

存储器系统

主存储器

基础概念

- 存储器是计算机系统中的记忆设备，用来**存放程序和数据**。
- 计算机中全部信息，包括输入的原始数据、计算机程序、中间运行结果和最终运行结果都保存在存储器中

存储器分类：速度越来越慢，容量越来越大，单位造价越来越低

1. 寄存器
2. Cache（高速缓冲存储器）
3. 主存储器
4. 辅存储器

存储器的存取方式

存取方式	读/写装置	数据块标志	访问特性	代 表
顺序存取	共享读/写装置	无	特定线性顺序	磁带
直接存取	共享读/写装置	数据分块，每块一个唯一标志	可直接移到特定数据块	磁盘
随机存取	每个可寻址单元专 有读/写装置	每个可寻址单元均有一个唯 一地址	随时访问任何一个存储 单元	主存储器
相联存取（属 随机存取）	每个可寻址单元专 用读/写装置	每个可寻址单元均有一个唯 一地址	根据内容而非地址来选 择读写点	Cache

存储器的性能

- **存取时间**：对于随机存取而言，就是完成一次读 / 写所花的时间；对非随机存取，就是将读 / 写装置移动到目的位置所花的时间
- **存储器带宽**：**每秒能访问的位数**。通常存储器周期是纳秒级(ns)。计算公式是：**1 / 存储器周期×每周可访问的字节数**。
- **数据传输率**：每秒输入/输出的数据位数；对于随机存取而言，传输率 $R=1/\text{存储器周期}$

主存储器的种类

- RAM:随机存储器，可读 / 写，只能暂存数据，断电后数据丢失
 1. SRAM:静态随机存储器，在不断电时信息能够一直保持，读写速度快，生产成本低，多用于容量较小的高速缓冲存储器；cache使用该种的RAM
 2. DRAM:动态随机存储器，需要定时刷新以维持信息不丢失，读写速度较慢，集成度高，生产成本低，多用于容量较大的主存储器；内存使用该种的RAM
- ROM:只读存储器，出厂前用掩膜技术写入，常用于存放BIOS和微程序控制
- PROM:**可编程ROM，只能够一次写入**，需用特殊电子设备进行写入
- EPROM:可擦除的PROM，用某种方法可擦去信息，**可写入多次**
- E2PROM:**电可擦除**EPROM，可以写入，但速度慢
- 闪速存储器(Flash Memory):其特性介于EPROM与E2PROM之间。但不能进行字节级别的删除操作

- CAM（相联存储器）：CAM是一种特殊的存储器，是一种基于数据内容进行访问的存储设备。其速度比基于地址进行读写的方式要快

辅助存储器

磁带

- 磁带是一种顺序存取的设备
- 特点：存储容量大，价格便宜。适合数据的备份存储

磁盘

RAID（独立磁盘冗余阵列）

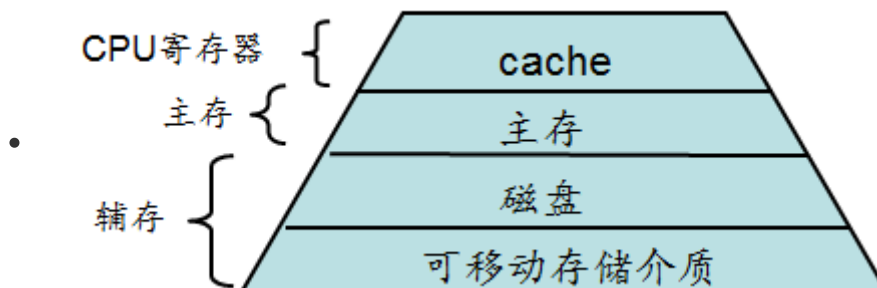
- 把多个相对便宜的磁盘组合起来，成为一个磁盘组，配合数据分散排列的设计，提升数据的安全性和整个磁盘系统效能
- 优点：
 1. 利用多磁盘来提高数据传输率
 2. 通过数据冗余与校验实现可靠性
- 主要技术：
 1. 分块技术
 2. 交叉技术
 3. 重聚技术
- RAID 0级（无冗余和无校验的数据分块）
 - 原理：把连续的数据分散到多个磁盘上存取，数据请求被多个磁盘并行执行，每个磁盘执行属于自己的那部分数据请求。这种数据上的并行操作充分利用总线的带宽，显著提高磁盘整体存取性能
 - 优点：具有最高的I/O性能和最高的磁盘空间利用率，易管理
 - 缺点：不提供数据冗余，一旦数据损坏，损坏的数据将无法得到恢复
 - 适用场景：对性能要求较高，对数据安全要求低的领域，如图形工作站等。对于个人用户，也是提高硬盘存储性能的绝佳选择
- RAID 1级（磁盘镜像阵列）
 - 原理：由磁盘对组成，每个工作盘都有其对应的镜像盘，上面保存着与工作盘完全相同的数据拷贝，具有最高的安全性，但磁盘空间利用率只有50%
 - 适用场景：主要用于存放系统软件、数据及其他重要文件
- RAID 2级（采用纠错海明码的磁盘阵列）
 - 原理
 - 采用海明码纠错技术，用户增加校验盘来提供纠错和验错功能，磁盘驱动器组中的第1个、第2个、第4个.....第2n个磁盘驱动器是专门的校验盘，用于校验和纠错，其余的用于存放数据。RAID2最少要三台磁盘驱动器方能运作
- RAID 3级（采用带奇偶校验码的并行传送）
 - 原理：RAID 3把数据分成多个“块”，按照奇偶校验算法存放在N+1个硬盘上，实际数据占用的有效空间为N个硬盘的空间总和，第N+1个硬盘上存储的数据是校验容错信息。当N+1个硬盘中的一个硬盘出现故障时，从其他N个硬盘中可以恢复原始数据。所以RAID 3，安全性可以得到保障
 - 适用场景：RAID 3比较适合大文件类型且安全性要求较高的应用，如视频编辑、硬盘播放机和大型数据库等
- RAID 4级（带奇偶校验码的独立磁盘结构）

- 原理：RAID4和RAID3很像，不同的是，它对数据的访问是按数据块进行的。也就是按磁盘进行的，每次是一个盘。RAID 4使用一块磁盘作为奇偶校验盘，每次写操作都需要访问奇偶盘，这时奇偶校验盘成为写操作的瓶颈。一个数据块是一个完整的数据集合，比如一个文件就是一个典型的数据块。一个数据块存储在一个磁盘上
- RAID 5级（无独立校验盘的奇偶校验码磁盘阵列）
 - 原理
 - RAID 5把数据和奇偶校验信息存储到组成RAID5的各个磁盘上并且奇偶校验信息和相对应的数据分别存储于不同的磁盘上
 - 当RAID5的一个磁盘数据损坏后，利用剩下的数据和相应的奇偶校验信息去恢复被损坏的数据
 - 优点：RAID5磁盘空间利用率较高： $(N-1)/N$
 - 缺点：RAID4和RAID5使用了独立存取技术，阵列中每一个磁盘都相互独立地操作，所以I/O请求可以并行处理。该技术非常适合于I/O请求率高的应用，不太适用于要求高数据传输率的应用
- RAID 6级（具有独立的数据硬盘与两个独立的分布式校验方案）
 - 原理：RAID6技术是在RAID 5基础上，为了进一步加强数据保护而设计的一种RAID方式，是一种扩展RAID 5等级
 - 与RAID 5的不同之处：除了每个硬盘上都有同级数据XOR校验区外，还有一个针对每个数据块的XOR校验区。当前盘数据块的校验数据不是存在当前盘而是交错存储的，每个数据块有了两个校验保护，所以RAID 6的数据冗余性能相当好
 - 缺点：由于增加了一个校验，所以写入的效率较RAID 5还差，而且控制系统的设计也更为复杂，第二块的校验区也减少了有效存储空间
- RAID 10级
 - 把RAID0和RAID1技术结合起来，即RAID0+1 是磁盘分段及镜像的结合，结合了RAID0及RAID1的优点
 - 它采用两组RAID0的磁盘阵列互为镜像，也就是它们之间又成为一个RAID1阵列

cache存储器

Cache(高速缓冲存储器)

- 高速缓冲存储器是位于主存与CPU之间的一级存储器，由静态存储芯片(SRAM)组成，容量比较小但速度比主存高得多，接近于CPU的速度。但其成本更高，因此Cache的容量要比内存小得多。Cache存储了频繁访问内存的数据

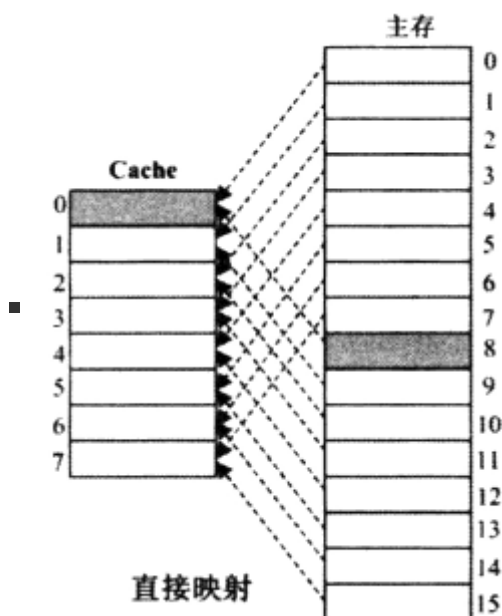


Cache原理、命中率、失效率

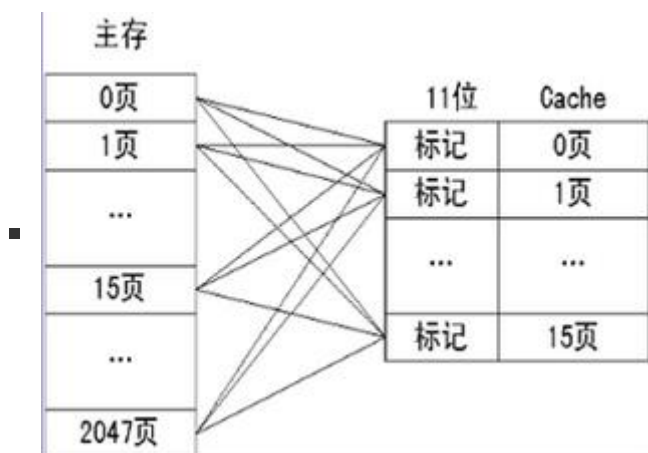
- 使用Cache改善系统性能的主要依据是程序的局部性原理（时间局部性：现在访问过的，一会儿还会访问；空间局部性：现在访问的地方，与其相邻的位置也会访问）
- Cache的访问命中率为 h （通常 $1-h$ 就是Cache的失效率），Cache的访问周期时间是 t_1 ，主存储器的访问周期时间是 t_2 ，则整个系统的平均访存时间就是： $t_3 = h \times t_1 + (1-h) \times t_2$

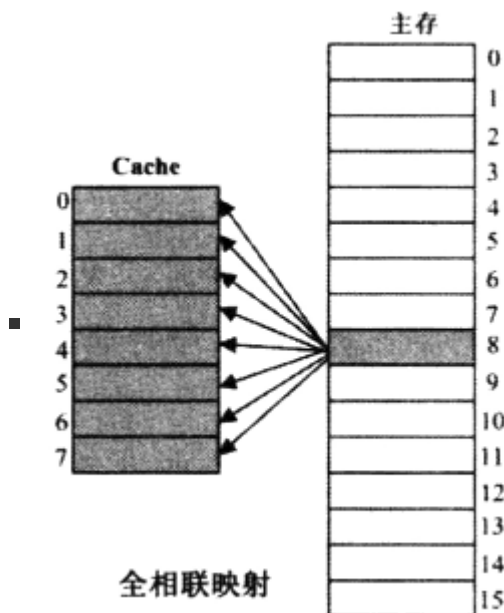
Cache存储器的映射机制：cache行的大小与物理内存的一块相同

- 分配给Cache的地址存放在一个相联存储器（CAM）中。CPU发生访存请求时，会先让CAM判断所要访问的数据是否在Cache中，如果命中就直接使用
- 常见的映射方法：
 - **直接映射**：是一种多（内存空间）对一（cache空间）的映射关系，但一个主存块只能够复制到Cache的一个特定位置上去
 - Cache的行号 i 和主存的块号 j 有函数关系： $i = j \% m$ （其中 m 为Cache总行数）



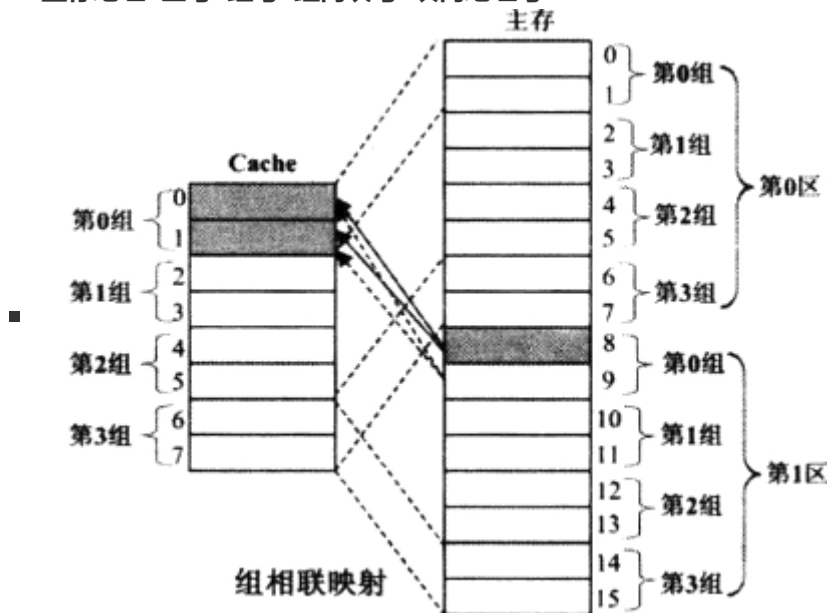
- **全相联映射**（速度最慢）：将主存中任一主存块能映射到Cache中任意行（主存块的容量等于Cache行容量）；根据主存地址不能直接提取Cache页号，而是需要将主存块标记与Cache各页的标记逐个比较，直到找到标记符合的页（访问Cache命中），或者全部比较完后仍无符合的标记（访问Cache失败）；主存块标记与Cache各页的标记逐个比较，所以这种映射方式速度很慢，失掉了高速缓存的作用，这是全相联映射方式的最大缺点。如果让主页标记与各Cache标记同时比较，则成本太高





- **组相联映射** (局部混乱、全局有序)：是前两种方式的折中方案。它将Cache中的块再分成组，各组之间是直接映像，而组内各块之间则是全相联映像

- **主存地址=区号+组号+组内块号+块内地址号**



Cache淘汰算法：平均命中率最高的是LRU算法

- 随机淘汰算法
- 先进先出淘汰算法(FIFO)
- 最近最少使用淘汰算法(LRU)

Cache存储器的写操作

- 写直达：当Cache写命中时，Cache与主存同时发生写修改
- 写回：当CPU对Cache写命中时，只修改Cache的内容而不立即写入主存，当此行被换出才写回主存
- 标记法：数据进入Cache后,有效位置1;当CPU对该数据修改时，数据只写入主存并将该有效位置0。当要从Cache中读取数据时要测试其有效位，若为1则直接从Cache中取数，否则从主存中取数