

计算机组成原理

PRINCIPLES OF COMPUTER ORGANIZATION

第17次课：7.3 虚拟存储器

杜国栋

信息科学与工程学院计算机科学与工程系

gddu@ysu.edu.cn



燕山大学
YANSHAN UNIVERSITY



课程目标

- 掌握主存-辅存层次与Cache-主存层次的比较；
- 熟悉页式虚拟存储器和段页式虚拟存储器的工作原理；
- 了解虚拟存储器工作的全过程。





主存与Cache的地址映射

- 主存地址定位到Cache中，称作地址映射。
- 由于采用**硬件**，这个过程很快，对程序员是透明的。
- 以固定大小的**数据块**为单位进行整体调度（交换）；
- 三种主存↔Cache映射
 - 直接映射
 - 全相联映射
 - 组相联映射



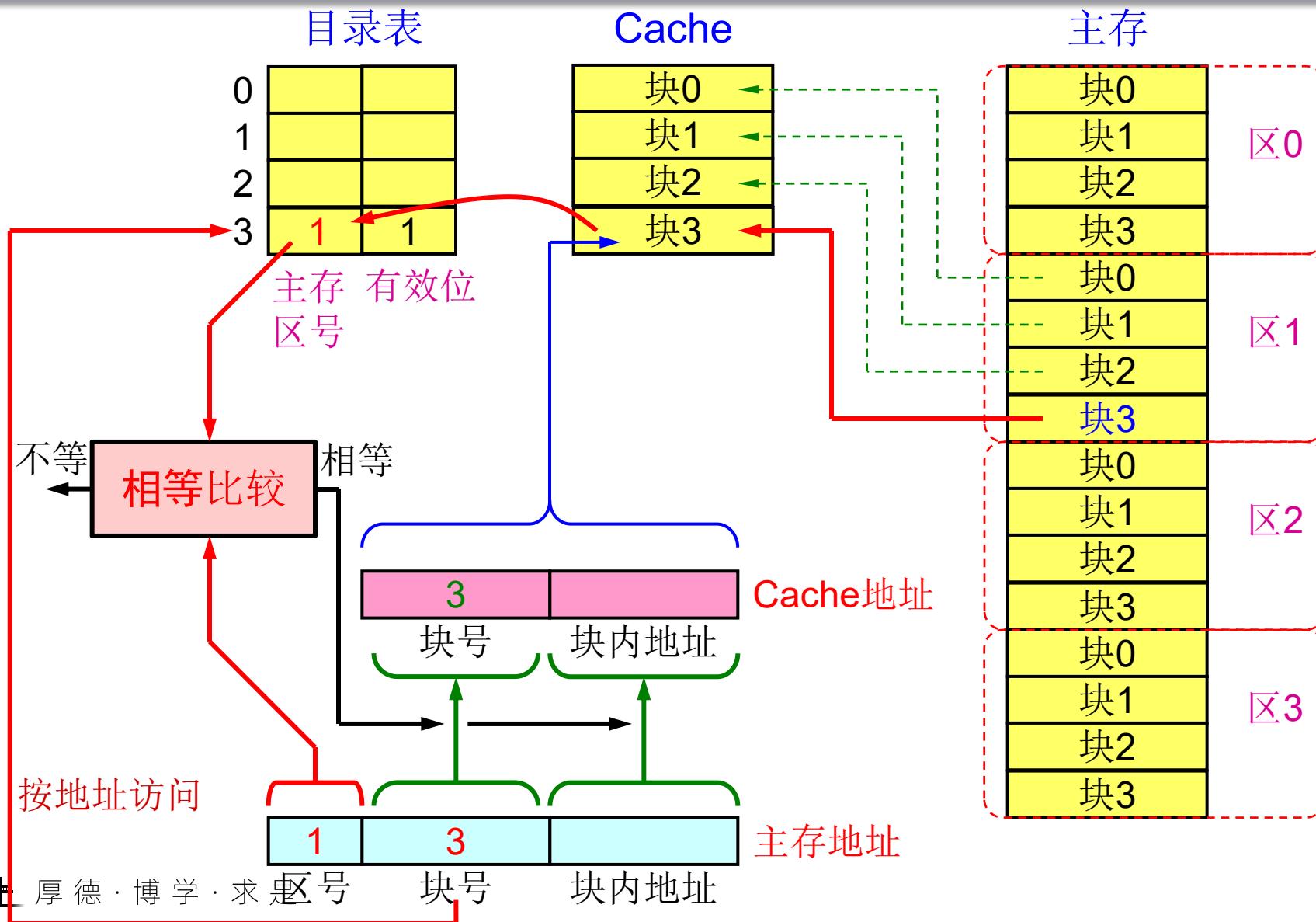


直接映射

- Cache: 只分块、不分组
- 主存: 既分块、也分组(每组的块数 = Cache块数)
- [映射规则] 主存的每一个数据块, 只能映射到与其组内序号相同的Cache数据块位置。

如果: K为Cache的块序号, J为主存块的序号, c为Cache块号的位数。

则 $K=J \bmod 2^c$



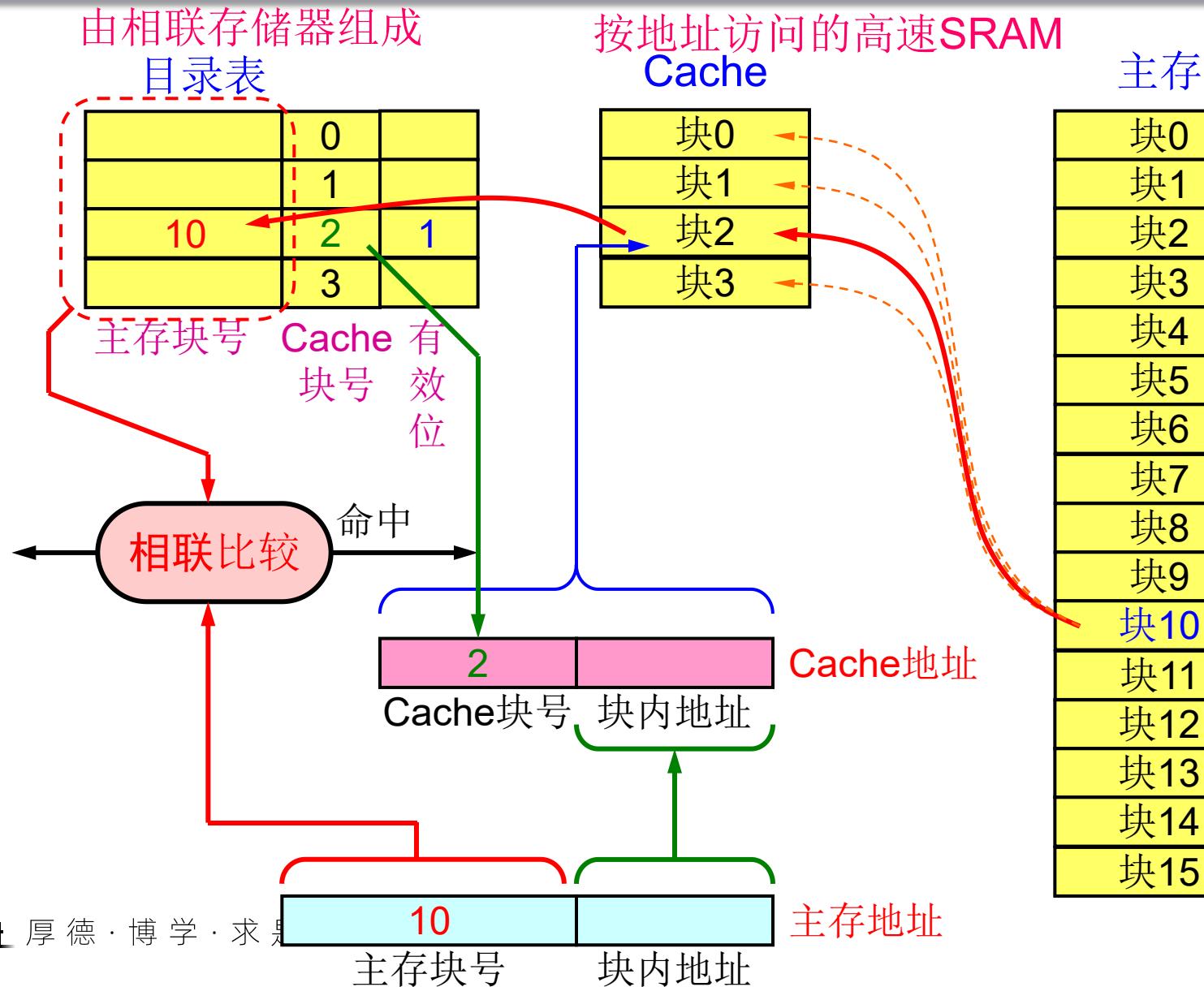


全相联映射

- Cache: 只分块、不分组
- 主存: 只分块、不分组
- [映射规则] 主存任何一个块都可以映射到 Cache 的任何一个数据块位置上。

存在的缺点：

- ✓ Cache 标记太长，判断时间太长。
- ✓ 硬件复杂、成本高、实现相对困难。



有一Cache系统，字长为16位，主存容量为16字×256块，Cache的容量为16字×8块。采用全相联映射，求

(1) 主存和Cache的容量各为多少字节，主存和Cache的字地址各为多少位？

(2) 如原先已经依次装入5块的信息，问字地址为338H所在的主存块将装入Cache块的块号及在Cache中的字地址是多少？

(3) 如块表中地址为1的行中标记着36H的主存块标记，Cache的块号标记位为5H，则在CPU送来主存的字地址为368H时是否命中？如果命中，此时Cache的字地址为多少？

作答



有一Cache系统，字长位16位，主存容量为16字×256块，Cache的容量为16字×8块。采用全相联映射，求

(1) 主存和Cache的容量各为多少**字节**，主存和Cache的字地址各为多少**位**？

(2) 如原先已经依次装入5块的信息，问字地址为338H所在的主存块将装入Cache块的块号及在Cache中的字地址是多少？

(1) 字长16位，一个字2B 主存容量： $16 \times 256 \times 2B = 8KB$

Cache容量： $16 \times 8 \times 2B = 256B$

主存字地址表示： $8KB / 2B = 4K = 2^{12}$ 需要12位表示。

Cache字地址表示： $256B / 2B = 128 = 2^7$ ，需要7位表示。

主存地址：块号8位需要表示，块内地址需要4位表示，共12位

Cache地址：块号需要3位，块内地址需要4位，共7位





有一Cache系统，字长位16位，主存容量为16字×256块，
Cache的容量为16字×8块。采用全相联映射，求

(1) 主存和Cache的容量各为多少**字节**，主存和Cache的字地址各为多少**位**？

(2) 如原先已经依次装入5块的信息，问字地址为338H所在的主存块将装入Cache块的块号及在Cache中的字地址是多少？

(2) 主存地址：338H-> 0011 0011 | 1000

块号 | 块内地址

已经装入5块 (01234)，再装入5号地址 (101)

Cache地址： 101 | 1000 -->1011000->58H

33H

8H

主存地址

主存块号 8位

厚德·博学·求是

5H

8H

Cache地址

Cache块号 3位 块内地址 4位



有一Cache系统，字长位16位，主存容量为16字×256块，Cache的容量为16字×8块。采用全相联映射，求

(3) 如块表中地址为1的行中标记着36H的主存块标记，Cache的块号标记位为5H，则在CPU送来主存的字地址为368H时是否命中？如果命中，此时Cache的字地址为多少？

(3) 块表中地址为1的行中标记着36H主存块号标记
当CPU送来主存地址368H时，主存块号为36H,命中。
此时Cache地址为58H

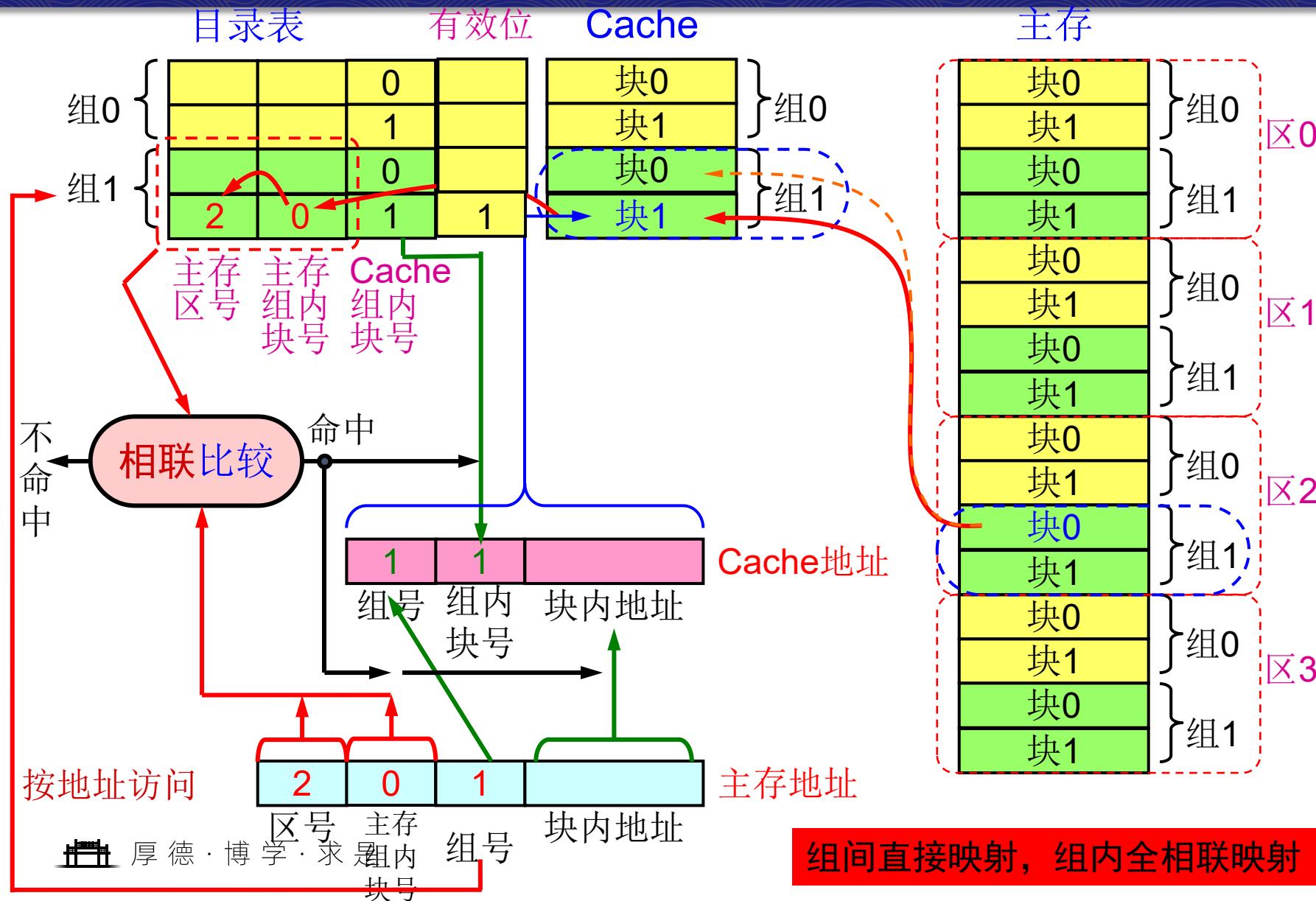
主存地址：368H-> 0011 0110 | 1000
 块号 块内地址





组相联映射

- Cache: 既分块、也分组
 - 主存: 既分块、也分组 (组内块数 = Cache组数)
 - 主存数据块, 映射到与自己组内块序号相同的Cache分组, 可占据Cache分组中的任意数据块位置。
 - 定位Cache的分组: 直接映射;
 - 定位Cache数据块: 全相联映射;
- 直接映射和全相联映射的折衷
- 速度快、硬件简单、成本低、易实现**



某计算机的Cache共有16块，采用2路-组相联映射方式(即每组包括2块)。存储器按字节编址，每个主存块大小为32字节，那么129号主存单元所在的主存块应装入到的Cache组号是()

- A 0
- B 2
- C 4
- D 6

 提交



Cache共有**16块**，采用**2路-组相联映射**方式(即每组包括2块)。
存储器按**字节**编址，每个主存块大小为**32字节**，
那么129号主存单元所在的主存块应装入到的Cache组号是

Cache如何分组、分块？

Cache有16个块，**2路-组相联** 每组**2块** 分**8组**

存储器按字节编制，每块**32B**，即：**32个存储单元**，**块内地址5位**

主存如何分组、分块？

主存分若干组，每组**2块**

$129/32=4\ldots..1$ 故129号单元位于第**4**块主存块
因此映射到Cache的 $4 \bmod 8 = 4$ 组。

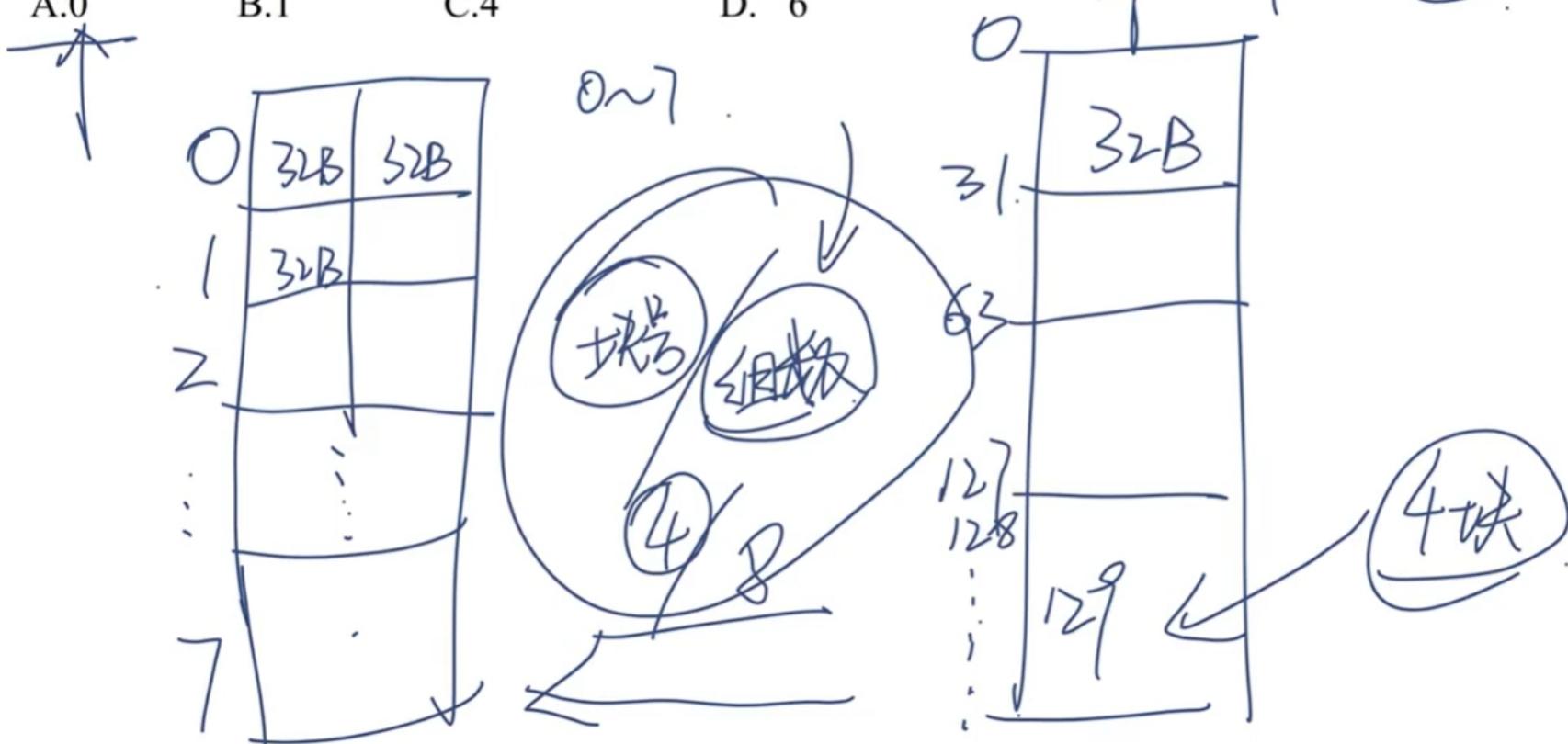
129 => 1 000 0001 -----> 0000...0000|1000 0001
 区号 块号 组号 块内地址
 厚德·博学·求是



14. 某计算机的 Cache 共有 16 块，采用 2 路组相联映射方式（即每组 2 块）。每个主存块大小为 32B，按字节编址。主存 129 号单元所在主存块应装入到的 Cache 组号是 ()。

- A. 0 B. 1 C. 4 D. 6

C



设某计算机的Cache采用**4路组相联**映像，已知Cache容量为**16KB**，主存容量为**2MB**，每个字块有**8个字**，**每个字有32位**。请回答：

- (1) 主存地址多少位(按字节编址)?各字段如何划分(各需多少位)?
- (2) 设Cache起始为空，CPU从主存单元0、1、...、100依次读出101个字(主存一次读出一个字)，并重复按此次序数读11次，问命中率为多少?若Cache速度是主存的5倍，问采用Cache与无Cache比较速度提高多少倍?

 作答

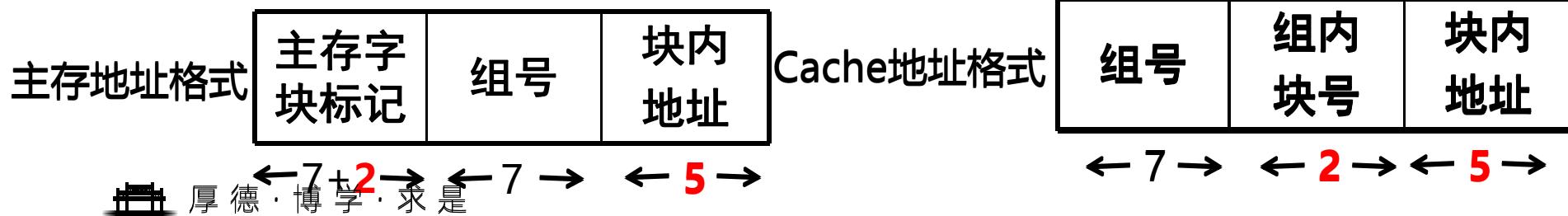


某计算机的Cache采用4路组相联映像，已知Cache容量为16KB，主存容量为2MB，每个字块有8个字，每个字有32位。请回答：

- (1) 主存地址多少位(按字节编址)?各字段如何划分(各需多少位)?
- (2) 设Cache起始为空，CPU从主存单元0、1、...、100依次读出101个字(主存一次读出一个字)，并重复按此次序数读11次，问命中率为多少?若Cache速度是主存的5倍，问采用Cache与无Cache比较速度提高多少倍?

- Cache容量为 $16\text{KB} = 2^{14}\text{B}$ ，每个字块有8个字，每个字有32位,每个字块 8×32 位 $= 8 \times 4\text{B} = 32\text{B} = 2^5\text{B}$ **块内地址5位**，4路组相联映像，每组4块，**块内地址2位**表示，每组的容量 $4 \times 2^5\text{B} = 2^7\text{B}$ ，共 $16\text{KB}/2^7\text{B} = 2^{14}/2^7\text{B} = 2^7$ 组,**组号7位表示**
- **主存容量为 $2\text{MB} = 2^{21}\text{B}$,主存共21位表示**，主存的区数为 $21 - 7 - 2 - 5 = 7$

$$2\text{MB}/16\text{KB} = 2^{21}/2^{14} = 2^7$$





某计算机的Cache采用4路组相联映像，已知Cache容量为16KB，主存容量为2MB，每个字块有8个字，每个字有32位。请回答：

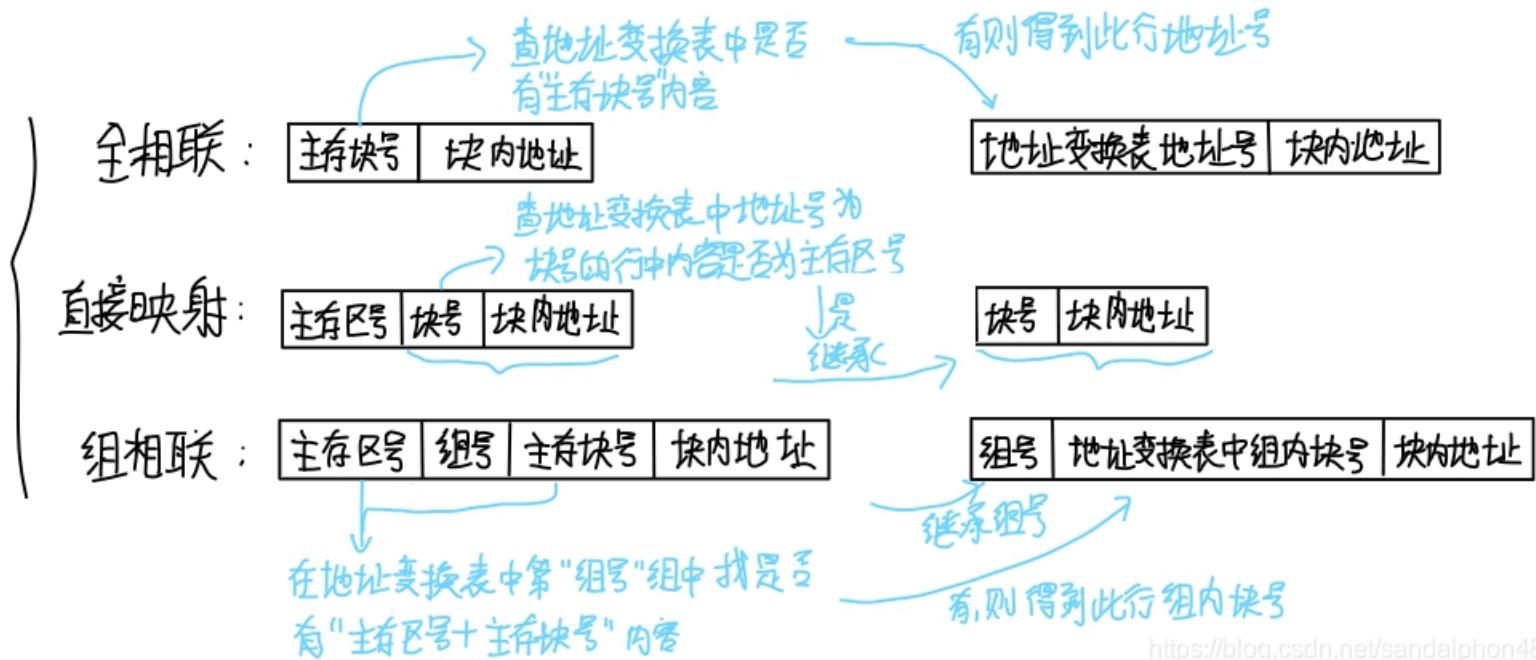
- (1) 主存地址多少位(按字节编址)?各字段如何划分(各需多少位)?
- (2) 设Cache起始为空，CPU从主存单元0、1、...、100依次读出101个字(主存一次读出一个字)，并重复按此次序数读11次，问命中率为多少?若Cache速度是主存的5倍，问采用Cache与无Cache比较速度提高多少倍?

- 每块中有8个字，101个字放入13个块中 ($101/8=12\ldots5$)
- 初态为空，所以主存读第0个单元时不命中
- 因此命中率为 $(11*101-13) / (11*101)=98.83\%$
- 设Cache读取时间为t,主存读取时间为5t,命中率为H
- 平均访问时间 $T=H*t+(1-H)*(5t+t)=98.83\%t+0.0117*6t=1.0585t$
- 有Cache和无Cache相比较速度提高了 $5t/T-1=5/1.0585-1=3.7236$





相联存储器



<https://blog.csdn.net/sandalphon4869>





替换算法

- 直接映射不需要替换算法。
- 全相联、组相联：
 - 先进先出算法(FIFO, First-in First-out Algorithm)
利用了历史信息没有反映局部性——最先调入的，
可能也是要使用的。
 - 近期最少使用算法 (LRU, Least Recently Used Algorithm)：
多/少-->有/无。 替换近期最久未用的块。





虚拟存储器

➤ 实地址与虚地址

- 随着程序占用存储容量的增长和多用户多任务的出现，在程序设计时在程序所需的存储器容量与计算机系统实际配备的主存储器的容量之间往往存在着矛盾。
- 为此，希望在编制程序时独立编址，既不考虑程序是否能在物理存储器中放得下（因为这与程序运行时的系统配置和当时其他程序的运行情况有关，在编程时一般无法确定），也不考虑程序应该存放在什么物理位置，而在程序运行时，则分配后每个程序一定的运行空间，由地址转换部件将编程时的地址转换成实际内存的物理地址。如果分配的内存不够，则只调入当前正在运行的或将要运行的程序块，其余部分暂时驻留在辅存中。



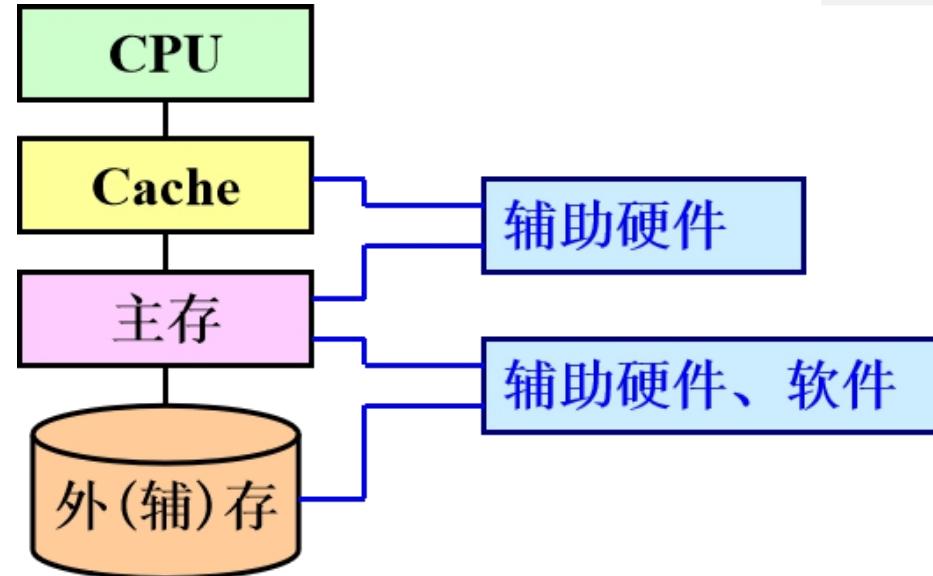


虚拟存储器

主存寻址: $2^{32} = 4\text{GB}$

➤ 虚拟存储器:

虚存: $2^{46} = 64\text{TB}$



主存储器 + 联机工作的外部存储器 + 辅助硬件 + 系统软件





虚拟存储器

➤ 虚存的访问过程

- 虚存空间的用户程序按照虚地址编程并存放在辅存中，程序运行时，由地址变换机构依据当时分配给该程序的实地址空间把程序的一部分调入实存。
- 有了虚存机制后，应用程序就可以透明地使用整个虚存空间，对应用程序而言，如果主存的命中率很高，虚存的访问时间就接近于主存访问时间，而虚存的大小，仅仅依赖于辅存的大小。
- 这样每个程序就可以拥有一个虚拟的存储器，它具有辅存的容量和接近主存的访问速度，但这个虚存是由主存和辅存以及辅存管理部件构成的概念模型，不是实际的物理存储器。
- 虚存是在主存和辅存之外附加一些硬件和软件实现的，由于软件的介入，使虚存对设计存储管理软件的系统程序员而言是不透明的，但对应用程序员而言仍然是透明的。





虚拟存储器

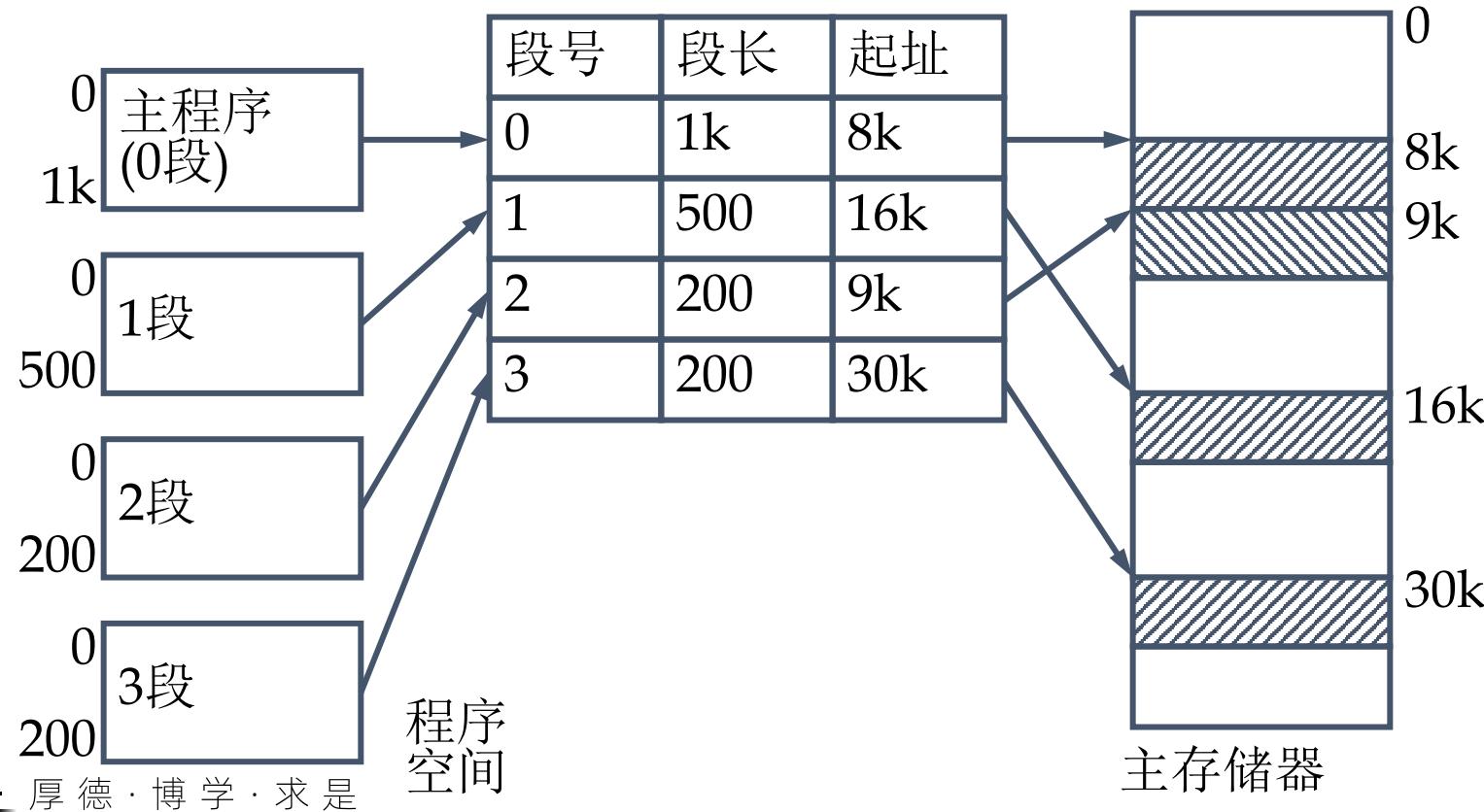
- 地址变换：
在程序运行时，把虚地址变换成主存实地址。
- 因地址映象和变换方法不同，有三种虚拟存储器：
 - 段式虚拟存储器
 - 页式虚拟存储器
 - 段页式虚拟存储器





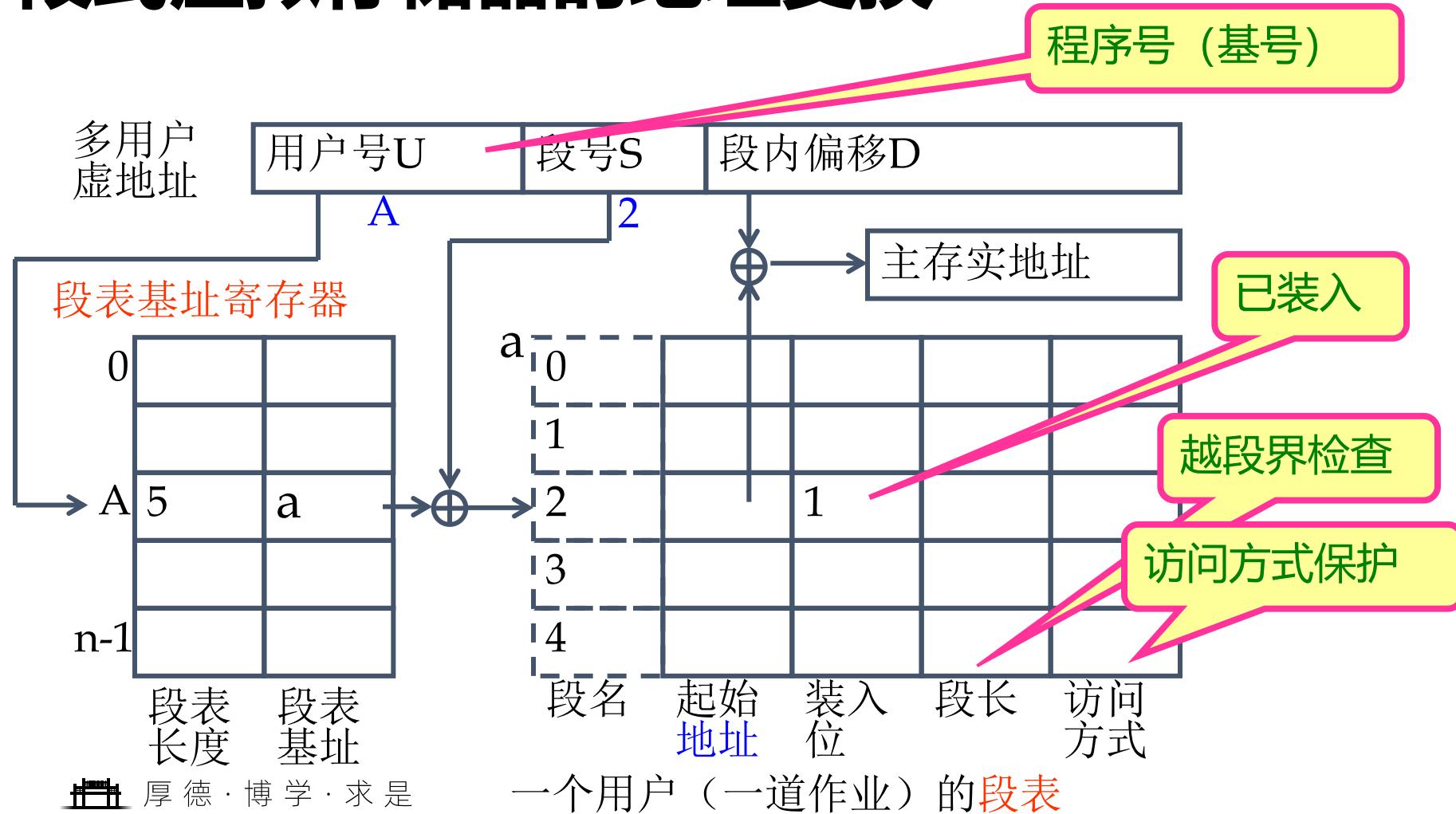
段式虚拟存储器

- 将程序按逻辑意义分成段，按段进行调入、调出和管理。



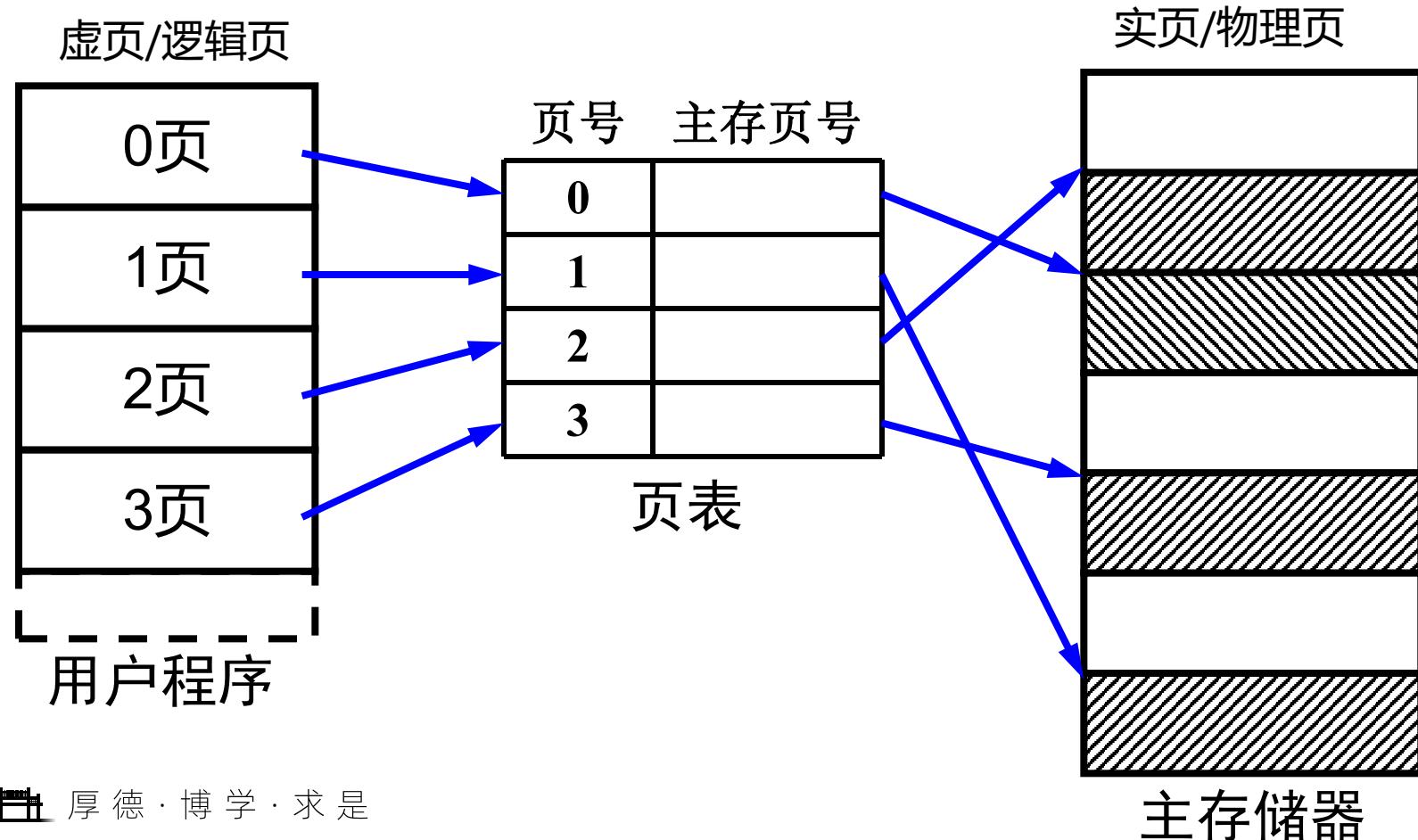


段式虚拟存储器的地址变换



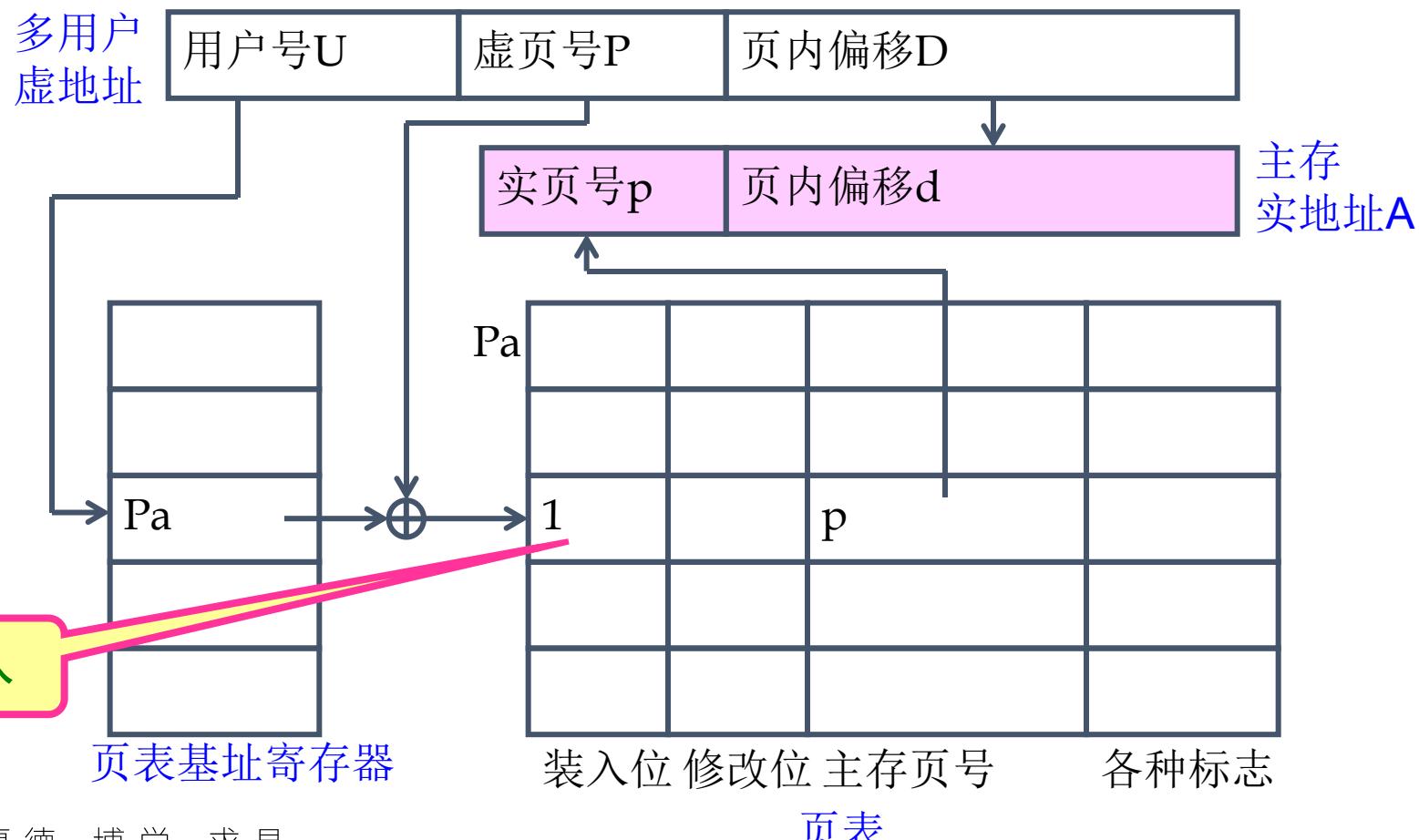


页式虚拟存储器



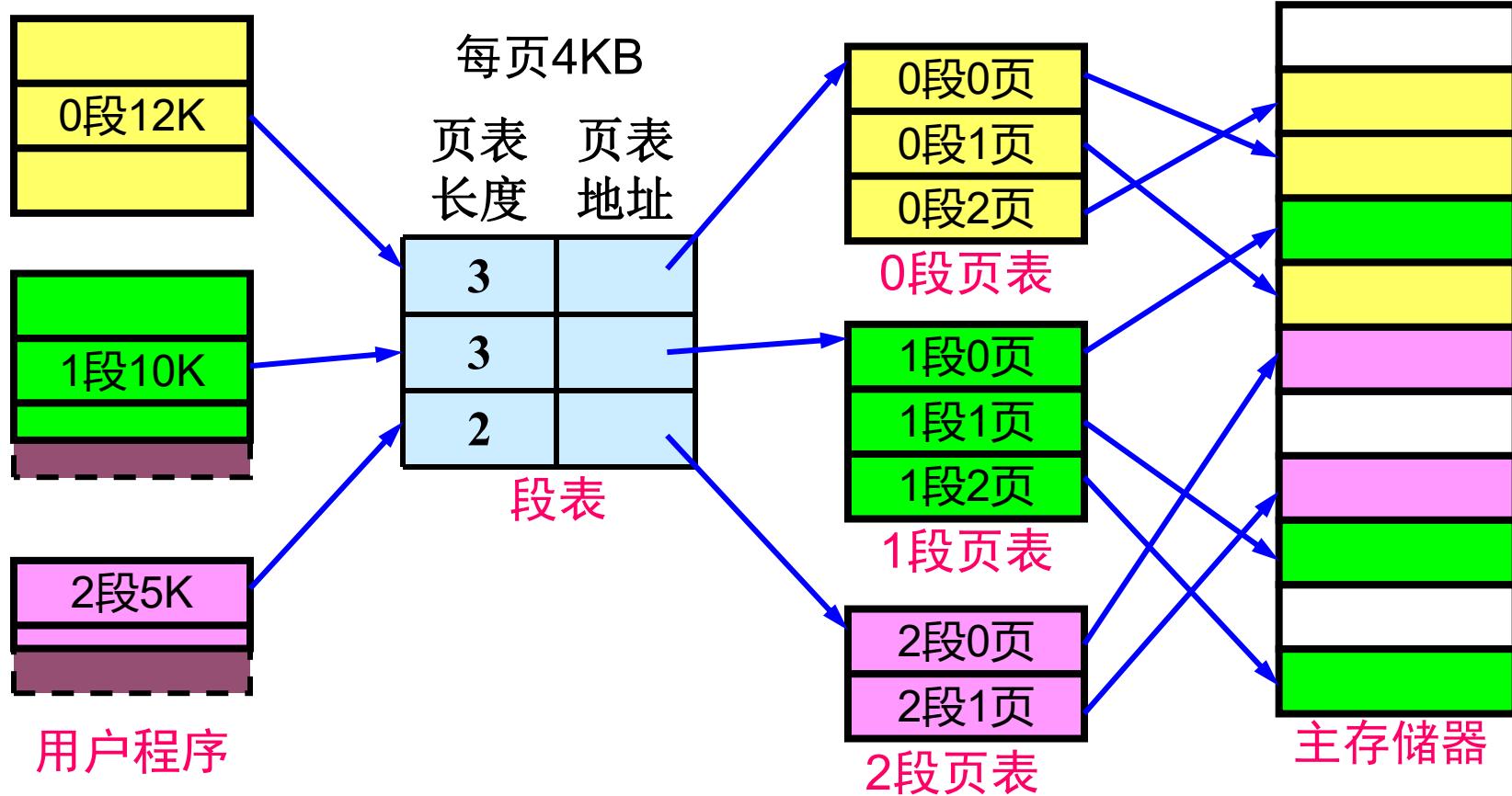


页式虚拟存储器的地址变换



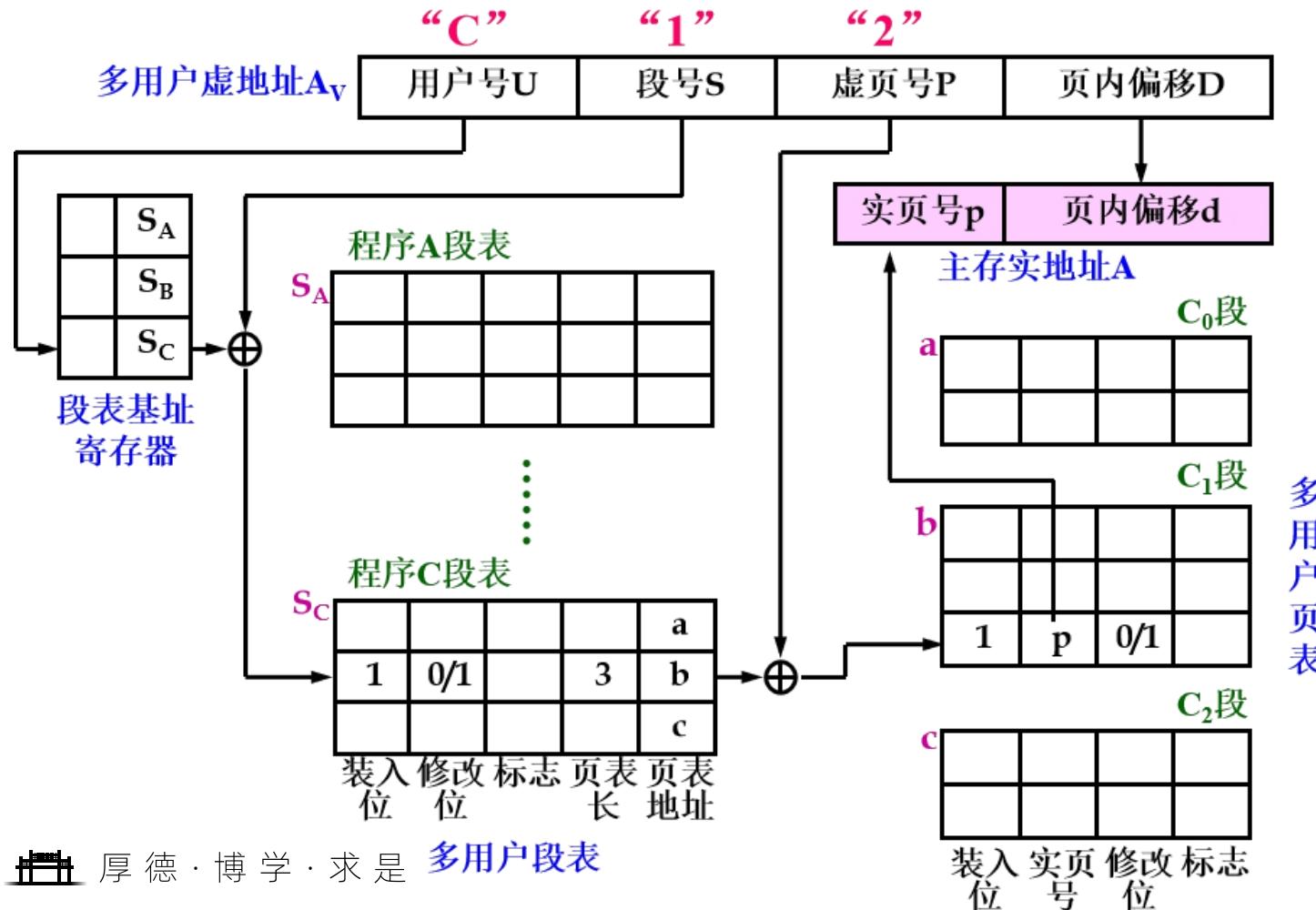


段页式虚拟存储器





段页式虚拟存储器的地址变换





Cache与虚存的异同

➤ Cache和主存之间以及主存和辅存之间分别有辅助硬件和辅助软硬件负责地址变换与管理，以便各级存储器能够组成有机的三级存储体系，Cache和主存构成了系统的内存，而主存和辅存依靠辅助软硬件的支持构成了虚拟存储器。





Cache与虚存的异同

➤ 在三级存储体系中，Cache-主存和主存-辅存这两个存储层次有许多相同点：

(1) **出发点相同**：都是为了提高存储系统的性能价格比而构造的分层存储体系，都力图使存储系统的性能接近高速存储器，而价格和容量接近低速存储器。

(2) **原理相同**：都是利用了程序运行时的**局部性原理**把最近常用的信息块从相对慢速而大容量的存储器调入相对高速而小容量的存储器。





Cache与虚存的异同

➤ Cache-主存和主存-辅存这两个存储层次不同之处：

- (1) **侧重点不同**: Cache解决主存和CPU的速度差异问题,虚存主要解决存储容量的问题。
- (2) **数据通路不同**: CPU与Cache和主存之间均可以有直接访问通路, Cache不命中时可直接访问主存, 而虚存所依赖的辅存与CPU之间不存在直接的数据通路, 当主存不命中时只能通过调页解决, CPU最终还是要访问主存。
- (3) **透明性不同**: Cache的管理完全由硬件完成, 对系统程序员和应用程序员均透明, 而虚存管理由软件(操作系统)和硬件共同完成, 由于软件的介入, 虚存对实现存储管理的系统程序员不透明, 而只对应用程序员透明。
- (4) **未命中时的损失不同**: 由于主存的存取时间是cache的存取时间的5~10倍, 而主存的存取速度通常比辅存的存取速度快上千倍, 故主存未命中时系统的性能损失要远大于cache未命中的损失。





虚拟存储器，TLB和Cache的协同操作

- 虚拟存储器和Cache系统如同一个层结构般一起工作
- 在最好的情况下，虚拟地址由TLB进行转换，然后被送到Cache，找到正确的数据并取回处理器，在最坏的情况下，一次访问会在存储器层次结构的三个组成部分都产生缺乏：TLB，页表和Cache。





虚拟存储器，TLB和Cache的协同操作

- 例：由页式虚拟存储器，TLB和cache组成的存储器层次结构中，访问存储器可能会遇到三种不同类型的缺失：Cache缺失、TLB缺失和缺页。Cache缺失、TLB缺失和缺页，研究这三种缺失会发生一个或多个时的所有可能的组合（7种可能性）。对每种可能性，说明这种情况是否真的会发生，在什么条件下发生？

TLB	页表	Cache	可能发生吗？如可能，发生的背景是什么
命中	命中	缺失	
缺失	命中	命中	
缺失	命中	缺失	
缺失	缺失	缺失	
命中	缺失	缺失	
命中	缺失	命中	
缺失	缺失	命中	





虚拟存储器，TLB和Cache的协同操作

- 例：由页式虚拟存储器，TLB和cache组成的存储器层次结构中，访问存储器可能会遇到三种不同类型的缺失：Cache缺失、TLB缺失和缺页。Cache缺失、TLB缺失和缺页，研究这三种缺失会发生一个或多个时的所有可能的组合（7种可能性）。对每种可能性，说明这种情况是否真的会发生，在什么条件下发生？

TLB	页表	Cache	可能发生吗？如可能，发生的背景是什么
命中	命中	缺失	可能
缺失	命中	命中	可能
缺失	命中	缺失	可能
缺失	缺失	缺失	可能
命中	缺失	缺失	不可能，如果页不在内存中，TLB不可能命中
命中	缺失	命中	不可能，如果页不在内存中，TLB不可能命中
缺失	缺失	命中	不可能，如果页不在内存中，数据可能在Cache中存在





有问题欢迎随时跟我讨论

办公地点：西校区信息馆423

邮 箱：gddu@ysu.edu.cn

