



燕山大学
YANSHAN UNIVERSITY

2022

操作系统A

第六章 输入输出系统

赵谷雨\信息科学与工程学院

Email: gaiazhao@ysu.edu.cn



目录

CONTENTS

7 缓冲区管理

8 磁盘存储器的性能和调度

第六章 输入输出系统

6.7 缓冲区管理

6.7 缓冲区管理

在现代操作系统中，几乎所有的I/O设备在与处理机交换数据时都用了缓冲区。缓冲区是一个存储区域，它可以由专门的硬件寄存器组成，但由于硬件的成本较高，容量也较小，一般仅用在对速度要求非常高的场合，如存储器管理中所用的联想存储器；设备控制器中用的数据缓冲区等。

6.7.1 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。

6.7.1 缓冲的引入

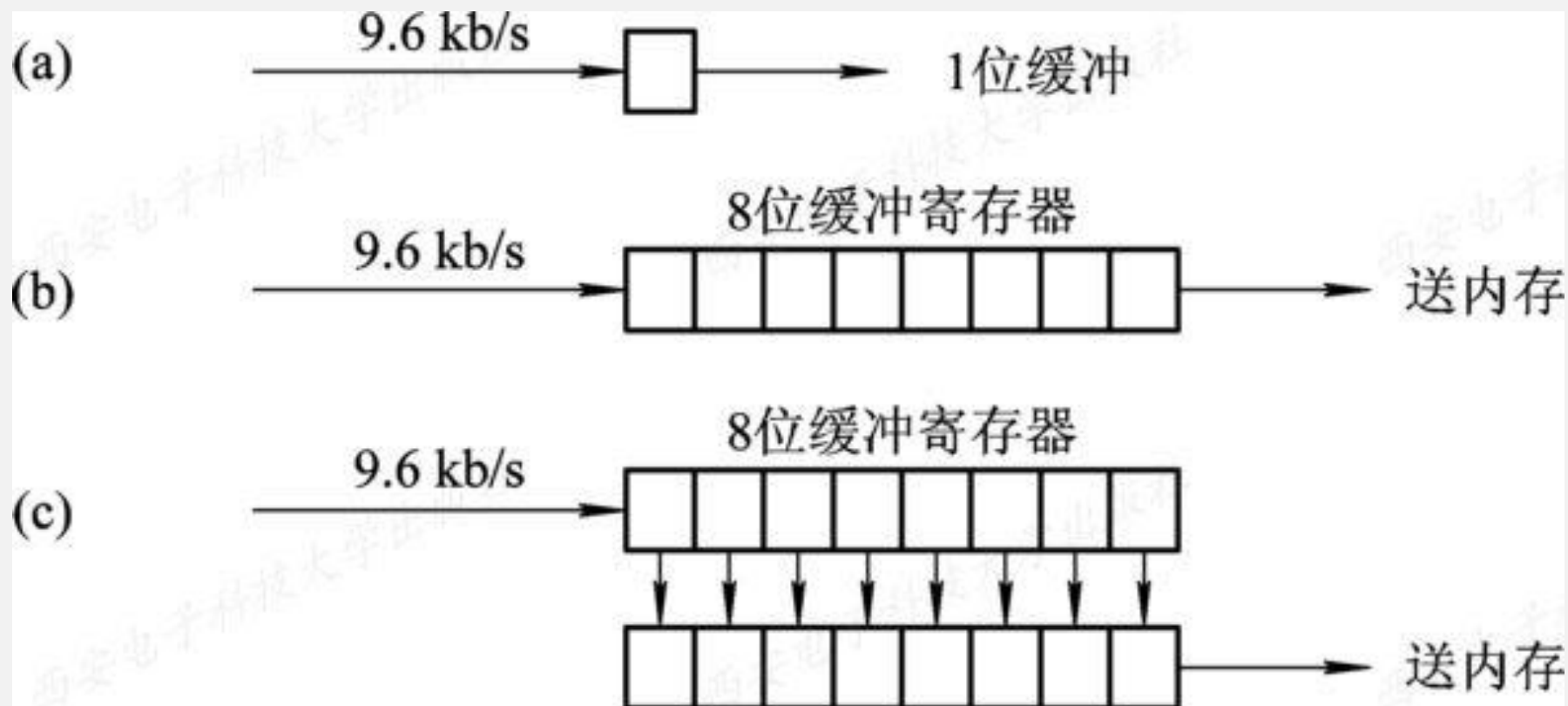


图6-22 利用缓冲寄存器实现缓冲

6.7.2 单缓冲区和双缓冲区

1. 单缓冲区(Single Buffer)

在单缓冲情况下，每当用户进程发出一I/O请求时，操作系统便在主存中为之分配一缓冲区，如图6-23所示。

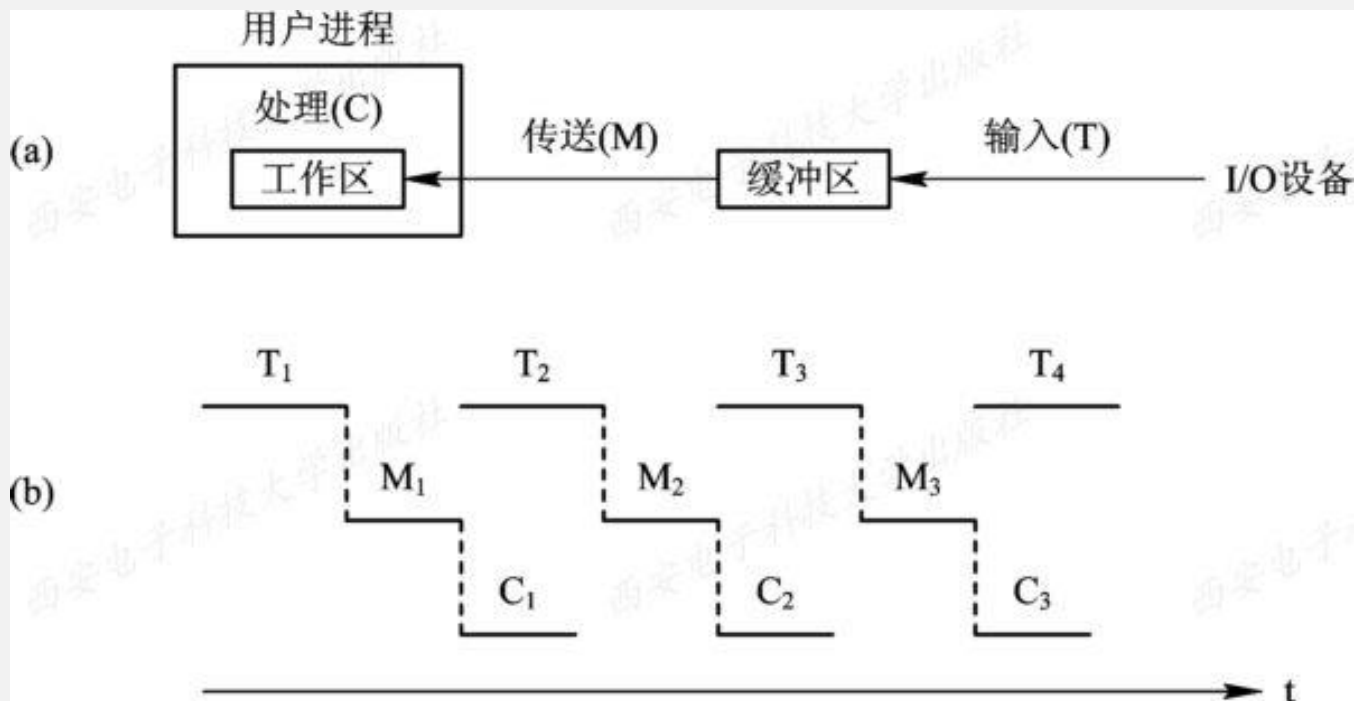


图6-23 单缓冲工作示意图

6.7.2 单缓冲区和双缓冲区

2. 双缓冲区(Double Buffer)

由于缓冲区是共享资源，生产者与消费者在使用缓冲区时必须互斥。如果消费者尚未取走缓冲区中的数据，即使生产者又生产出新的数据，也无法将它送入缓冲区，生产者等待。如果为生产者与消费者设置了两个缓冲区，便能解决这一问题。

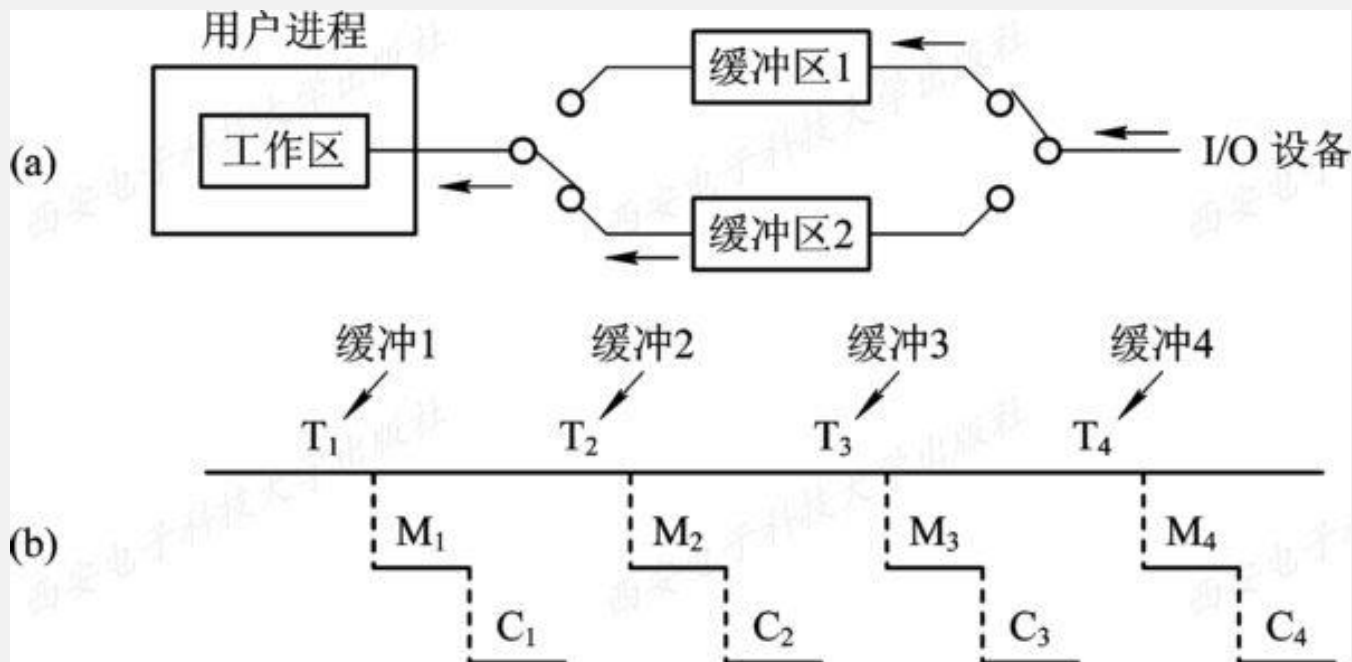
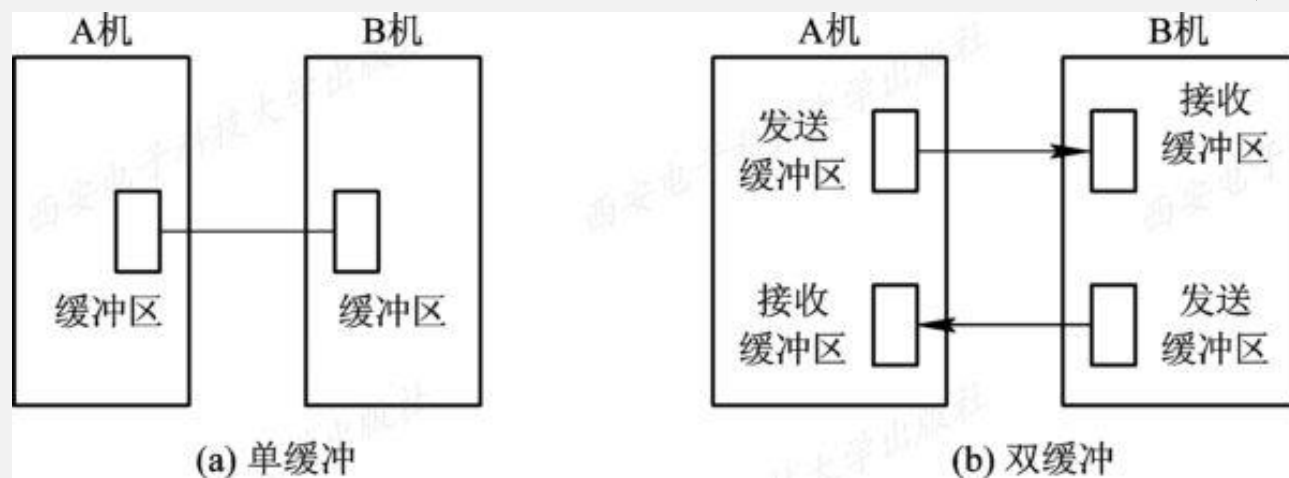


图6-24 双缓冲工作示意图

6.7.2 单缓冲区和双缓冲区

如果在实现两台机器之间的通信时仅为它们配置了单缓冲，如图6-25(a)所示，那么，它们之间在任一时刻都只能实现单方向的数据传输。例如，只允许把数据从A传送到B，或者从B传送到A，而绝不允许双方同时向对方发送数据。为了实现双向数据传输，必须在两台机器中都设置两个缓冲区，一个用作发送缓冲区，另一个用作接收缓冲区，如图6-25(b)所示。

图6-25 双机通信时缓冲区的设置



6.7.3 环形缓冲区

1. 环形缓冲区的组成

(1) 多个缓冲区。在环形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，如图6-26所示。

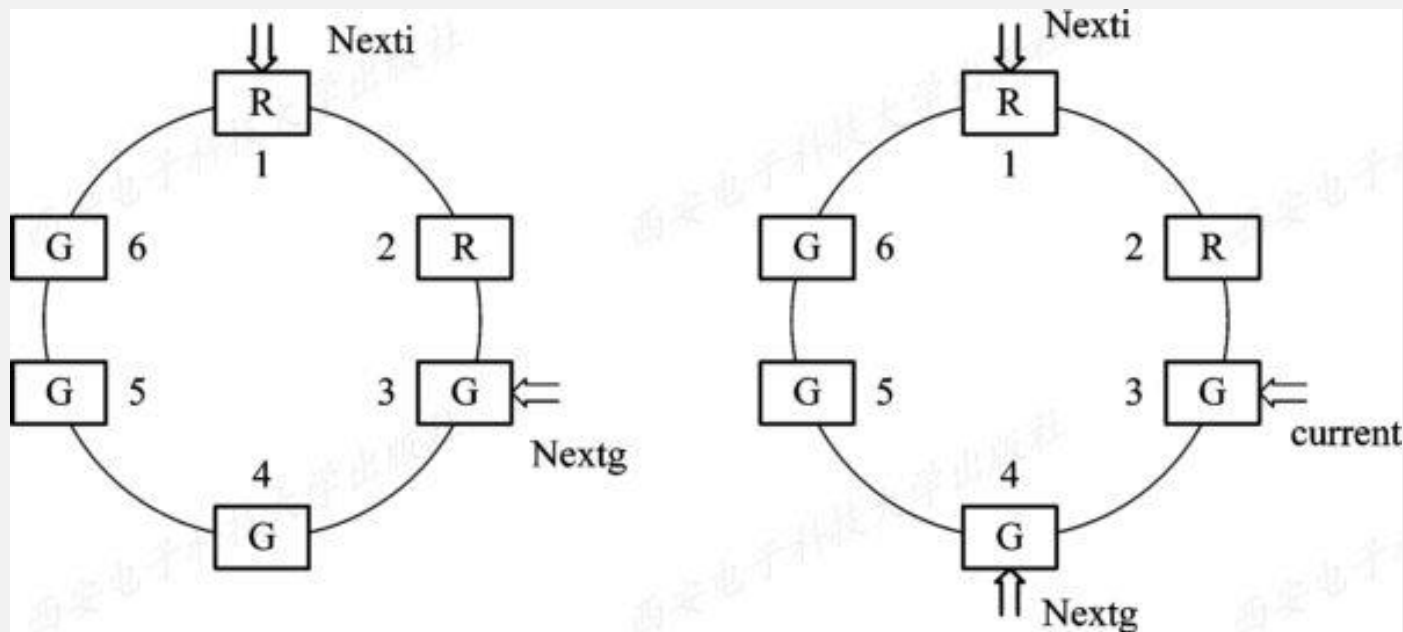


图6-26 环形缓冲区

6.7.3 环形缓冲区

2. 环形缓冲区的使用

计算进程和输入进程可利用两个过程来使用形环缓冲区。

(1) Getbuf过程。

当**计算进程**要使用缓冲区中的数据时，调用Getbuf过程。指针Nextg所指的缓冲区提供给进程使用，把它改为现行工作缓冲区，并令Current指针指向该缓冲区的第一个单元，同时将Nextg移向下一个G缓冲区。每当**输入进程**要使用空缓冲区来装入数据时，调用Getbuf过程，将Nexti所指缓冲区提供给输入进程使用，Nexti指针移向下一个R缓冲区。

(2) Releasebuf过程。

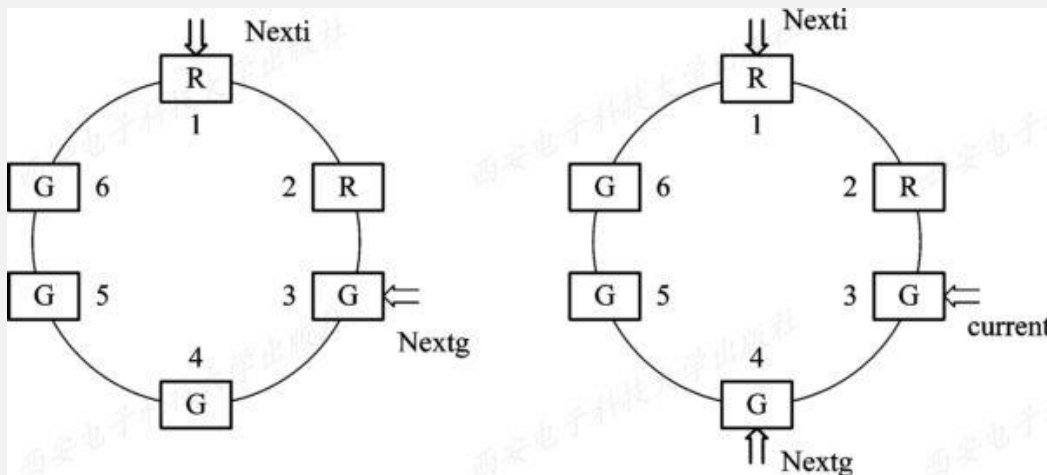
6.7.3 环形缓冲区

3. 进程之间的同步问题

使用输入循环缓冲，可使输入进程和计算进程并行执行。相应地，指针Nexti和指针Nextg将不断地沿着顺时针方向移动，这样就可能出现下述两种情况：

(1) Nexti指针追赶上Nextg指针。

(2) Nextg指针追赶上Nexti指针。



6.7.4 缓冲池(Buffer Pool)

1. 缓冲池的组成

缓冲池管理着多个缓冲区，每个缓冲区由用于标识和管理的缓冲首部以及用于存放数据的缓冲体两部分组成。缓冲首部一般包括缓冲区号、设备号、设备上的数据块号、同步信号量以及队列链接指针等。为了管理上的方便，一般将缓冲池中具有相同类型的缓冲区链接成一个队列，于是可形成以下三个队列：

- (1) 空白缓冲队列emq。
- (2) 输入队列inq。
- (3) 输出队列outq。

6.7.4 缓冲池(Buffer Pool)

2. Getbuf过程和Putbuf过程

在数据结构课程中，曾介绍过队列和对队列进行操作的两个过程，第一个是 Addbuf(type, number)过程。该过程用于将由参数number所指示的缓冲区B挂在type队列上。第二个是 Takebuf(type)过程。它用于从type所指示的队列的队首摘下一个缓冲区。

为使诸进程能互斥地访问缓冲池队列，为每个队列设置一个互斥信号量MS(type)。为保证诸进程同步使用缓冲区，为每个缓冲队列设置一个资源信号量RS(type)。

6.7.4 缓冲池(Buffer Pool)

3. 缓冲区的工作方式

缓冲区可以工作在如下四种工作方式，如图6-27所示。

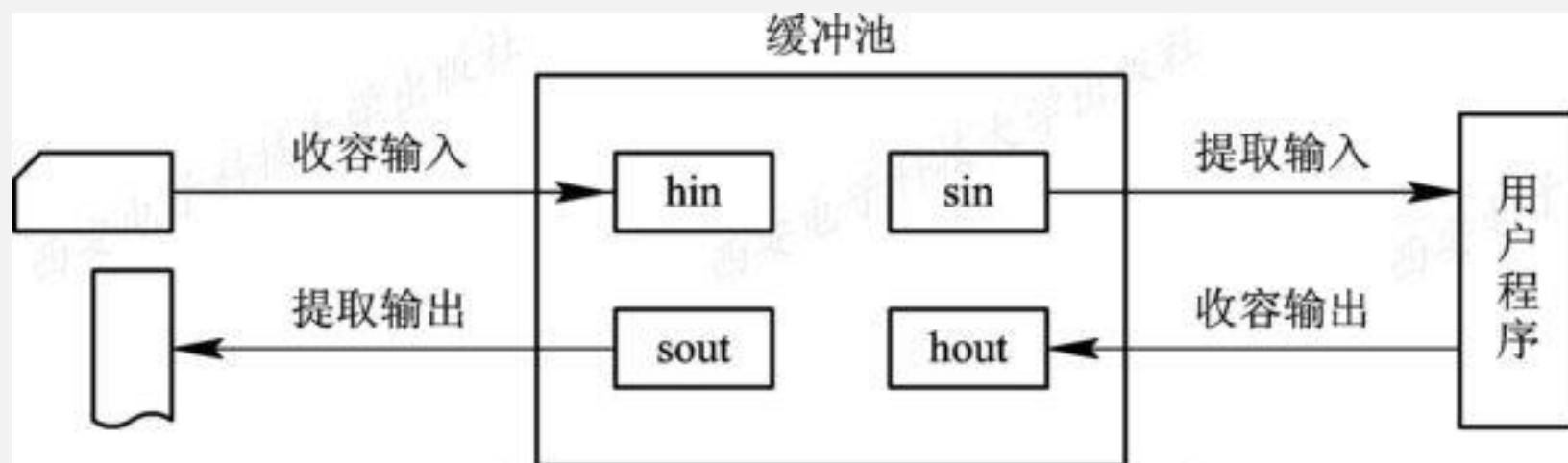


图6-27 缓冲区的工作方式

第六章 输入输出系统

6.8 磁盘存储器的性能和调度

6.8.1 磁盘性能简述

磁盘设备是一种相当复杂的机电设备，在此仅对磁盘的某些性能，如数据的组织、磁盘的类型和访问时间等方面做扼要的阐述。

6.8.1 磁盘性能简述

1. 数据的组织和格式

磁盘设备可包括一个或多个物理盘片，每个磁盘片分一个或两个存储面(Surface)(见图6-28(a))，每个盘面上有若干个磁道(Track)，磁道之间留有必要的间隙(Gap)。为使处理简单起见，在每条磁道上可存储相同数目的二进制位。

6.8.1 磁盘性能简述

1. 数据的组织和格式

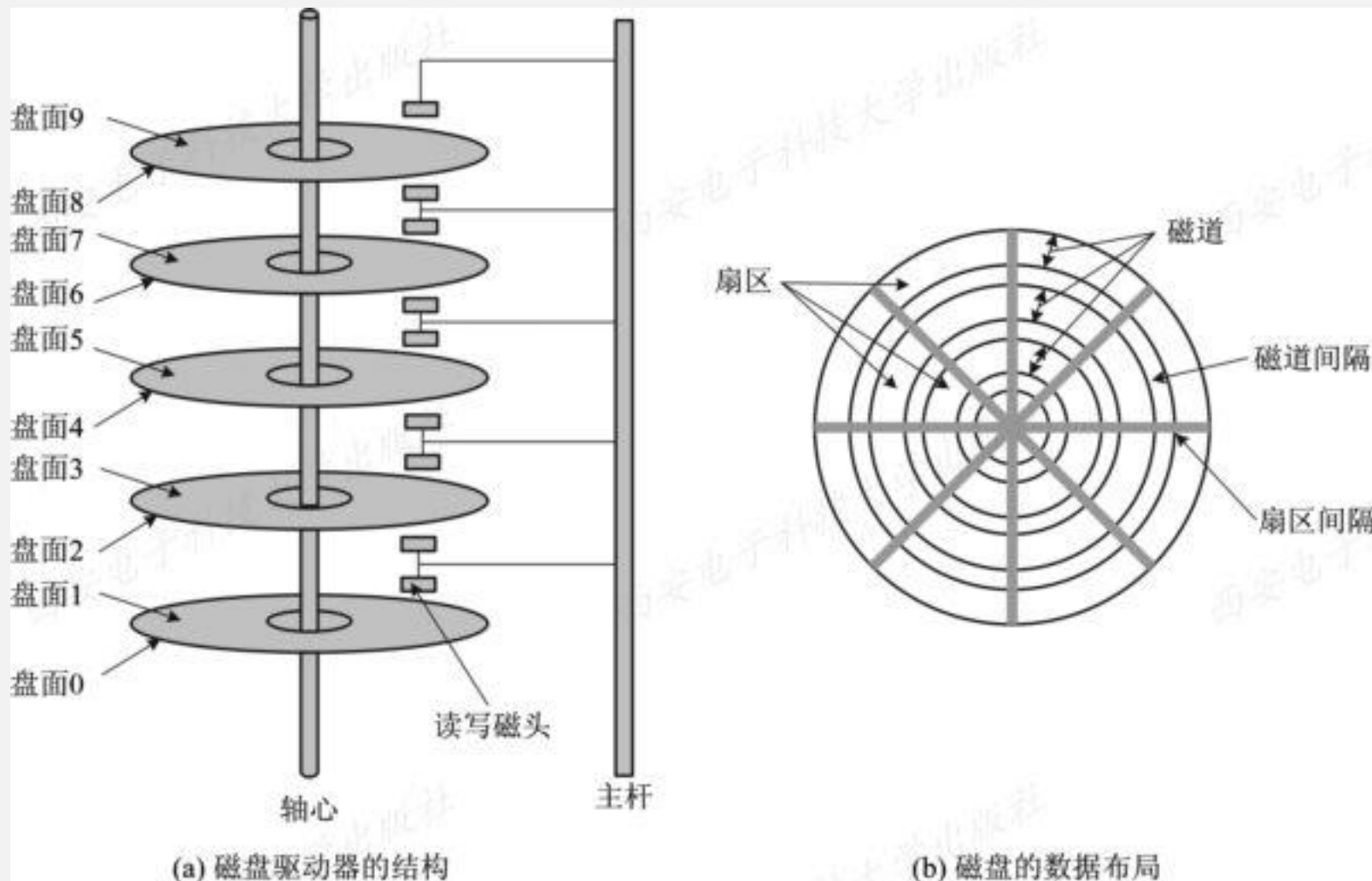


图6-28 磁盘的结构和布局

6.8.1 磁盘性能简述



6.8.1 磁盘性能简述



6.8.1 磁盘性能简述

1. 数据的组织和格式

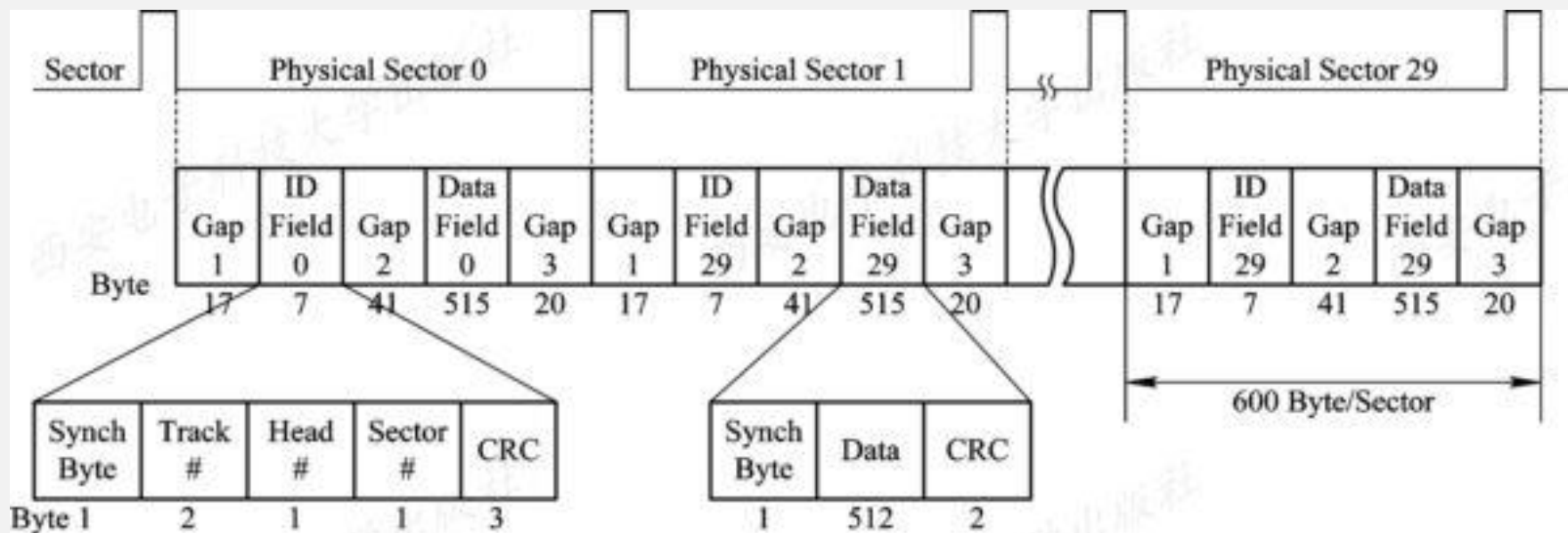


图6-29 磁盘的格式化

6.8.1 磁盘性能简述

2. 磁盘的类型

对于磁盘，可以从不同的角度进行分类。最常见的有：将磁盘分成硬盘和软盘、单片盘和多片盘、固定头磁盘和活动头(移动头)磁盘等。下面仅对固定头磁盘和移动头磁盘做些介绍。

(1) 固定头磁盘。

(2) 移动头磁盘。

6.8.1 磁盘性能简述

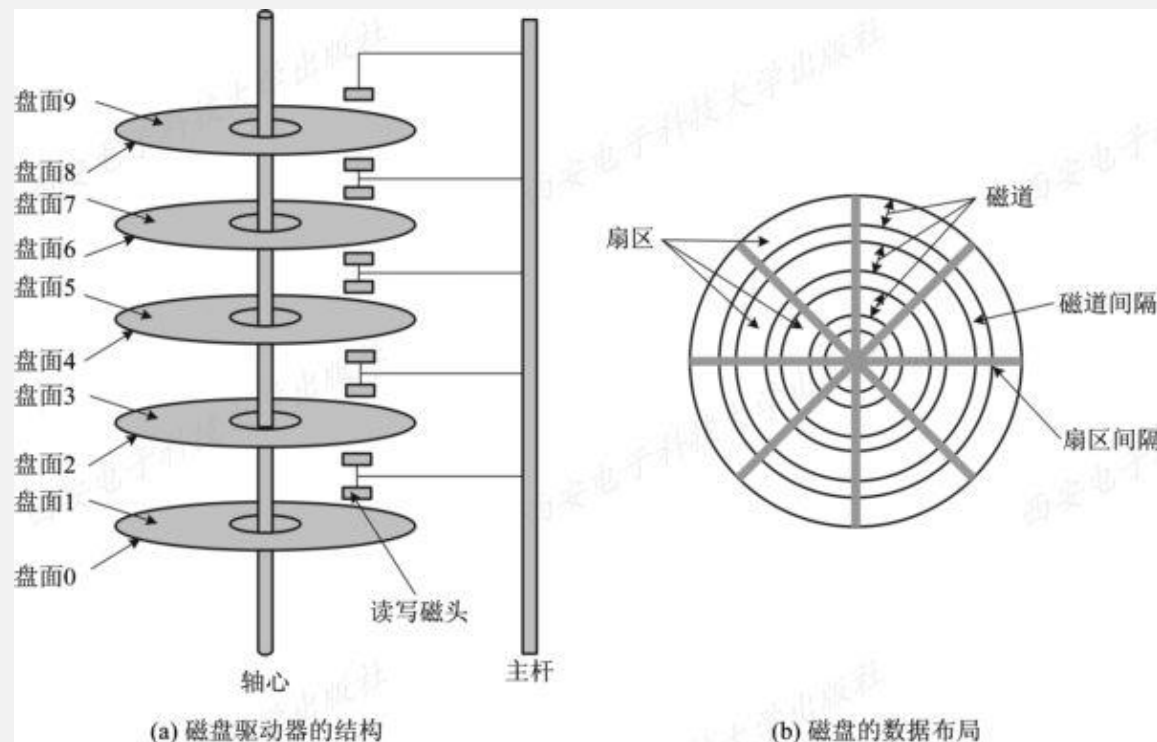
3. 磁盘访问时间

磁盘设备在工作时以恒定速率旋转。为了读或写，磁头必须能移动到所指定的磁道上，并等待所指定的扇区的开始位置旋转到磁头下，然后再开始读或写数据。

1. 寻道时间 T_s

2. 旋转延迟时间 T_r

3. 传输时间 T_t



6.8.2 早期的磁盘调度算法

1. 先来先服务(FCFS)

这是最简单的磁盘调度算法。它根据进程请求访问磁盘的先后次序进行调度。

图6-30 FCFS调度算法

55,58,39,18,90,160,150,38,184

(从100号磁道开始)	
被访问的 下一个磁道号	移动距离 (磁道数)
55	45
58	3
39	19
18	21
90	72
160	70
150	10
38	112
184	146
平均寻道长度: 55.3	

6.8.2 早期的磁盘调度算法

2. 最短寻道时间优先(SSTF)

该算法选择这样的进程，其要求访问的磁道与当前磁头所在的磁道距离最近，以使每次的寻道时间最短，但这种算法不能保证平均寻道时间最短。

图6-31 SSTF调度算法

55,58,39,18,90,160,150,38,184

(从100号磁道开始)	
被访问的 下一个磁道号	移动距离 (磁道数)
90	10
58	32
55	3
39	16
38	1
18	20
150	132
160	10
184	24
平均寻道长度: 27.5	

6.8.3 基于扫描的磁盘调度算法

1. 扫描(SCAN)算法

SSTF算法的实质是基于优先级的调度算法，因此就可能导致优先级低的进程发生“饥饿”(Starvation)现象。因为只要不断有新进程的请求到达，且其所要访问的磁道与磁头当前所在磁道的距离较近，这种新进程的I/O请求必然优先满足。在对SSTF算法略加修改后，则可防止低优先级进程出现“饥饿”现象。

6.8.2 早期的磁盘调度算法

1. 扫描(SCAN)算法

55,58,39,18,90,160,150,38,184

(从100# 磁道开始, 向磁道号 增加方向访问)	
被访问的 下一个磁道号	移动距离 (磁道数)
150	50
160	10
184	24
90	94
58	32
55	3
39	16
38	1
18	20
平均寻道长度: 27.8	

图6-32 SCAN调度算法示例

6.8.3 基于扫描的磁盘调度算法

2. 循环扫描(CSCAN)算法

SCAN算法既能获得较好的寻道性能，又能防止“饥饿”现象，故被广泛用于大、中、小型机器和网络中的磁盘调度。但也存在这样的问题：当磁头刚从里向外移动而越过了某一磁道时，恰好又有一进程请求访问此磁道，这时，该进程必须等待，待磁头继续从里向外，然后再从外向里扫描完处于外面的所有要访问的磁道后，才处理该进程的请求，致使该进程的请求被大大地推迟。

6.8.2 早期的磁盘调度算法

2. 循环扫描(CSCAN)算法

55,58,39,18,90,160,150,38,184

(从100# 磁道开始, 向磁道号 增加方向访问)	
被访问的 下一个磁道号	移动距离 (磁道数)
150	50
160	10
184	24
18	166
38	20
39	1
55	16
58	3
90	32
平均寻道长度: 35.8	

图6-33 CSCAN调度算法示例



谢谢大家！ Q & A