磁盘管理

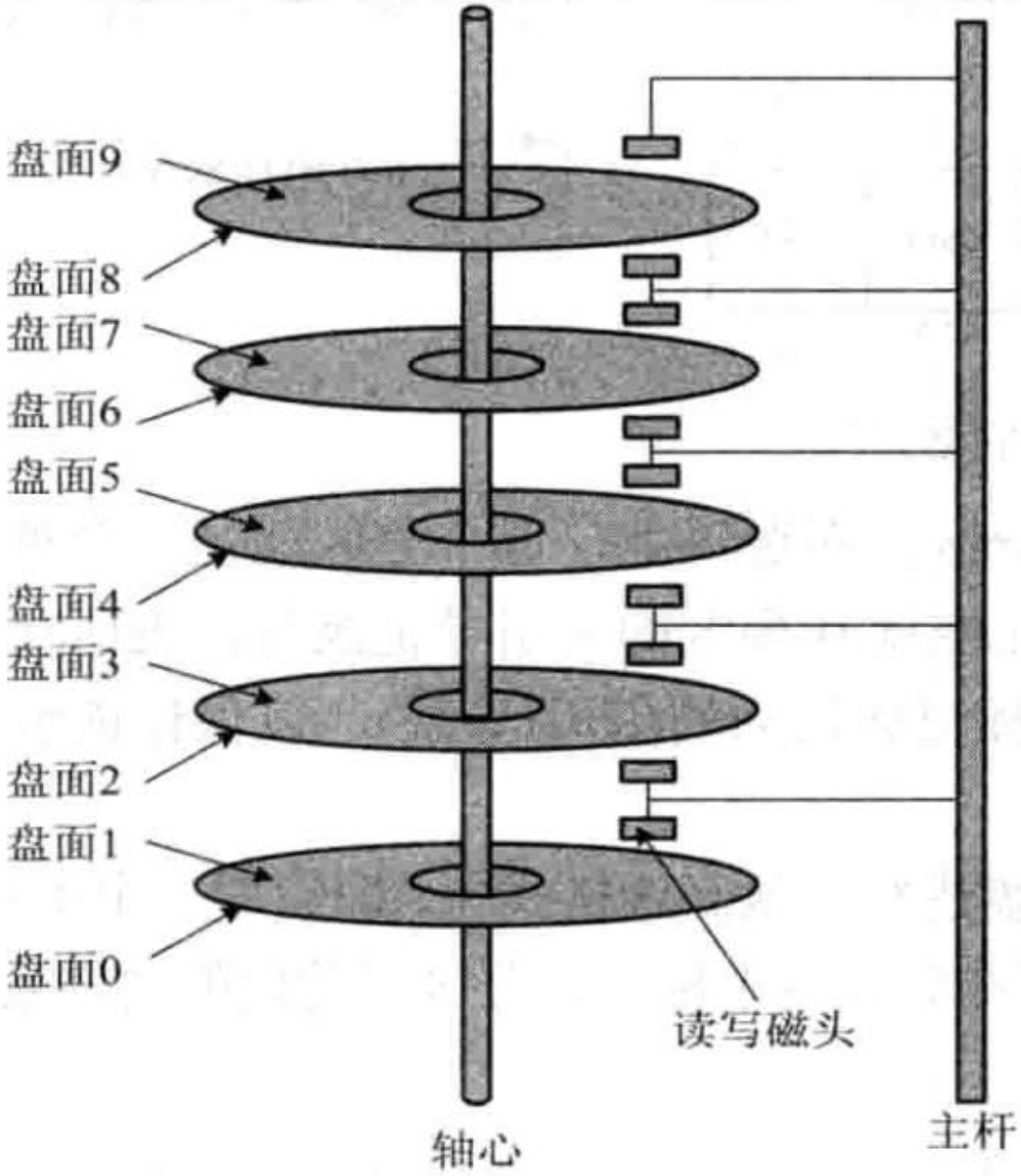
6、设备管理

I/O 系统；I/O 控制方式；缓冲管理；I/O 软件；设备分配；磁盘存储器的管理（磁盘性能、磁盘调度、磁盘高速缓存）。

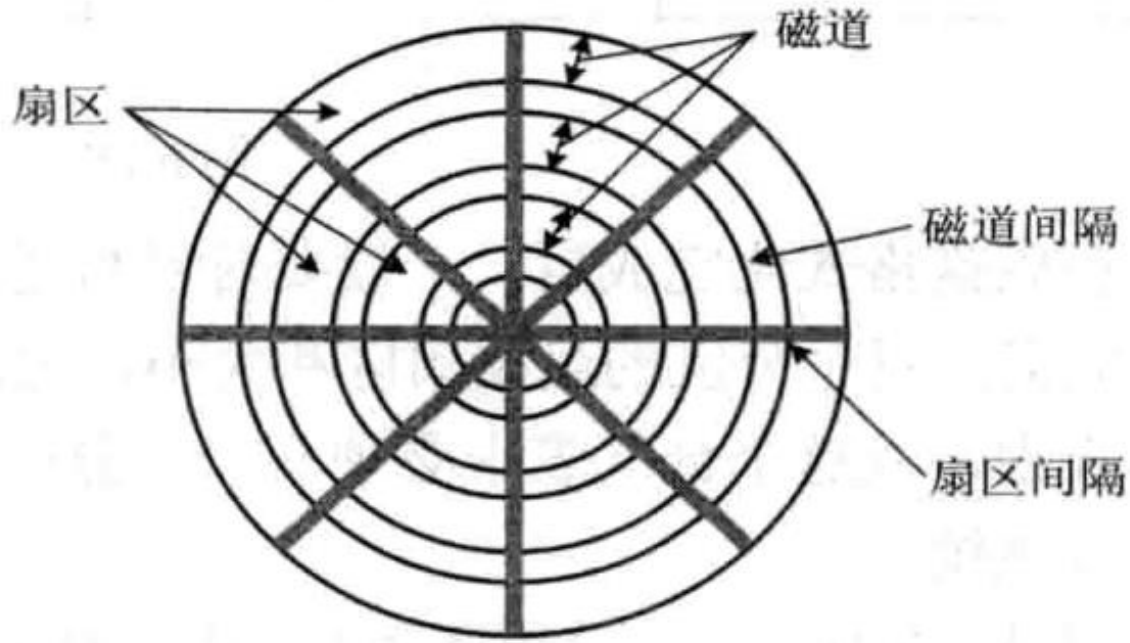
5.3.1 磁盘

磁盘（Disk）是由表面涂有磁性物质的物理盘片，通过一个称为*磁头*的导体线圈从磁盘存取数据。在读/写操作期间，磁头固定，磁盘在下面高速旋转。如图 5.13 所示，磁盘盘面上的数据存储在一组同心圆中，称为*磁道*。每个磁道与磁头一样宽，一个盘面有上千个磁道。磁道又划分为几百个扇区，每个扇区固定存储大小，一个扇区称为一个盘块。相邻磁道及相邻扇区间通过一定的间隙分隔开，以避免精度错误。注意，由于扇区按固定圆心角度划分，所以密度从最外道向里道增加，磁盘的存储能力受限于最内道的最大记录密度。

磁盘安装在一个磁盘驱动器中，它由磁头臂、用于旋转磁盘的主轴和用于数据输入/输出的电子设备组成。如图 5.14 所示，多个盘片垂直堆叠，组成磁盘组，每个盘面对应一个磁头，所有磁头固定在一起，与磁盘中心的距离相同且一起移动。所有盘片上相对位置相同的磁道组成柱面。扇区是磁盘可寻址的最小单位，磁盘上能存储的物理块数目由扇区数、磁道数及磁盘面数决定，磁盘地址用“柱面号·盘面号·扇区号”表示。



(a) 磁盘驱动器的结构



(b) 磁盘的数据布局

6.8 磁盘存储器的性能和调度

磁盘存储器是计算机系统中的最重要的存储设备，在其中存放了大量的文件。对文件的读、写操作都将涉及到对磁盘的访问。磁盘 I/O 速度的高低和磁盘系统的可靠性，将直接影响到系统的性能。可以通过多种途经来改善磁盘系统的性能。首先可通过选择好的磁盘调度算法，以减少磁盘的寻道时间；其次是提高磁盘 I/O 速度，以提高对文件的访问速度；第三采取冗余技术，提高磁盘系统的可靠性，建立高度可靠的文件系统。第二和第三点我们将它放在磁盘存储器管理一章中介绍。

磁盘访问时间

5.3.3 磁盘调度算法

一次磁盘读写操作的时间由寻找（寻道）时间、旋转延迟时间和传输时间决定。

1）寻找时间 T_s。活动头磁盘在读写信息前，将磁头移动到指定磁道所需要的时间。这个时间除跨越 n 条磁道的时间外，还包括启动磁臂的时间 s，即

$$T_s = m \times n + s$$

式中，m 是与磁盘驱动器速度有关的常数，约为 0.2ms，磁臂的启动时间约为 2ms。

2）旋转延迟时间 T_r。磁头定位到某一磁道的扇区所需要的时间，设磁盘的旋转速度为 r，则

$$T_r = \frac{1}{2r}$$

对于硬盘，典型的旋转速度为 5400 转/分，相当于一周 11.1ms，则 T_r 为 5.55ms；对于软盘，其旋转速度为 300～600 转/分，则 T_r 为 50～100ms。

3）传输时间 T_t。从磁盘读出或向磁盘写入数据所经历的时间，这个时间取决于每次所读/写的字节数 b 和磁盘的旋转速度：

$$T_t = \frac{b}{rN}$$

式中，r 为磁盘每秒的转数，N 为一个磁道上的字节数。

在磁盘存取时间的计算中，寻道时间与磁盘调度算法相关；而延迟时间和传输时间都与磁盘旋转速度相关，且为线性相关，所以在硬件上，转速是磁盘性能的一个非常重要的参数。

总平均存取时间 T_a 可以表示为

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

虽然这里给出了总平均存取时间的公式，但是这个平均值是没有太大实际意义的，因为在实际的磁盘 I/O 操作中，存取时间与磁盘调度算法密切相关。

为了减少对文件的访问时间，应采用一种最佳的磁盘调度算法，以使各进程对磁盘的平均访问时间最小。由于在访问磁盘的时间中主要是寻道时间，因此，磁盘调度的目标是使磁盘的平均寻道时间最少。目前常用的磁盘调度算法有先来先服务、最短寻道时间优先及扫描等算法。下面逐一介绍。

磁盘调度算法

- 先来先服务FCFS 磁头当前位置
- 最短寻找时间优先SSTF
- 扫描算法SCAN（电梯调度算法） 磁头移动方向 往返迂回
- 循环扫描C-SCAN 单向循环
- LOOK 往返迂回 不触碰边缘
- C-LOOK 单向循环 不触碰边缘

注意，若无特别说明，也可以默认 SCAN 算法和 C-SCAN 算法为 LOOK 和 C-LOOK 调度（请读者认真领悟，并通过结合后面的习题进一步加深对以上相关算法的理解）。

磁盘高速缓存（Disk Cache）

在前面介绍的高速缓存，是指在内存和 CPU 之间所增设的一个小容量高速存储器。而在这里所要介绍的磁盘高速缓存，是指在内存中为磁盘盘块设置的一个缓冲区，在缓冲区中保存了某些盘块的副本。当出现一个访问磁盘的请求时，由核心先去查看磁盘高速缓冲器，看所请求的盘块内容是否已在磁盘高速缓存中，如果在，便可从磁盘高速缓存中去获取，这样就省去了启动磁盘操作，而且可使本次访问速度提高几个数量级；如果不在，才需要启动磁盘将所需要的盘块内容读入，并把所需盘块内容送给磁盘高速缓存，以便以后又需要访问该盘块的数据时，便可直接从高速缓存中提取。在设计磁盘高速缓存时需要考虑的问题有：

(1) 如何将磁盘高速缓存中的数据传送给请求进程；

(2) 采用什么样的置换策略；

(3) 已修改的盘块数据在何时被写回磁盘。

下面对它们做简单介绍。

如何将磁盘高速缓存中的数据传送给进程

1. 数据交付(Data Delivery)方式

如果 I/O 请求所需要的数据能从磁盘高速缓存中获取，此时就需要将磁盘高速缓存中的数据传送给请求进程。所谓的数据交付就是指将磁盘高速缓存中的数据传送给请求者进程。系统可以采取两种方式将数据交付给请求进程：

(1) 数据交付，这是直接将高速缓存中的数据传送到请求者进程的内存工作区中；

(2) 指针交付，只将指向高速缓存中某区域的指针交付给请求者进程。后一种方式由于所传送的数据量少，因而节省了数据从磁盘高速缓存存储空间到进程的内存工作区的时间。

采用什么样的置换方法

2. 置换算法

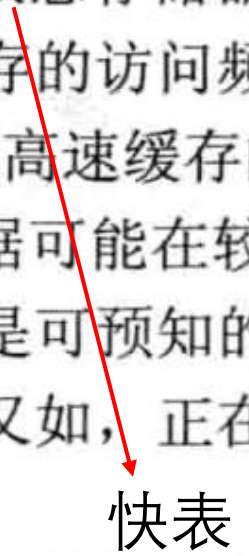
如同请求调页(段)一样，在将磁盘中的盘块数据读入高速缓存时，同样会出现因高速缓存中已装满盘块数据，而需要将其中某些盘块的数据先换出的问题。相应地，也存在着采用哪种置换算法的问题。较常用的算法仍然是最近最久未使用算法 LRU、最近未使用算法 NRU 及最少使用算法 LFU 等。由于请求调页中的联想存储器与高速缓存(磁盘 I/O 中)的工作情况不同，因而使得在置换算法中所应考虑的问题也有所差异。因此，现在不少系统在设计其高速缓存的置换算法时，除了考虑到最近最久未使用这一原则外，还考虑了以下几点：

时钟置换算法

(1) 访问频率。通常，每执行一条指令时，便可能访问一次联想存储器，亦即联想存储器的访问频率基本上与指令执行的频率相当。而对磁盘高速缓存的访问频率，则与磁盘I/O 的频率相当。因此，对联想存储器的访问频率远远高于对磁盘高速缓存的访问频率。

(2) 可预见性。在磁盘高速缓存中的各盘块数据，有哪些数据可能在较长时间内不会再被访问，又有哪些数据可能很快就再被访问，会有相当一部分是可预知的。例如，对二次地址及目录块等，在它被访问后，可能会很久都不再被访问。又如，正在写入数据的未满盘块，可能会很快又被访问。

(3) 数据的一致性。由于磁盘高速缓存在内存中，而内存是一种易失性的存储器，一旦系统发生故障，存放在缓存中的数据将会丢失；而其中有些盘块(如索引结点盘块)中的数据已被修改，但尚未拷回磁盘。因此，当系统发生故障后，可能造成数据的不一致性。



已修改的盘块数据在何时被写回磁盘

3. 周期性地写回磁盘

还有一种情况值得注意，那就是根据 LRU 算法，那些经常要被访问的盘块数据可能会一直保留在高速缓存中，长期不会被写回磁盘。这是因为链中任一元素在被访问之后，又被挂到链尾而不被写回磁盘，只有一直未被访问的元素才有可能移到链首，而被写回磁盘。为了解决这一问题，在 UNIX 系统中专门增设了一个修改(update)程序，使之在后台运行，该程序周期性地调用一个系统调用 SYNC。其主要功能是强制性地将所有在高速缓存中已修改的盘块数据写回磁盘。一般是把两次调用 SYNC 的时间间隔定为 30 s。这样，因系统故障所造成的工作损失不会超过 30 s 的工作量。

数据一致性

8.5 数据一致性控制

在实际应用中，经常会在多个文件中都含有同一个数据。所谓数据一致性问题是指，保存在多个文件中的同一数据，在任何情况下都必需能保证相同。例如，当我们发现某种商品的进价有错时，我们必须同时修改流水账，付费账、分类账及总账等一系列文件中的该商品的价格，方能保证数据的一致性。但如果在修改进行到中途时系统突然发生故障，就会造成各个账目中该数据的不一致性，进而使多个账目不一致。为了保证数据的一致性，在现代 OS 中都配置了能保证数据一致性的软件。

事务的定义

1. 事务的定义

事务是用于访问和修改各种数据项的一个程序单位。事务也可以被看做是一系列相关读和写操作。被访问的数据可以分散地存放在同一文件的不同记录中，也可放在多个文件中。只有对分布在不同位置的同一数据所进行的读和写(含修改)操作全部完成时，才能以托付操作(Commit Operation)，也称为提交操作，结束事务，确认事务的变化。其后其它的进程或用户才将可以查看到事务变化后的新数据。但是，只要这些操作中有一个读、写或修改操作失败，便必须执行夭折操作(Abort Operation)，也称为回滚操作或取消操作。这些读或写操作的失败可能是由于逻辑错误，也可能是系统故障所导致的。

```
begin transaction;
    update account set money = money-100 where name = '张三';
    update account set money = money+100 where name = '李四';
commit transaction;
```

事务的特性

一个被夭折的事务，通常已执行了一些操作，因而可能已对某些数据做了修改。为使夭折的事务不会引起数据的不一致性，需将该事务内刚被修改的数据项恢复成原来的情况，使系统中各数据项与该事务未执行时的数据项内容完全相同。此时，可以说该事务“已被退回”(rolled back)。不难看出，一个事务在对一批数据执行修改操作时，应该是要么全部完成，并用修改后的数据去代替原来的数据，要么一个也不修改。事务操作所具有的这种特性，就是我们在第二章中曾讲过的“原子操作”，即事务具有原子性(Atomic)。

作为单个程序单元执行的一系列操作，并不是都可以成为事务，也就是说，如果定义其为事务，则必须同时满足四个属性，即事务属性 ACID。除了上述的原子性外，事务还应具备的属性是：① 一致性(Consistent)，即事务在完成时，必须使所有的数据都保持一致状态；② 隔离性(Isolated)，即对一个事务对数据所作的修改，必须与任何其它与之并发事务相隔离，换言之，一个事务查看数据时数据所处的状态，要么是另一并发事务修改它之前的状态，要么是另一事务修改它之后的状态，而不会是任何中间状态的数据；③ 持久性(Durable)，即事务完成之后，它对于系统的影响是永久性的。

并发控制

8.5.3 并发控制(Concurrent Control)

在多用户系统和计算机网络环境下，可能有多个用户在同时执行事务。由于事务具有原子性，这使各个事务的执行必然是按某种次序依次进行的，只有在一个事务执行完后，才允许另一事务执行，即各事务对数据项的修改是互斥的。人们把这种特性称为顺序性，而把用于实现事务顺序性的技术称为并发控制。该技术在应用数据库系统中已被广泛采用，现也广泛应用于 OS 中。虽然可以利用第二章所介绍的信号量机制来保证事务处理的顺序性，但在数据库系统和文件服务器中应用得最多的，还是较简单的且较灵活的同步机制——锁。

并发控制有哪些方法？（实现事务顺序性的技术有哪些？）

1. 利用互斥锁实现“顺序性”

实现顺序性的一种最简单的方法，是设置一种用于实现互斥的锁，简称为互斥锁(Exclusive Lock)。在利用互斥锁实现顺序性时，应为每一个共享对象设置一把互斥锁。当某一事务 T_i 要去访问某对象时，应先获得该对象的互斥锁。若成功，便用该锁将该对象锁住，于是事务 T_i 便可对该对象执行读或写操作；而其它事务由于未能获得该锁，因而不能访问该对象。如果 T_i 需要对一批对象进行访问，则为了保证事务操作的原子性，T_i 应先获得这一批对象的互斥锁，以将这些对象全部锁住。如果成功，便可对这一批对象执行读或写操作；操作完成后又将所有这些锁释放。但如果在这一批对象中的某一个对象已被其它事物锁住，则此时 T_i 应对此前已被 T_i 锁住的其它对象进行开锁，宣布此次事务运行失败，但不致引起数据的变化。

2. 利用互斥锁和共享锁实现顺序性

利用互斥锁实现顺序性的方法简单易行。目前有不少系统都是采用这种方法来保证事务操作的顺序性，但这却存在着效率不高的问题。因为一个共享文件虽然只允许一个事务去写，但却允许多个事务同时去读；而在利用互斥锁来锁住文件后，则只允许一个事务去读。为了提高运行效率而又引入了另一种形式的锁——共享锁(Shared Lock)。共享锁与互斥锁的区别在于：互斥锁仅允许一个事务对相应对象执行读或写操作，而共享锁则允许多个事务对相应对象执行读操作，但不允许其中任何一个事务对对象执行写操作。

在为一个对象设置了互斥锁和共享锁的情况下，如果事务 T_i 要对 Q 执行读操作，则只需去获得对象 Q 的共享锁。如果对象 Q 已被互斥锁锁住，则 T_i 必须等待；否则，便可获得共享锁而对 Q 执行读操作。如果 T_i 要对 Q 执行写操作，则 T_i 还须去获得 Q 的互斥锁。若失败，须等待；否则，可获得互斥锁而对 Q 执行写操作。利用共享锁和互斥锁来实现顺序性的方法非常类似于我们在第二章中所介绍的读者—写者问题的解法。

真题

- 2016.09 位示图+磁盘结构计算
- 2018.05 事务 名词解释
- 2018.13 磁盘调度算法计算
- 2019.04 并发控制 名词解释

练习

- 1.目前常用的磁盘调度算法有哪几种？每种算法优先考虑的问题是什么？
- 2.磁盘的访问时间由哪几部分构成？
- 3.何谓磁盘高速缓存？在设计磁盘高速缓存时需要考虑哪些问题？
- 4.可以采用哪几种方式将磁盘高速缓存中的数据传送给进程？
- 5.何谓事务，如何保证事务的原子性？
- 6.事务有哪些特性？

- 1.（1）先来先服务算法优先考虑进程请求访问磁盘的先后次序;

（2）最短寻道时间优先算法优先考虑要求访问的磁道与当前磁头所在磁道距离是否最近:

（3）扫描算法考虑欲访问的磁道与当前磁道间的距离，更优先考虑磁头当前的移动方向。

- 2.磁盘访问时间由寻道时间Ts、旋转延迟时间Tr、传输时间Tt三部分组成。
- 3.磁盘高速缓存是指在内存中为磁盘盘块设置的一个缓冲区，在缓冲区中保存了某些盘块的副本。要考虑的问题有:

(1) 如何将磁盘高速缓存中的数据传送给请求进程;

(2) 采用什么样的置换策略;

(3) 已修改的盘块数据在何时被写回磁盘。

- 4.（1）数据交付，直接将高速缓存中的数据传送到进程的内存工作区中。（2）指针交付，将指向高速缓存中某区域的指针给到进程。
- 5.事务是用于访问和修改各种数据项的一个程序单位。一个事务在对一批数据执行修改操作时，要么全部完成，并用修改后的数据去替代原来的数据，要么一个也不修改，从而保证了事务的原子性。
- 6.四大特性：

  （1）原子性，事务要么全部执行完成，要么一个也不执行

  （2）一致性，事务完成后，必须使所有数据保持一致状态

  （3）隔离性，一个事务对数据进行修改，必须与其他并发事务相隔离

  （4）持久性，事务完成后，他对于系统的影响是永久性的