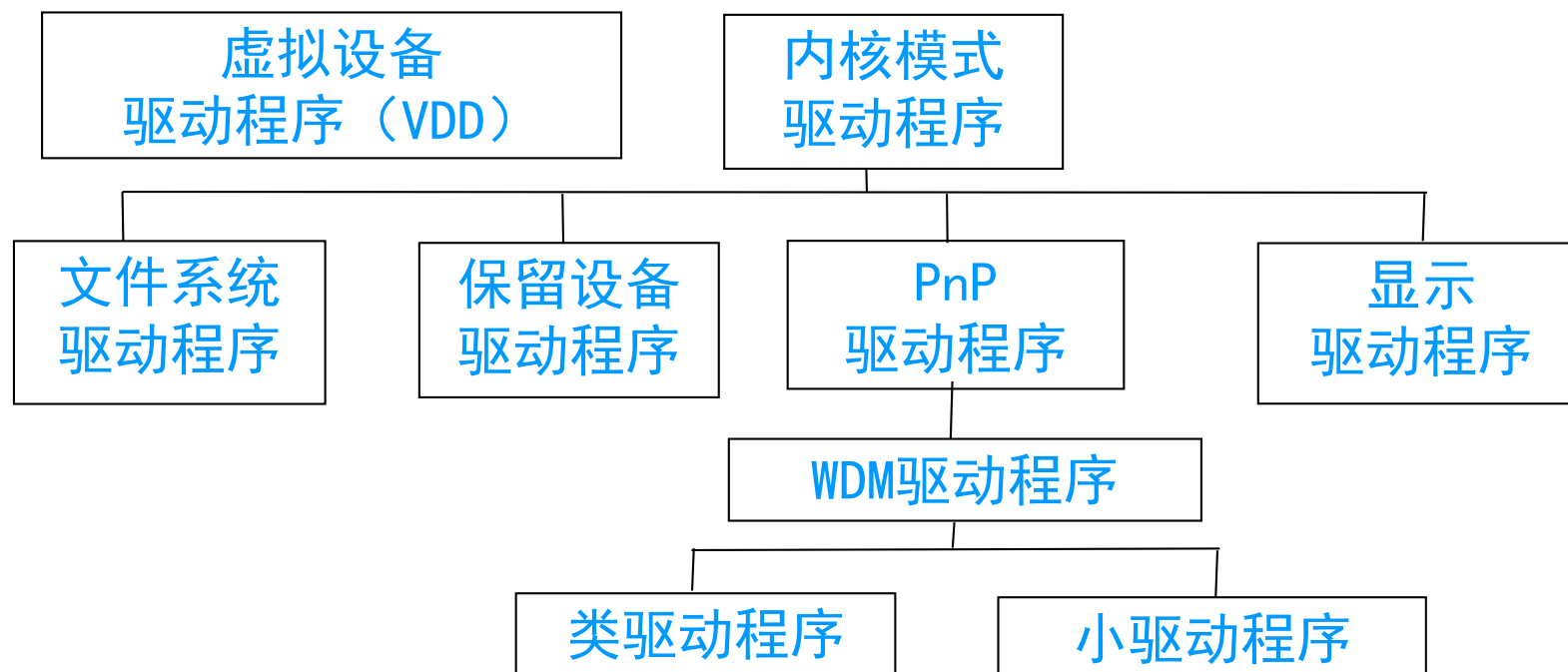
块设备的存储分配，独占设备的分配与释放

报告错误信息

设备无关性

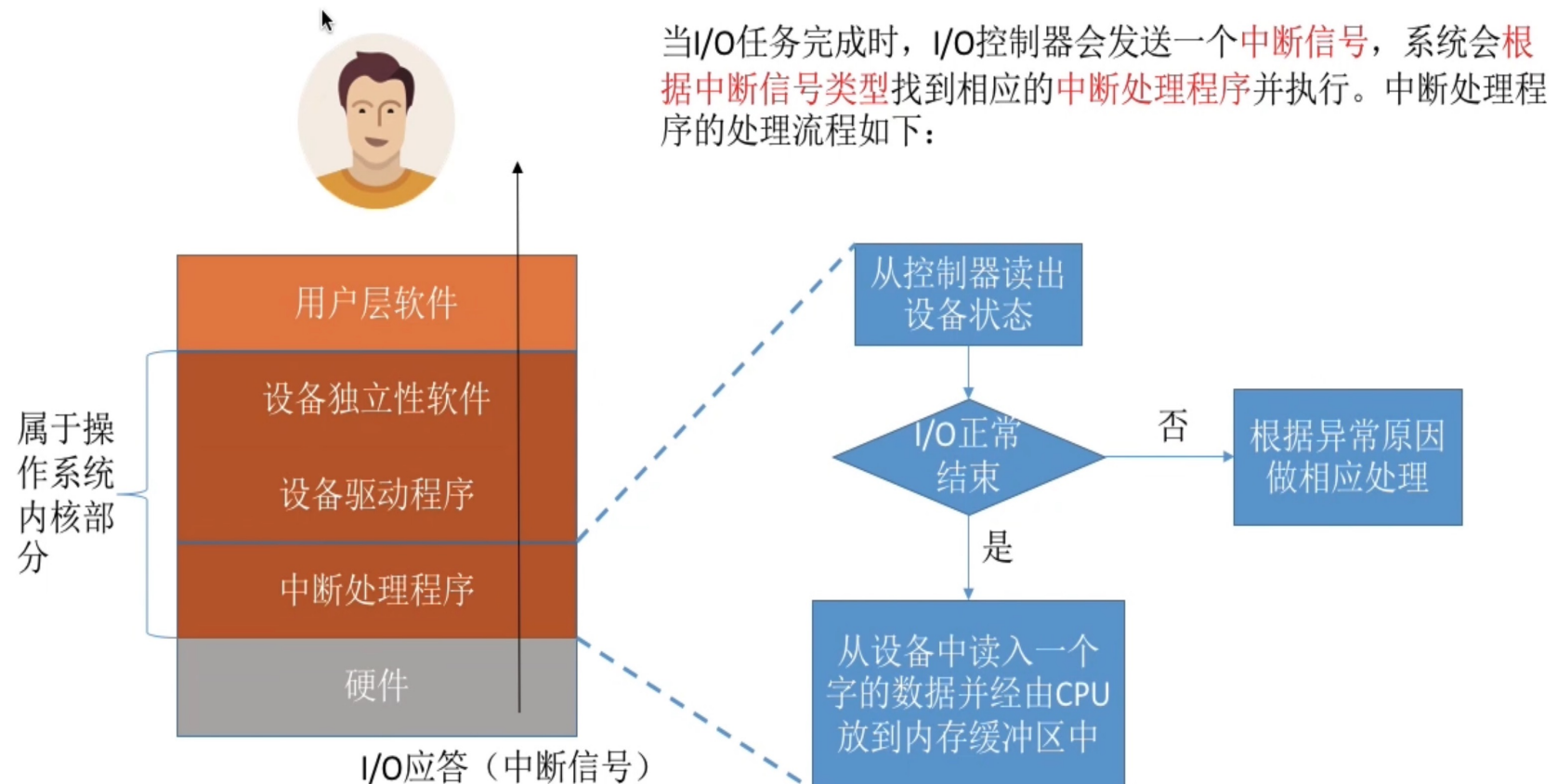
- 用户编写的程序可以访问任意 IO 设备，无需实现指定。
- 优势：实现设备分配的灵活性，重定向。
- 从用户角度：用户在编制程序时，使用逻辑设备名由系统实现从逻辑设备到物理设备的（实际设备）转换，并实施 I/O 操作
- 从系统角度：设计并实现 I/O 软件时，除了直接与设备打交道的低层软件之外，其他部分的软件不依赖于硬件

Windows 2000/XP的设备驱动程序



逻辑设备名	物理设备名	驱动程序入口地址
/dev/打印机1	3	1024
/dev/打印机2	5	2046
...

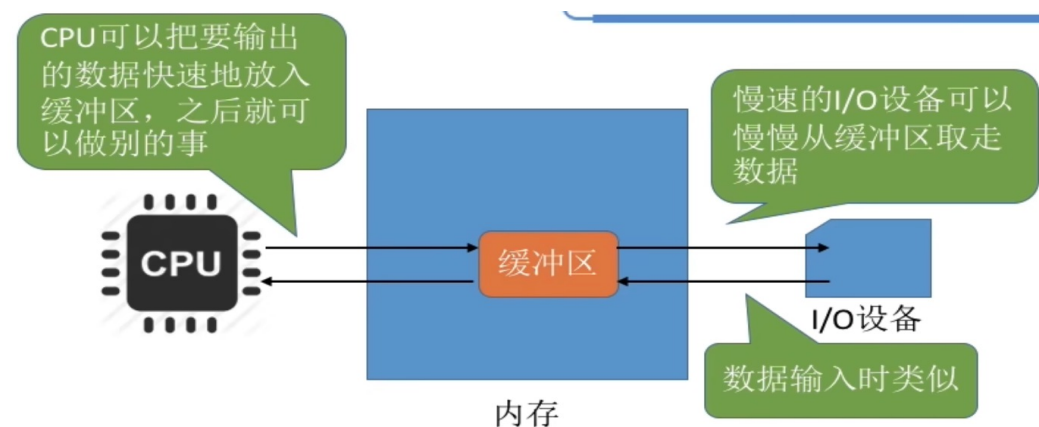
中断处理程序



6.5 缓冲技术

■ 缓冲的引入

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

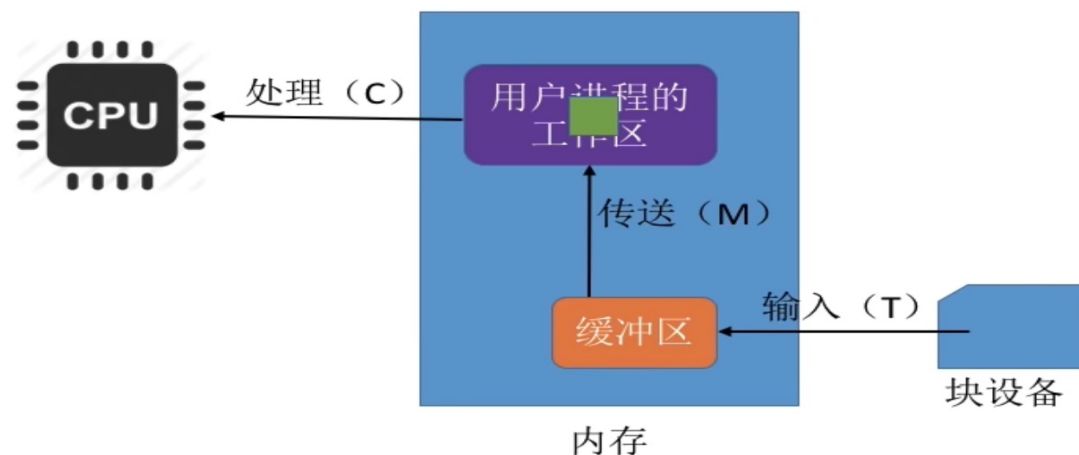


常用的缓冲技术

- **硬件缓冲器**：在设备控制器中有硬件缓冲器，通常容量较小
- **软件缓冲技术**：由缓冲区和对缓冲区的管理两部分组成
 - 1、单缓冲
 - 2、双缓冲
 - 3、环形缓冲
 - 4、缓冲池（多缓冲，循环缓冲）：统一管理多个缓冲区，采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使用

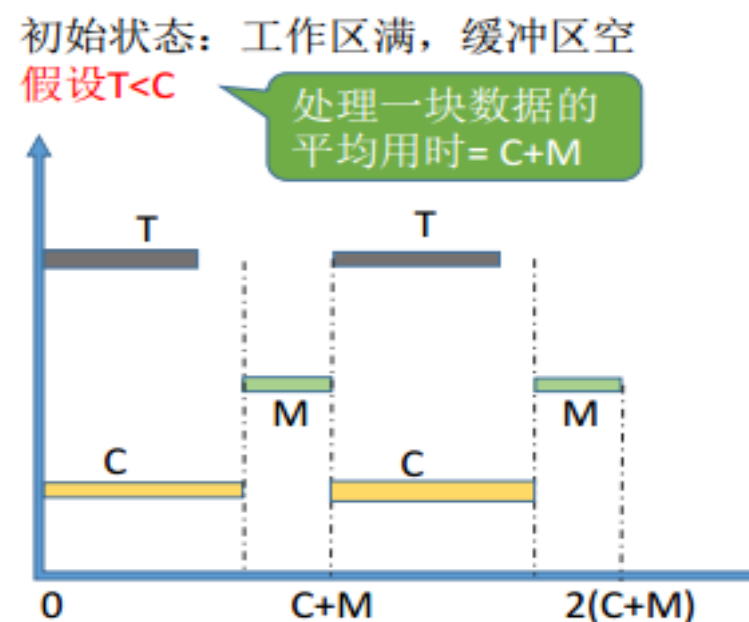
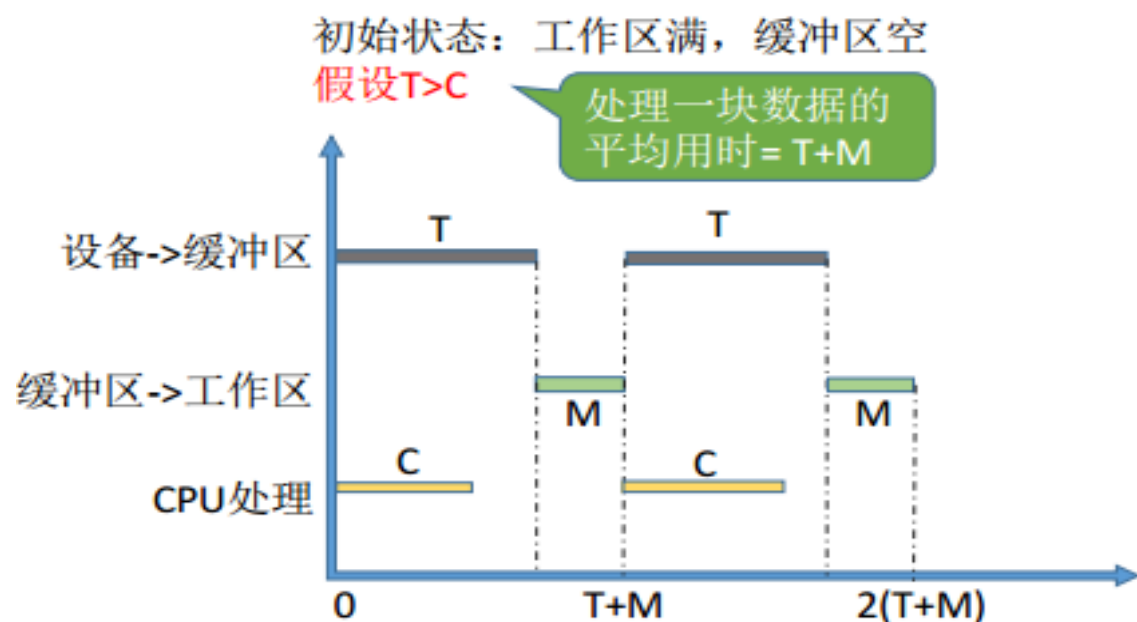
1、单缓冲

- **最简单的一种缓冲形式。**当进程发出一I/O请求时，OS为之分配一缓冲区。
- **对于输入：**设备先将数据送入缓冲区，OS再将数据传给进程。
- **对于输出：**进程先将数据传入缓冲区，OS再将数据送出到设备。
- **思考：**单缓冲能加快进程的执行速度吗？



假设某用户进程请求某种块设备读入若干块的数据。若采用**单缓冲**的策略，操作系统会在**主存中**为其分配一个**缓冲区**（若题目中没有特别说明，一个缓冲区的大小就是一个块）。

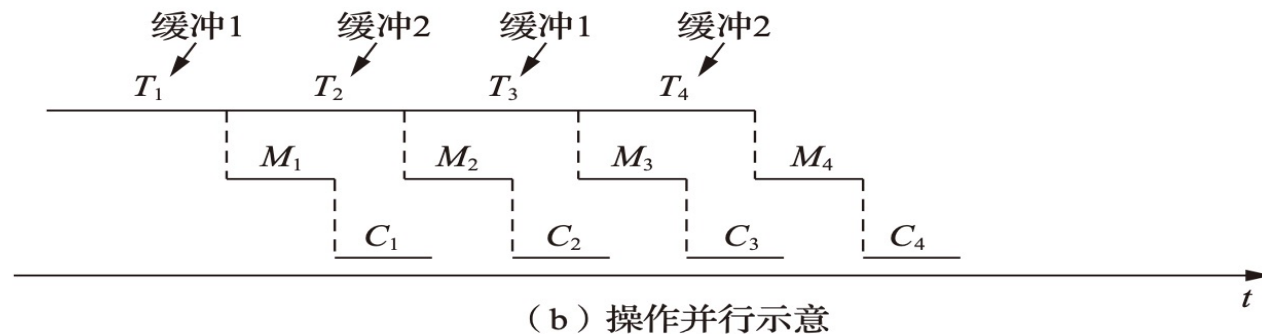
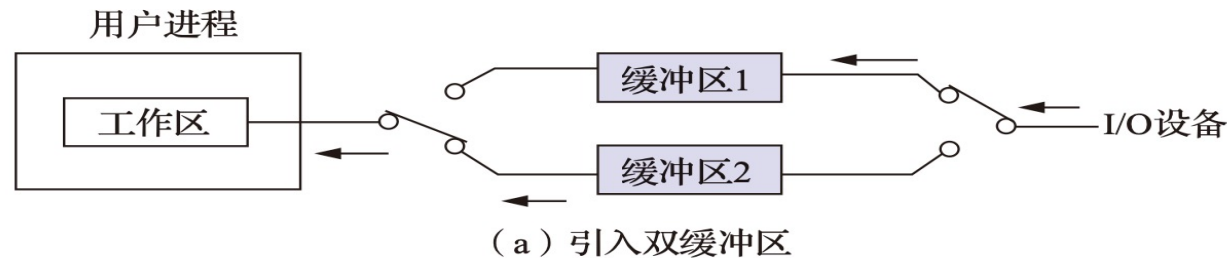
注意：当缓冲区数据非空时，不能往缓冲区冲入数据，只能从缓冲区把数据传出；当缓冲区为空时，可以往缓冲区冲入数据，但必须把缓冲区充满以后，才能从缓冲区把数据传出。



结论：采用单缓冲策略，处理一块数据平均耗时 $\text{Max}(C, T) + M$

2、双缓冲技术

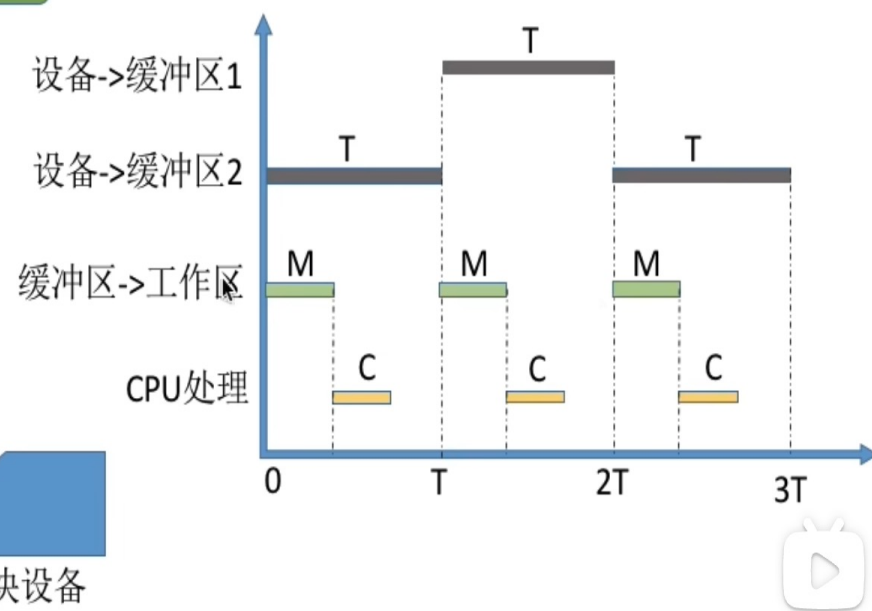
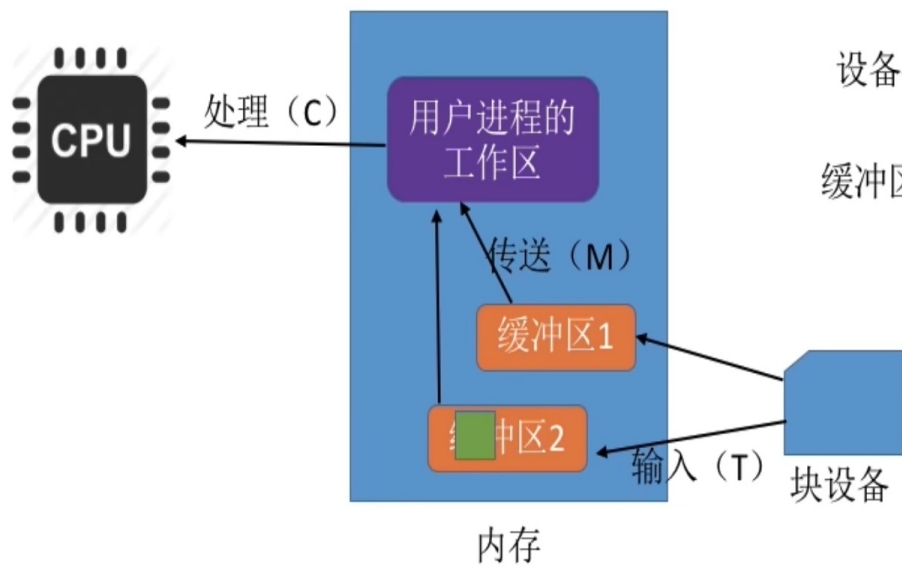
- 为了加快输入输出速度，引入双缓冲技术。
- 原理：设置两个缓冲区buf1和buf2。读入数据时，首先输入设备向buf1填入数据，然后进程从buf1提取数据，在进程从buf1提取数据的同时。输入设备向buf2中填数据。当buf1取空时，进程又从buf2中提取数据，与此同时输入设备向buf1填数。如此交替使用两个缓冲区，使CPU和设备的并行操作的程度进一步提高。



双缓冲题目中，假设初始状态为：工作区空，其中一个缓冲区满，另一个缓冲区空

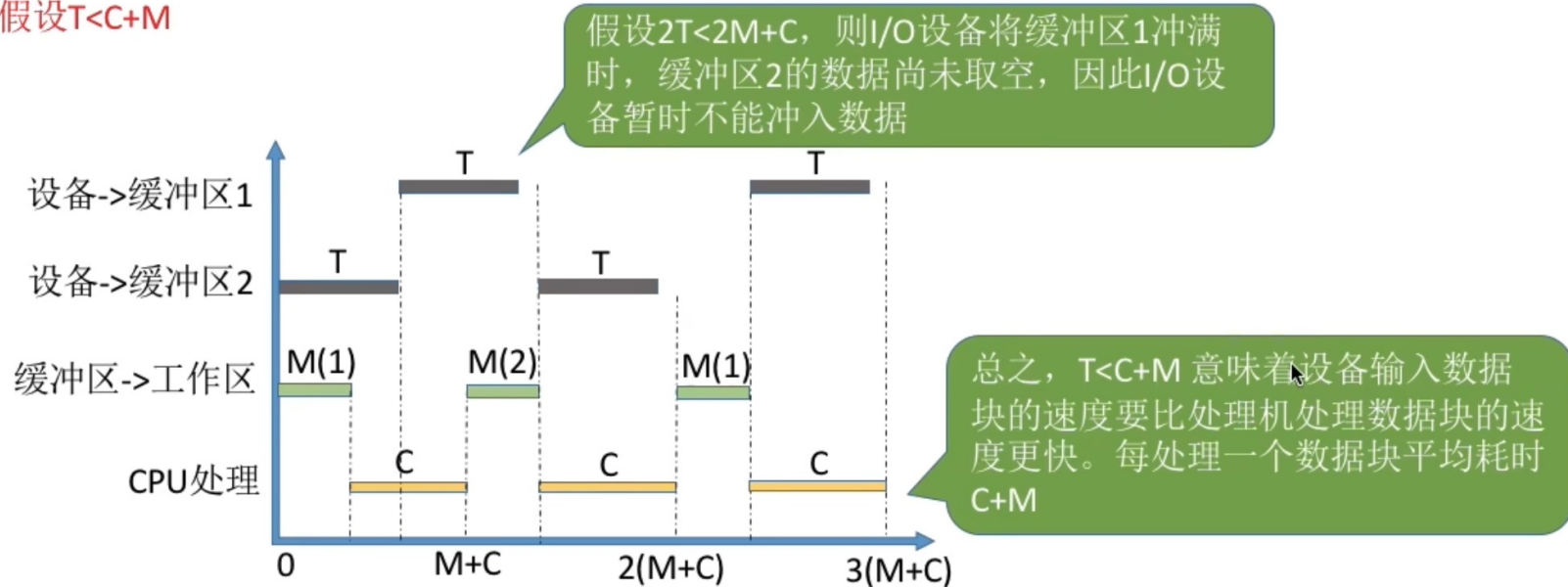
假设 $T > C + M$

处理一块数据的平均用时 = T



2、双缓冲技术

双缓冲题目中，假设初始状态为：工作区空，其中一个缓冲区满，另一个缓冲区空
假设 $T < C + M$

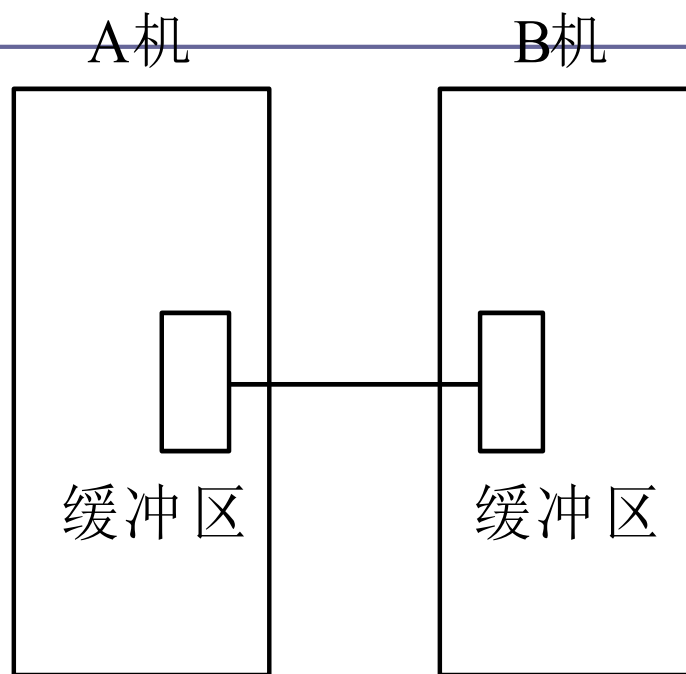


注：M(1) 表示“将缓冲区1中的数据传送到工作区”；M(2) 表示“将缓冲区2中的数据传送到工作区”

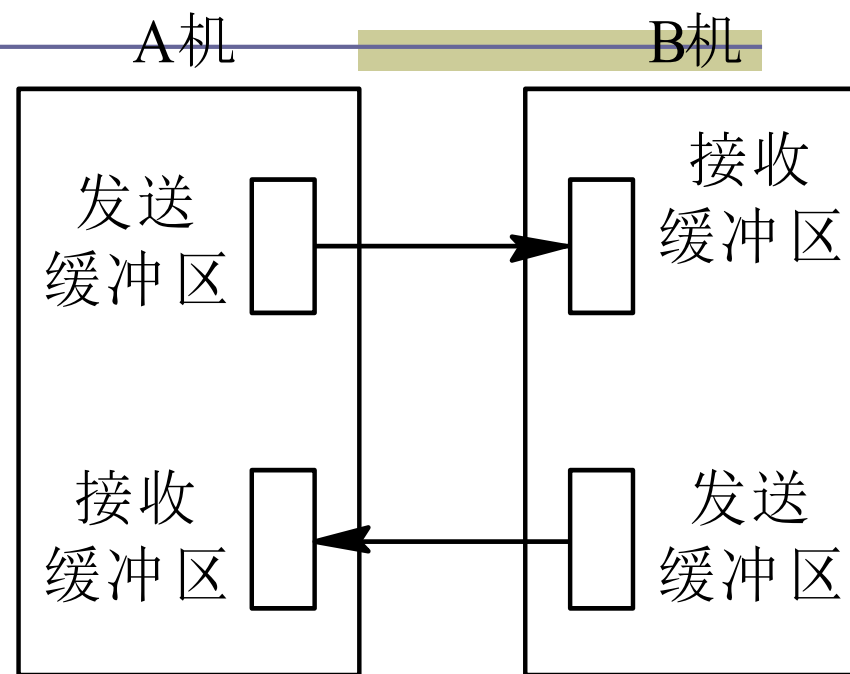
在双缓冲区情况下，系统处理一块数据的时间可以认为是 $\text{Max}(C + M, T)$

如果 $C + M < T$ ，则可使块设备连续输入；

如果 $C + M > T$ ，则可使CPU不必等待设备



(a) 单缓冲



(b) 双缓冲

显然，若两个相互通信的机器只设置单缓冲区，在任一时刻只能实现数据的单向传输。

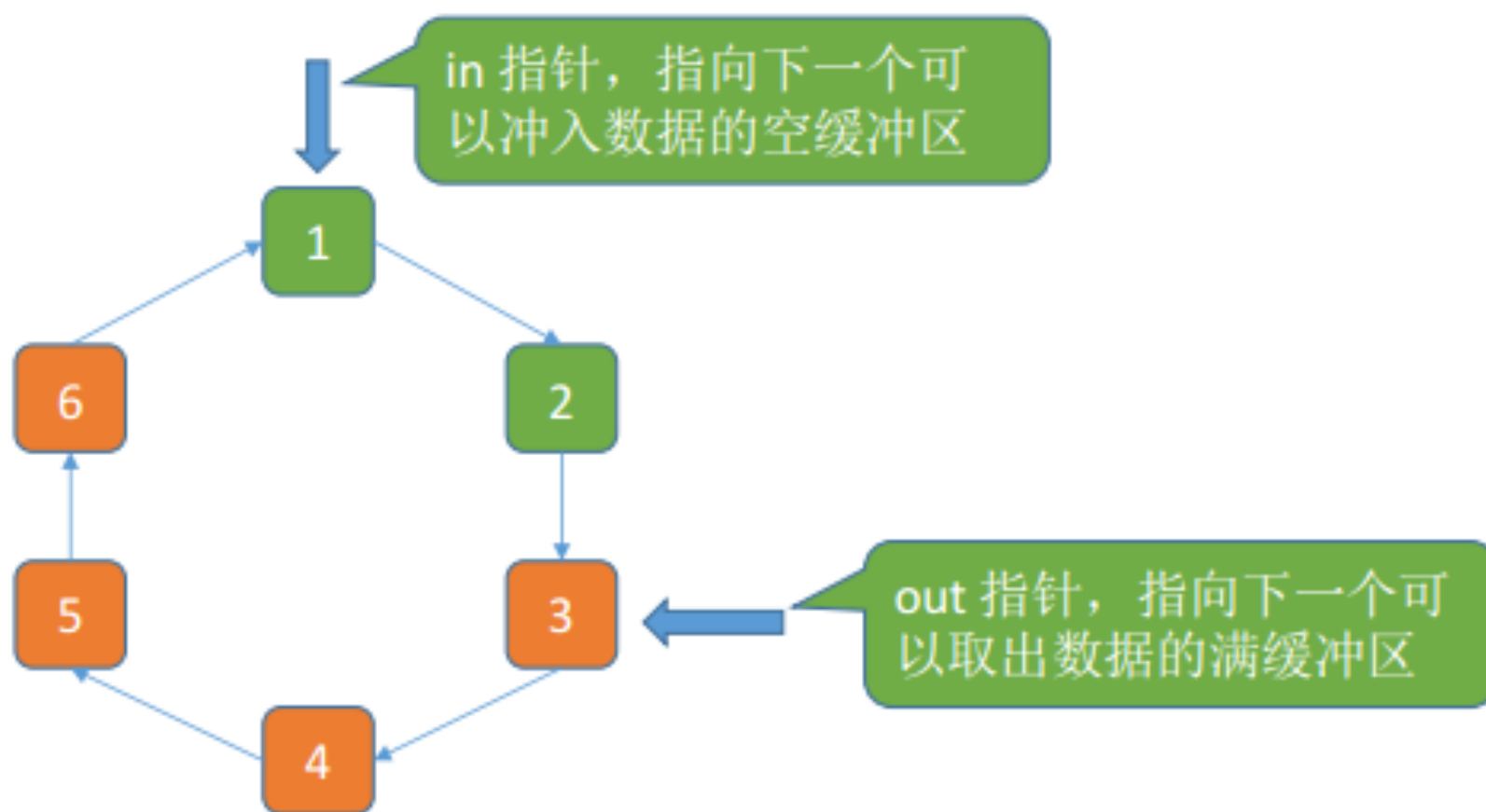
若两个相互通信的机器设置双缓冲区，则同一时刻可以实现双向的数据传输。

注：管道通信中的“管道”其实就是缓冲区。要实现数据的双向传输，必须设置两个管道

3、环形缓冲技术

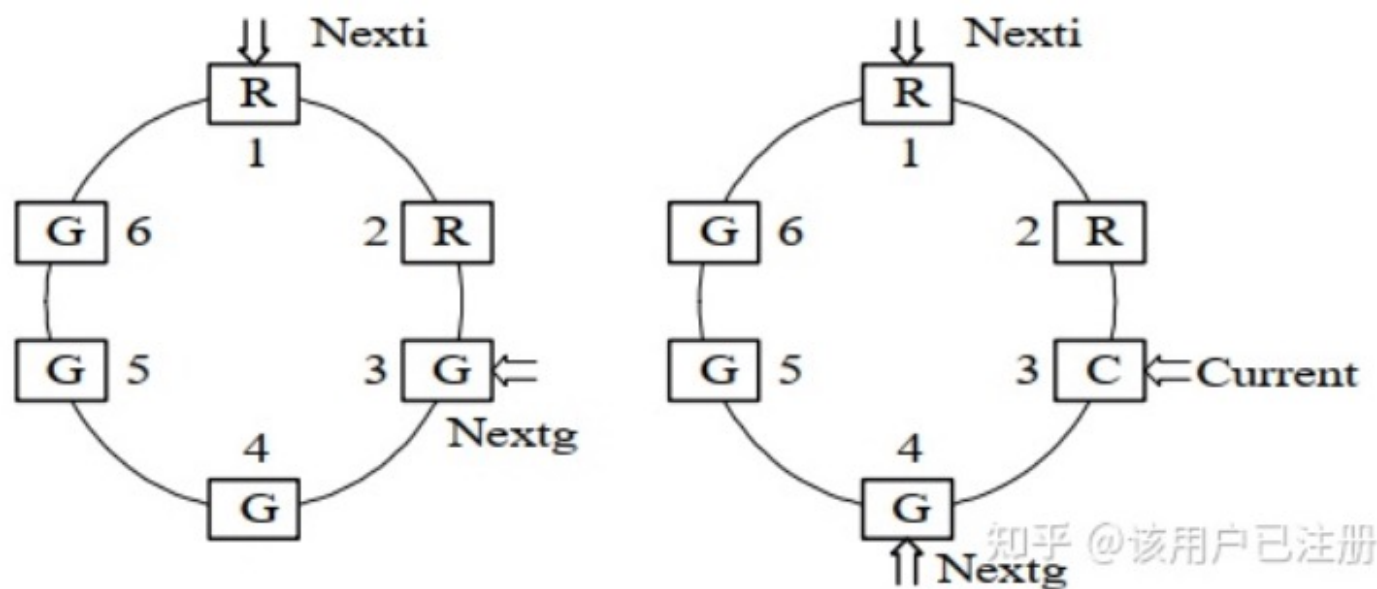
将多个大小相等的缓冲区链接成一个循环队列。

注：以下图示中，橙色表示已充满数据的缓冲区，绿色表示空缓冲区。



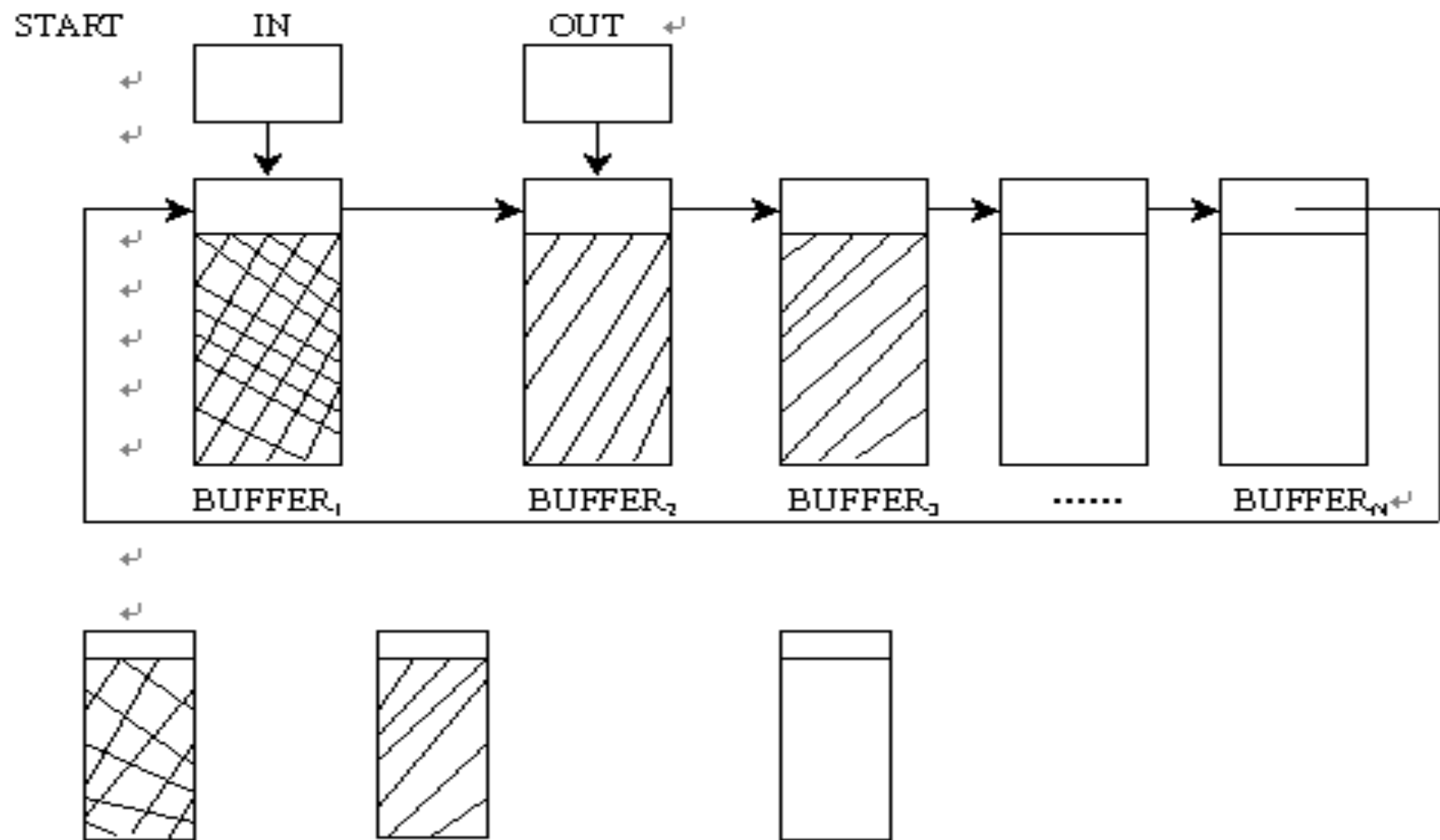
3、环形缓冲技术

- 多个缓冲区。在循环缓冲中包括多个缓冲区，其每个缓冲区的大小相同。作为输入的多缓冲区可分为三种类型：用于装输入数据的空缓冲区 R、已装满数据的缓冲区 G 以及计算进程正在使用的现行工作缓冲区 C，如下图所示。



循环缓冲

- 多个指针。作为输入的缓冲区可设置三个指针：用于指示计算进程下一个可用缓冲区 G 的指针 Nextg、指示输入进程下次可用的空缓冲区 R 的指针 Nexti，以及用于指示计算进程正在使用的缓冲区 C 的指针 Current。



数据已被取用又成为可用的缓冲区

装满数据,未取用的缓冲区

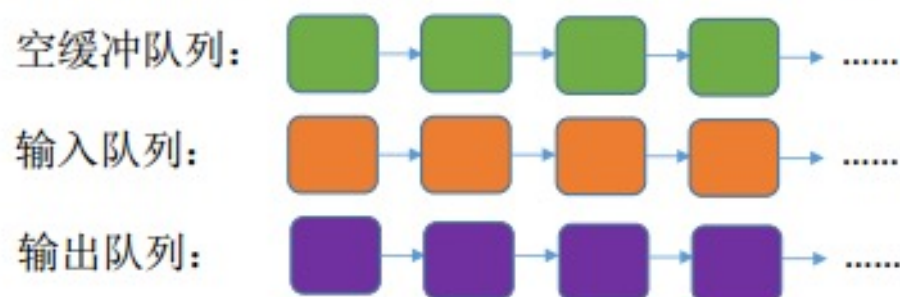
空闲缓冲区

4、缓冲池

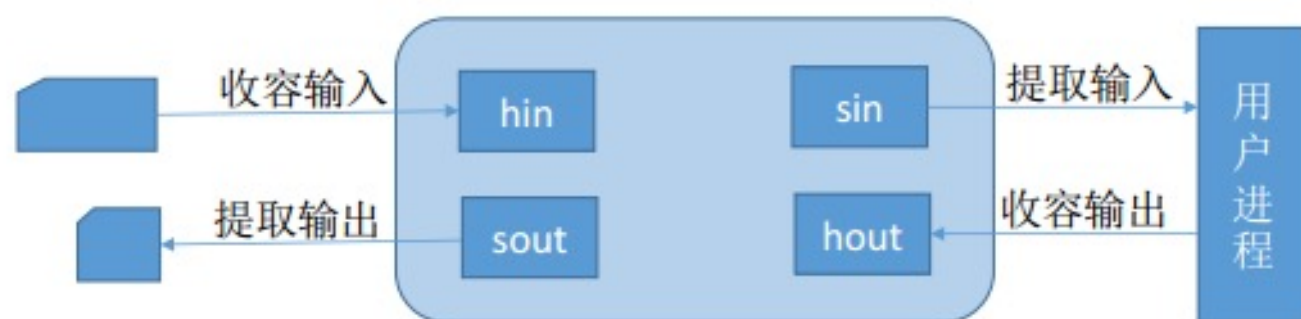
- 缓冲池由内存中一组大小相等的缓冲区组成
- 缓冲池属于系统资源，由系统进行管理
- 缓冲池中各缓冲区可根据需要组成各种缓冲区队列。
 - ① 空(闲)缓冲区；
 - ② 装满输入数据的缓冲区；
 - ③ 装满输出数据的缓冲区。

缓冲池由系统中共用的缓冲区组成。这些缓冲区按使用状况可以分为：空缓冲队列、装满输入数据的缓冲队列（输入队列）、装满输出数据的缓冲队列（输出队列）。

另外，根据一个缓冲区在实际运算中扮演的功能不同，又设置了四种工作缓冲区：用于收容输入数据的工作缓冲区（**hin**）、用于提取输入数据的工作缓冲区（**sin**）、用于收容输出数据的工作缓冲区（**hout**）、用于提取输出数据的工作缓冲区（**sout**）

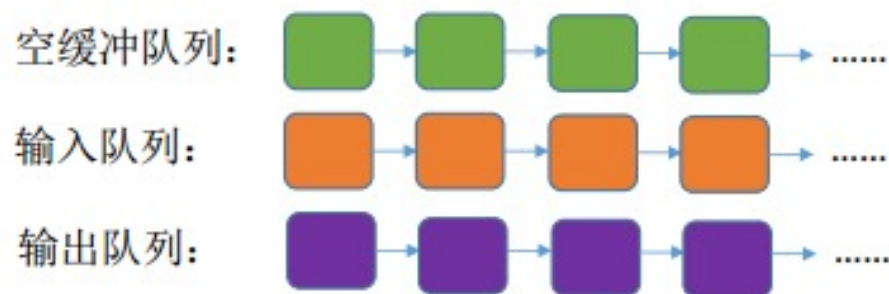


①输入进程请求输入数据

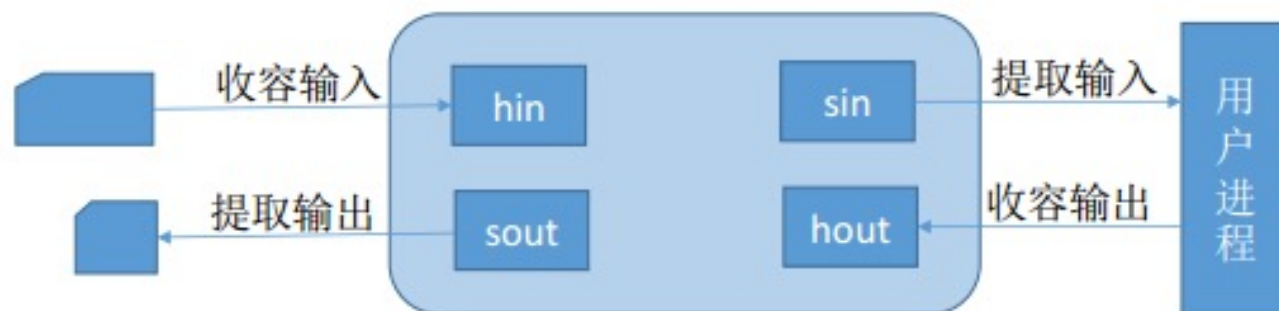


从空缓冲队列中取出一块作为收容输入数据的工作缓冲区（**hin**）。冲满数据后将缓冲区挂到输入队列队尾

缓冲池由系统中共用的缓冲区组成。这些缓冲区按使用状况可以分为：空缓冲队列、装满输入数据的缓冲队列（输入队列）、装满输出数据的缓冲队列（输出队列）。
另外，根据一个缓冲区在实际运算中扮演的功能不同，又设置了四种工作缓冲区：用于收容输入数据的工作缓冲区（hin）、用于提取输入数据的工作缓冲区（sin）、用于收容输出数据的工作缓冲区（hout）、用于提取输出数据的工作缓冲区（sout）



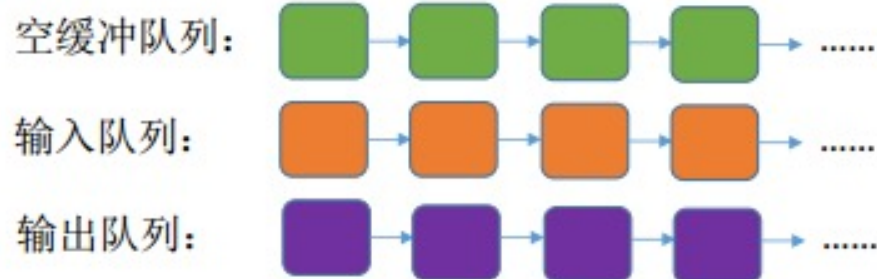
- ①输入进程请求输入数据
- ②计算进程想要取得一块输入数据



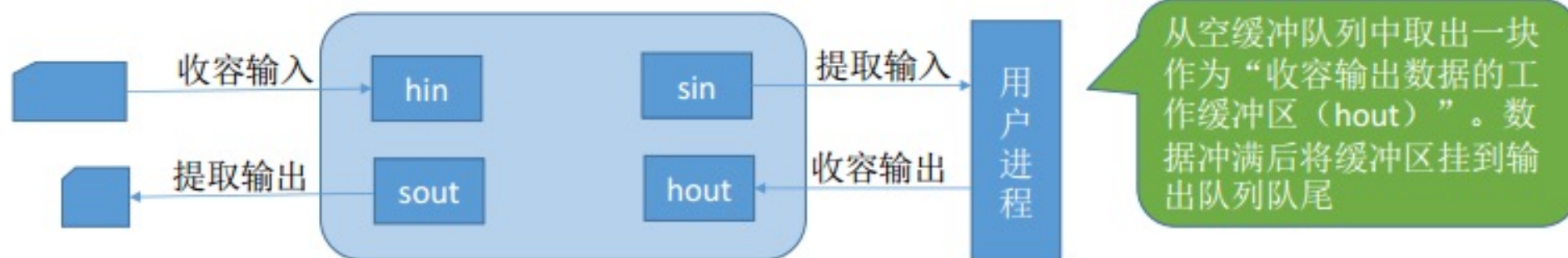
从输入队列中取得一块充满输入数据的缓冲区作为“提取输入数据的工作缓冲区（sin）”。缓冲区读空后挂到空缓冲区队列

缓冲池由系统中共用的缓冲区组成。这些缓冲区按使用状况可以分为：空缓冲队列、装满输入数据的缓冲队列（输入队列）、装满输出数据的缓冲队列（输出队列）。

另外，根据一个缓冲区在实际运算中扮演的功能不同，又设置了四种工作缓冲区：用于收容输入数据的工作缓冲区（hin）、用于提取输入数据的工作缓冲区（sin）、用于收容输出数据的工作缓冲区（hout）、用于提取输出数据的工作缓冲区（sout）

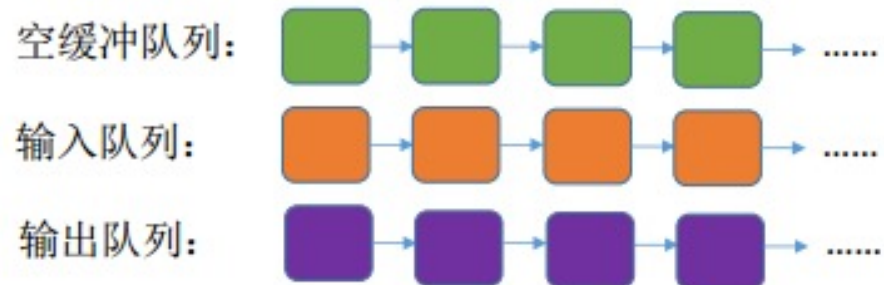


- ①输入进程请求输入数据
- ②计算进程想要取得一块输入数据
- ③计算进程想要将准备好的数据冲入缓冲区

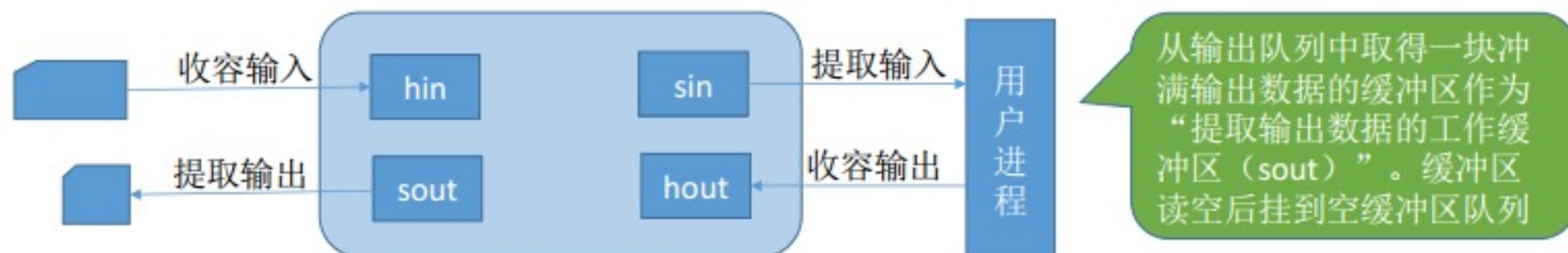


缓冲池由系统中共用的缓冲区组成。这些缓冲区按使用状况可以分为：空缓冲队列、装满输入数据的缓冲队列（输入队列）、装满输出数据的缓冲队列（输出队列）。

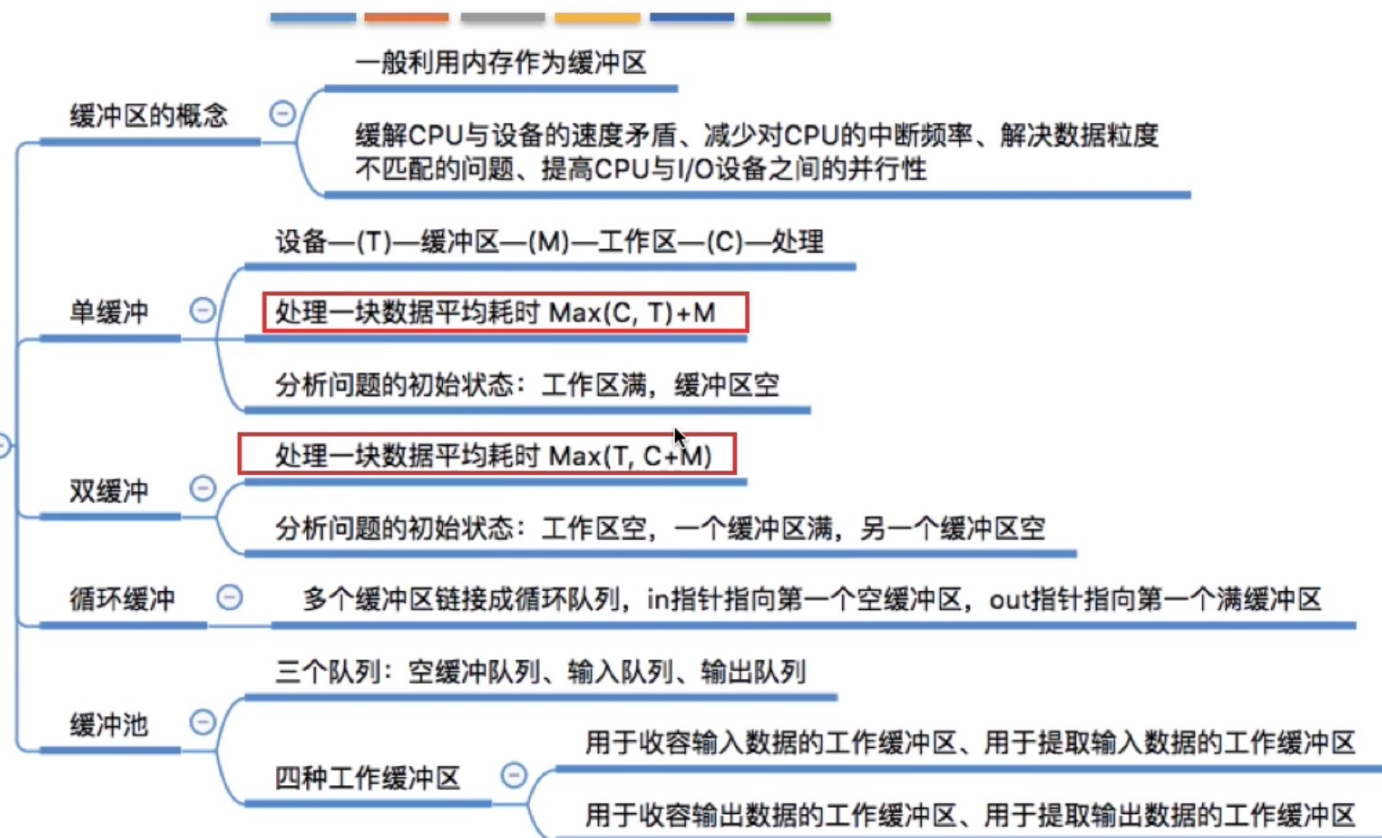
另外，根据一个缓冲区在实际运算中扮演的功能不同，又设置了四种工作缓冲区：用于收容输入数据的工作缓冲区（hin）、用于提取输入数据的工作缓冲区（sin）、用于收容输出数据的工作缓冲区（hout）、用于提取输出数据的工作缓冲区（sout）



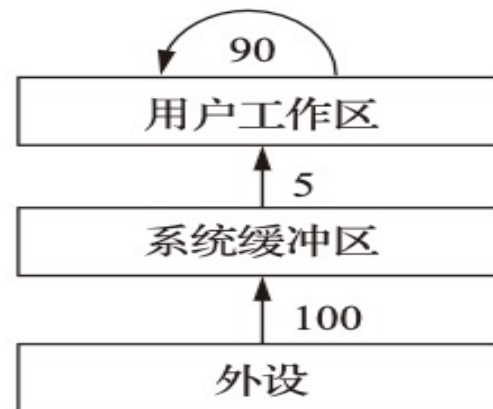
- ①输入进程请求输入数据
- ②计算进程想要取得一块输入数据
- ③计算进程想要将准备好的数据冲入缓冲区
- ④输出进程请求输出数据



缓冲区管理



- 设系统缓冲区和用户工作区均采用单缓冲区，从外设读入1个数据块到系统缓冲区的时间为100，从系统缓冲区读入1个数据块到用户工作区的时间为5，对用户工作区中的1个数据块进行分析的时间为90。进程从外设读入并分析2个数据块的最短时间是多少？



- 数据块1从外设到用户工作区的总时间为 105，在这段时间内，数据块2没有进行操作。在数据块1进行分析处理时，数据块2从外设到用户工作区的总时间为 105，在这段时间内两个过程是并行的。加上处理数据块2的时间90，总共是 $105+105+90=300$ 。

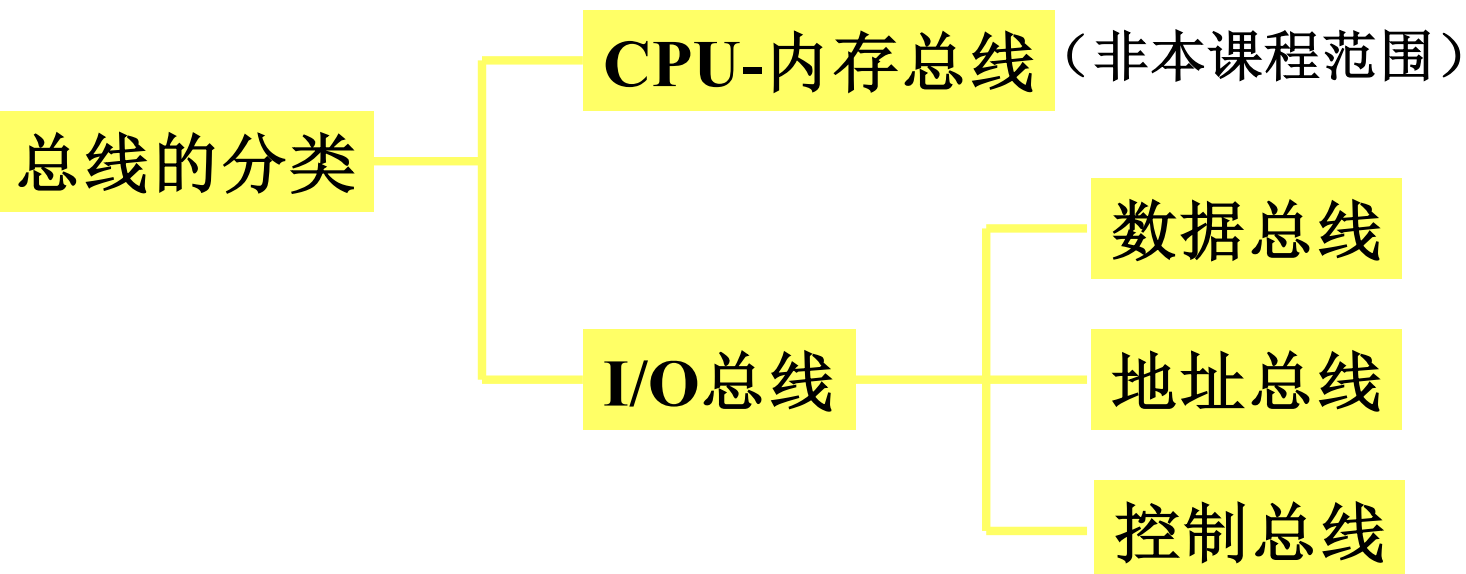
- 假定把磁盘上一个数据块中的信息输入一单缓冲区的时间 T 为 $100\mu\text{s}$ ，将缓冲区中的数据传送到用户区的时间 M 为 $50\mu\text{s}$ ，CPU 对这一块数据进行计算的时间 C 为 $50\mu\text{s}$ 。请问，系统对一块数据的处理时间为多少？
- 如果将单缓冲区改为双缓冲区，则系统对一块数据的处理时间为多少？

- 单缓冲区处理一块数据的时间为 $\max(C, T) + M = \max(50\mu s, 100\mu s) + 50\mu s = 150\mu s$;
- 双缓冲区处理一块数据的时间为 $\max(C + M, T) = \max(50\mu s + 50\mu s, 100\mu s) = 100\mu s$ 。

6.6 其它技术

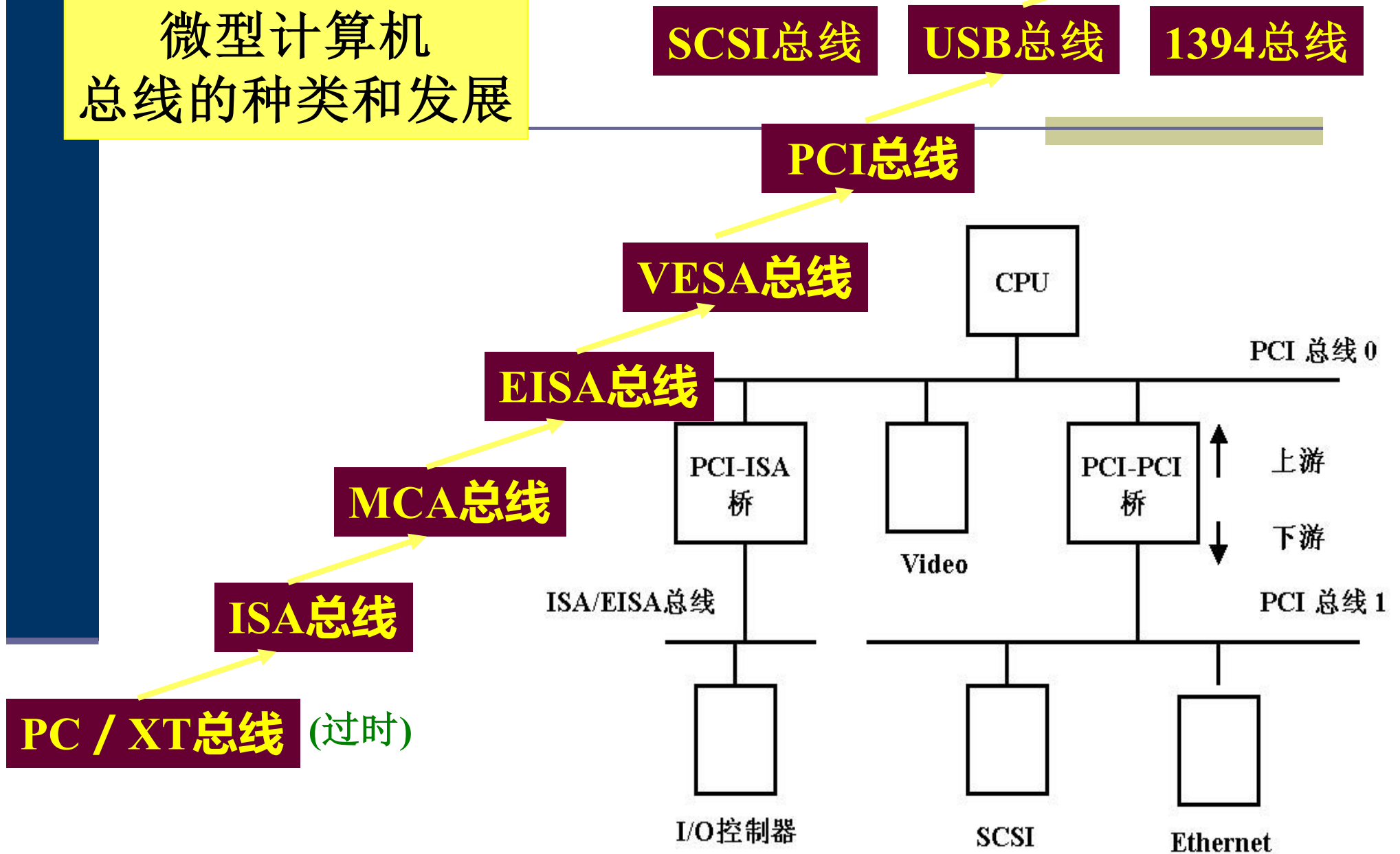
- 总线技术
- USB技术
- SCSI接口技术

6.6.1 总线技术



在计算机系统内各种子系统，如CPU、内存、I/O设备等之间，构建公用的信号或数据传输通道，这种可共享的传输通道称为总线

微型计算机 总线的种类和发展



ISA（工业标准结构）

- ISA基于PC / AT总线，是由IEEE（美国电气电子工程师协会）1987年正式确立的标准。
- ISA槽是一个黑色的62 + 36线插槽。
- ISA工作频率定在8.33MHz，数据传输率为8.33MB / s。
- 随着系统工作频率的迅速提高，其配用的扩展卡也逐渐被淘汰，现在最新的主板已开始取消ISA槽。

PCI（外围部件互连）

- 1993年Intel发表PCI2.0版，PCI开始走进主板。
- PCI有32位和64位两种，32位PCI槽124线，64位槽188线，目前常用的是32位插槽。
- PCI槽的时钟频率为33.3MHz，32位PCI的数据传输率为133MB / s，大大高于ISA。所以PCI问世后迅速成了扩展总线的主流，流行的扩展卡也都转移到PCI上，如显示卡、声卡、网卡、MODEM卡等等。

AGP (加速图形端口)

- 1996年Intel公司在PCI的基础上专为显示卡接口提出AGP标准。
- AGP使用32位数据总线，工作频率为66.6MHz
- AGP 1x的数据传输率可达266MB/s，AGP 2x在一个时钟周期的上升沿和下降沿各传输一次资料，其数据传输率可达到533MB / s，而AGP 4x的理论传输率为1.066GB / s。

IEEE1394

- IEEE1394是1995年由IEEE将APPLE公司高速串行总线 “FIRE WIRE”标准化而成，目前还在发展中。
- IEEE1394适用于声音、图像和视频多媒体产品、高速打印机和扫描仪产品、硬盘等存储设备、数码摄影机、显示器和影音录放设备等。
- 标准数据传输率分三种：100Mbps、200Mbps和400Mbps, IEEE1394商业联盟计划将它提高到800Mbps、1Gbps和1.6Gbps;
- 支持同步模式传输,可实现 “准实时” 的多媒体数据传输;

6.6.2 USB技术

- USB (Universal Serial Bus) 通用串行总线
- 适用于低、中速的外围设备
- USB的传输方式
 - (1) 等时传输方式
 - (2) 中断传输方式
 - (3) 控制传输方式
 - (4) 批传输方式

USB 的特点

- 数据传输具有1.5Mbps和12Mbps两种方式;
- 连接方便, 易于扩展, 可使用集线器进行树形连接, 设备最多可达6层127个, 支持热插拔;
- 连接的设备之间不是平等关系而是亲子关系, 上下游的关系明确, 对上和对下的电缆插头不一样, 各个分设备只能同主设备进行通信并受主设备的控制;
- 单根线缆最长为5米;
- **USB 2.0规范将最高速率提高到480 Mbps**

6.6.3 SCSI接口技术

- 小型计算机系统接口 (Small Computer System Interface)
- IDE与SCSI二者的区别主要在于：
 - IDE的工作方式需要CPU的全程参与
 - SCSI接口则完全通过独立的高速的SCSI卡来控制数据的读写操作

优缺点

SCSI接口优点：

1. 适应面广，在一块SCSI控制卡上就可以同时挂接15个设备
2. 高性能（具有很多任务、宽带宽及少CPU占用率等特点）
3. 具有外置和内置两种

SCSI接口缺点：

价格昂贵、安装复杂



That's all.



Thank you very much!