

引入缓冲技术解决什么问题？

操作系统中最早引入的技术

→ 解决CPU与I/O设备之间速度的不匹配问题

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

中断次数，而放宽 CPU 对中断响应时间的要求。



缓冲技术实现

● 缓冲区分类

硬缓冲：由硬件寄存器实现（例如：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）：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



采用的是有界缓冲区的生产者 / 消费者这样一个模型对缓冲区进行相应的管理。

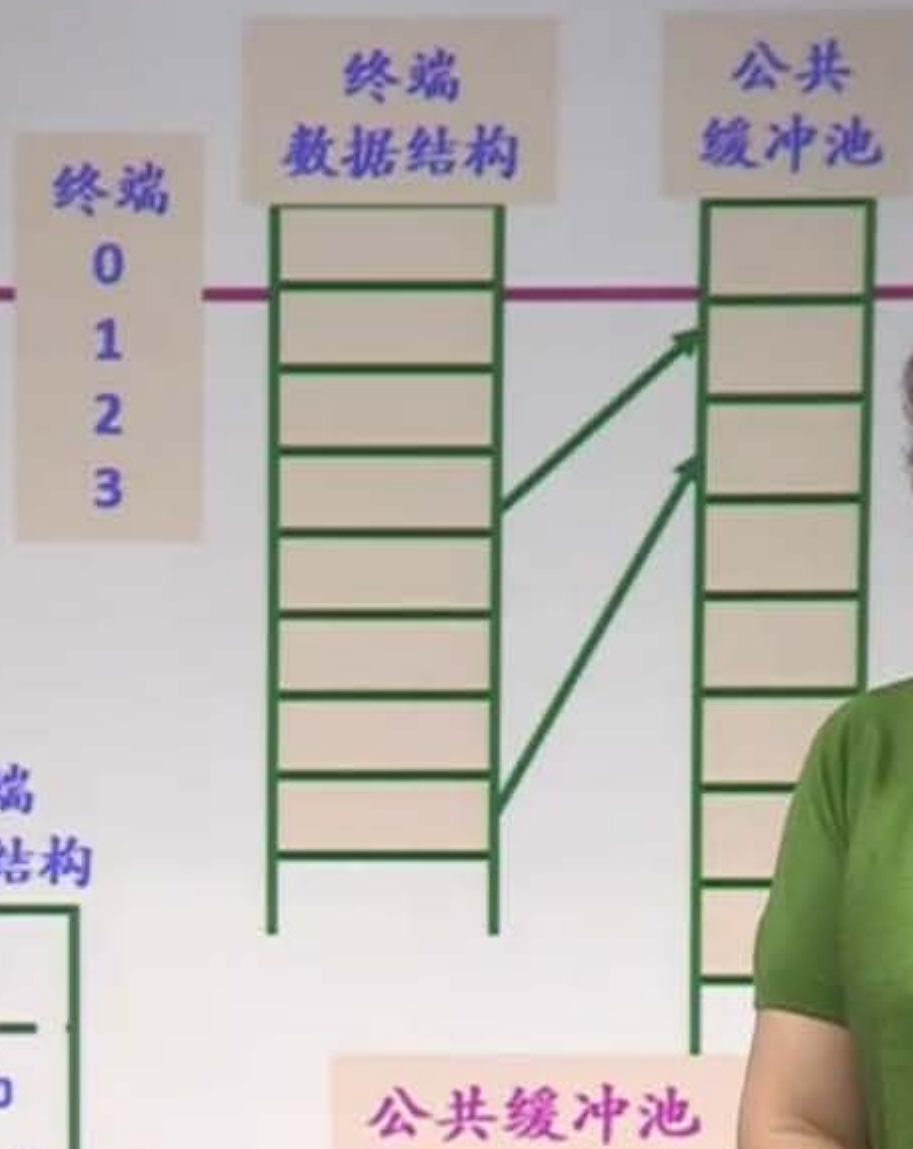
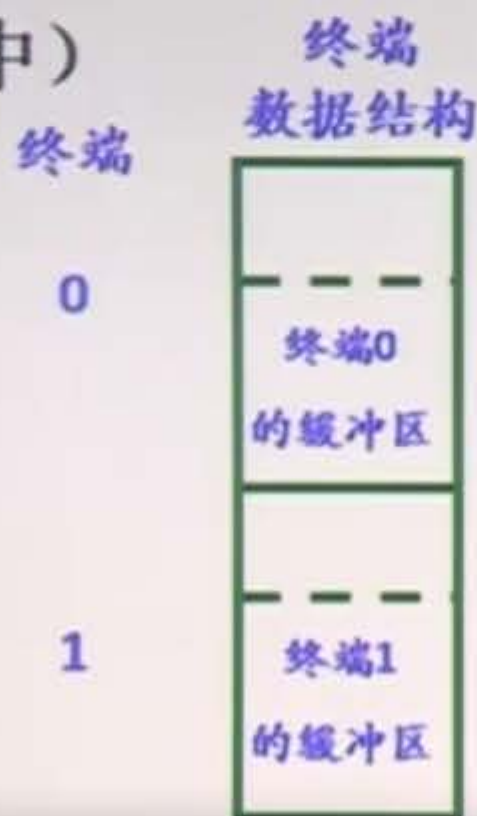
例子

终端输入软件中的键盘驱动程序

其任务之一：收集字符

两种常见的字符缓冲方法：

- ✓ 公共缓冲池（驱动程序中）
- ✓ 终端数据结构缓冲



UNIX SYSTEM V缓冲技术(1/5)

- 采用缓冲池技术，可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速度很有帮助
- 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（尽可能减少磁盘I/O的次数，提高系统运行的速度）

的次数， 来提高系统运行的速度。



UNIX SYSTEM V缓冲技术(2/5)

- 缓冲池：200个缓冲区(512字节或1024字节)

- 每个缓冲区由两部分组成：
缓冲控制块或缓冲首部 + 缓冲数据区

系统通过缓冲控制块来实现对缓冲区的管理

- 空闲缓冲区队列(av链)

队列头部为bfreelist

- 设备缓冲队列(b链)

链接所有分配给各类设备使用的缓冲区，按照散列方式组织

buf用b双向链，可以有64个队列，每个队列首部有头标
设备为b_dev上的逻辑块b在散列队列的头标为：

$$i = (b_dev + b) \bmod 64$$



UNIX SYSTEM V 缓冲技术(3/5)

说明:

- 逻辑设备号和盘块号分别标志出文件系统和数据所在的盘块号，是缓冲区的唯一标志
- 状态标识缓冲区的当前状态：忙/闲、上锁/开锁、是否延迟写、数据有效性等
- 两组指针（**av**和**b**）用于对缓冲池的分配管理

设备号
盘块号
状态
指向缓冲数据区的指针
指向缓冲队列的后继指针 b-back
指向缓冲队列的前驱指针 b-forw
指向空闲队列的后继指针 av-back
指向空闲队列的前驱指针 av-forw



UNIX缓冲技术(4/5)

每个缓冲区同时在**av**链和**b**链:

- 开始: 在空闲**av**链(缓冲区未被使用时)
- 开始IO请求: 在设备IO请求队列和设备**b**链
- IO完成: 在空闲**av**链和设备**b**链

这是缓冲区在不同的队列里的一个情况。



UNIX SYSTEM V缓冲技术(5/5)

- ✓ 当进程想从指定的盘块读取数据时，系统根据盘块号从设备b链(散列队列)中查找，如找到缓冲区，则将该缓冲区状态标记为“忙”，并从空闲av队列中取下，并完成从缓冲区到内存用户区的数据传送
- ✓ 如果在设备b链中未找到时，则从空闲av链队首摘取一个缓冲区，插入设备I/O请求队列；并从原设备b链中取下，插入由读入信息盘块号确定的新的设备b链中
- ✓ 当数据从磁盘块读入到缓冲区后，缓冲区从设备I/O请求队列取下；当系统完成从缓冲区到内存用户区的数据传送后，要把缓冲区释放，链入空闲av链队尾
- ✓ 当数据从磁盘块读入到缓冲区，并传送到内存用户区后，该缓冲区一直保留在原设备b链中，即它的数据一直有效。若又要使用它，则从空闲av链中取下，使用完后插入到空闲av链队尾。若一直未使用它，则该缓冲区从空闲av链队尾慢慢升到队首，最后被重新分配，旧的盘块数据才被置换



下面我们介绍一下 I/O 相关技术，我们主要介绍的是缓冲技术。引入缓冲技术解决什么问题呢？首先我们知道，缓冲技术是操作系统当中最早引入的技术，它当时引入是为了解决 CPU 与外部设备之间速度不匹配的问题。发展到今天，在这样一种情况下，凡是数据到达和离去的速度不匹配的地方，其实都可以采用缓冲技术。采用了缓冲技术，在今天主要也是为了提高 CPU 与设备之间的一种并行的工作能力。减少 I/O 设备对 CPU 的中断次数，而放宽 CPU 对中断响应时间的要求。我们看一下缓冲技术的实现，首先我们看一下缓冲区的分类，一种缓冲区呢，是在硬件设备当中设置的缓冲区，比如说我们的显示器我们会，显卡里头会设置一个很大的缓冲区那么这就是硬缓冲；当然我们也可以在内存当中开辟一个空间呢，用作缓冲区呢，这我们称之为软缓冲。缓冲区的管理呢，这里头包括了单缓冲、双缓冲和缓冲池。单缓冲只设一个缓冲区；双缓冲呢，我们可以两个缓冲区，这当然是在发展的过程当中，都可能会碰到的这种，缓冲区管理的一个需要。那么现在更多用的是呢，缓冲池，也就是多个缓冲或者是循环缓冲，那么操作系统会去统一管理多个缓冲区。采用的是有界缓冲区的生产者/消费者这样一个模型对缓冲区进行相应的管理。我们看一个例子，那么在我们终端，经常在键盘上输入数据，输入、敲击键盘，那么终端的输入软件当中，就有一个键盘的驱动程序。在键盘驱动程序当中呢，我们就要考虑缓冲问题，因为键盘的驱动程序的任务之一，是收集字符，那么收集字符就需要有缓冲区。那么这里提供了两种，啊，常见的字符缓冲区的方法。第一种叫做公共缓冲池，也就是在驱动程序当中，设定这么一个公共缓冲池。那么假设有不同的终端，那么我们每个终端都有数据结构，每个数据结构都可以在公共缓冲池当中，去存取一部分的，啊数据。第二种呢，叫做终端数据结构缓冲，也就是每一个终端跟着一个缓冲区，所以一个终端数据结构，这里头有一部分是放缓冲区的。这是缓冲区的一个举例。下面我们介绍一下 UNIX 系统五采用的缓冲技术。在 UNIX 系统五当中通过这个缓冲技术，可以达到以下几个目标：第一个呢，是用了缓冲池，来平滑和加快信息在内存和磁盘之间的传输。第二个呢，是这个缓冲区的设置还结合了提前读和延迟写技术，将它们结合在一起呢，对于这种重复性的或者阵发性的这种 I/O 进程，对于提高 I/O 的速度都很有帮助。另外一个目标呢，实际上是我们前面所介绍的块高速缓存的思路，也就是可以充分利用之前从磁盘读入，虽然已经传给了用户，但是呢，仍然保存在缓冲区当中的这些数据，我们可以把它充分地重复利用。利用的结果呢，主要是为了减少磁盘 I/O 的次数，来提高系统

运行的速度。我们具体来看一下缓冲技术的实现。在 UNIX 系统五当中，我们的缓冲池呢是设置了 200 个缓冲区。每个缓冲区的大小呢，是 512 字节或者是 1024 字节。每个缓冲区呢，实际上是由两部分组成。一部分是缓冲区的描述信息，也我们称之为缓冲控制块或者缓冲区首部。那么通过这样一些信息来管理缓冲区，另外呢就是缓冲的数据部分，200 个缓冲区呢，我们组织成多个队列。第一个队列呢，是空闲缓冲区队列，简称为 av 链。我们可以通过队列的首部，找到这个链。第二个队列呢，是设备缓冲队列，简称 b 链。它是链接了所有分配给各类设备使用的缓冲区，而这种组织方式呢，是按照散列方式来组织。我们简单地来说明一下相关的数据结构里的一些信息。在这个数据结构当中，我们有一个设备号，有一个盘块号，还有一个状态。那么逻辑设备号和盘块号呢，分别给出来文件系统和数据所在的盘块号。那么这两个号组合起来，是缓冲区的一个唯一标志。那么状态呢，标识了缓冲区目前的情况，比如说它是分配给某一个进程了，还是空闲的。那么为、对数据进行操作的时候呢，是上锁的呢，还是开锁的；那么是否是延迟写、数据的有效性等等。另外呢，就是各种指针。其中包括了：指向缓冲区数据区的指针、以及在不同的队列里头的双链的指针。那么在 UNIX 系统五当中，每个缓冲区可以同时存在 av 链和 b 链当中。我们分析一下不同的场景，在一开始的时候，所有的缓冲区都会在空闲 av 链里头，因为这个缓冲区还没有被使用。当有了 I/O 请求来了之后，某些缓冲区就分配给了这个 I/O 请求。那么，这个缓冲区就会出现在设备 I/O 请求队列当中以及设备 b 链当中，当然它就从 av 链出来，进入了这两个队列。当这个 I/O 请求完成了之后，那么这个缓冲区就应该还回给空闲的 av 链。但是呢，同时它还继续保留在，设备 b 链当中，它就从设备请求队列里出来进入了 av 链，同时在设备 b 链当中继续保持。这是缓冲区在不同的队列里的一个情况。下面我们介绍缓冲区的使用过程，当进程想从指定的盘块当中读取数据的时候，系统会根据盘块号到设备的 b 链当中进行查找，如果找到了这个缓冲区，就把这个缓冲区的状态标志，标识为"忙"，那么由于这个缓冲区可能还保存在 av 链当中，因此要把它从 av 链当中取下，之后呢，就把缓冲区的内容拷贝到内存的用户区，因为这正好是用户所需要的内容。如果在 b 链当中，没有找到这个缓冲区，就从空闲的 av 链当中取一个缓冲区，把它插入到了 I/O 请求队列当中，由于这个缓冲区可能在某个设备的 b 链当中存在，因此呢，我们就把它从这个 b 链当中

取下，插入到这次，啊，我们所处理的这个设备的 b 链当中。当数据从磁盘块读到缓冲区之后，那么缓冲区从 I/O 请求队列当中取下来；那么系统完成了从缓冲区到用户区的数据拷贝工作之后呢，那么就把这个缓冲区呢释放了，把它送到了 av 链。那么当数据从磁盘块读入到缓冲区之后，并且把它送到了用户的内存区，那么这个缓冲区呢，实际上就是保留在了，原来这个设备的 b 链当中。那么它的数据一直是有效的。如果还要再使用其中的内容，就把它从 av 链当中取下，啊，继续使用。使用完了，再把它送到 av 链的队尾。但是如果这个内容，第一次使用完之后，以后很长时间没有使用了。慢慢的，那么这个缓冲区在 av 链当中，就会升到了队首。直到某一次需要的时候，把它内容覆盖掉，放入新的，啊，数据块的内容。这就是一个 UNIX 的缓冲区的一个组织的过程。当然由于只有二百个缓冲区，所以当缓冲区都满了的时候，那么我们需要进行 置换，我们可以采用类似 LRU 的算法，来对缓冲区进行相应的置换管理。