

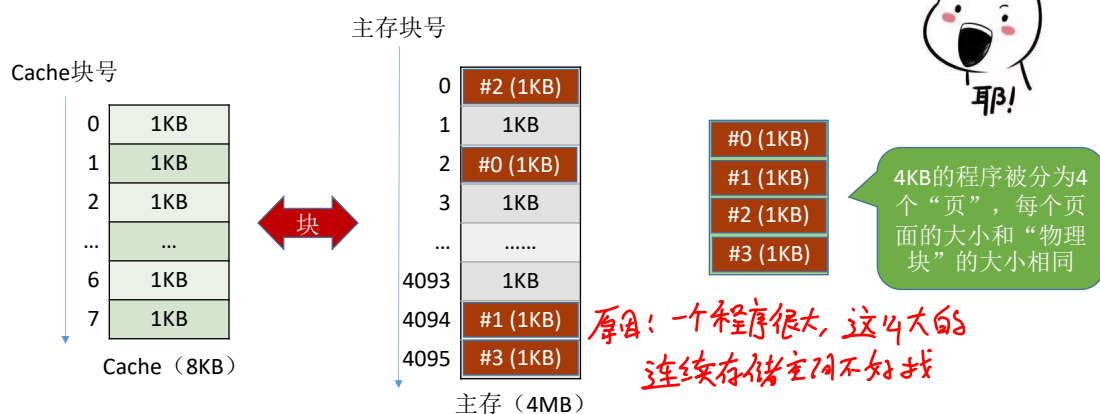
本节内容

页式 存储器

王道考研/CSKAOYAN.COM

1

页式存储




页式存储系统：一个程序(进程)在逻辑上被分为若干个大小相等的“页面”，“页面”大小与“块”的大小相同。每个页面可以离散地放入不同的主存块中。

王道考研/CSKAOYAN.COM

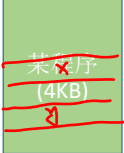
2

虚地址 vs 实地址

取变量 x 至 ACC 寄存器
 机器指令: **000001 001000000011**
 操作码+地址码 (使用逻辑地址)



程序, 搞起来很轻松的就是头冷



某程序 (4KB)

操作系统将程序分“页”

逻辑页号

| |
|----------|
| #0 (1KB) |
| #1 (1KB) |
| #2 (1KB) |
| #3 (1KB) |

若干位

页内地址

| |
|-----|
| 10位 |
|-----|

10位

主存 (4MB)

主存块号

| | |
|------|----------|
| 0 | #2 (1KB) |
| 1 | 1KB |
| 2 | #0 (1KB) |
| 3 | 1KB |
| ... | |
| 4093 | 1KB |
| 4094 | #1 (1KB) |
| 4095 | #3 (1KB) |

主存 (4MB)

主存的物理地址共22位:

| 主存块号 | 块内地址 |
|------|------|
| 12位 | 10位 |

32位物理地址

变量 x 的逻辑地址: **001000000011**

变量 y 的逻辑地址: **1100000001010**

变量 x 的物理地址: **0000000000101000000011**

变量 y 的物理地址: **11111111111100000001010**

王道考研/CSKAOYAN.COM

3

页表: 逻辑页号 → 主存块号

取变量 x 至 ACC 寄存器
 机器指令: **000001 001000000011**
 操作码+地址码 (使用逻辑地址)

逻辑页号

| |
|----|
| #0 |
| #1 |
| #2 |
| #3 |

若干位

主存块号

| |
|------|
| 2 |
| 4094 |
| 0 |
| 4095 |

10位

页表

页表数据在主存里

操作系统将程序分“页”

主存块号

| | |
|------|----------|
| 0 | #2 (1KB) |
| 1 | 1KB |
| 2 | #0 (1KB) |
| 3 | 1KB |
| ... | |
| 4093 | 1KB |
| 4094 | #1 (1KB) |
| 4095 | #3 (1KB) |

主存 (4MB)

主存的物理地址共22位:

| 主存块号 | 块内地址 |
|------|------|
| 12位 | 10位 |

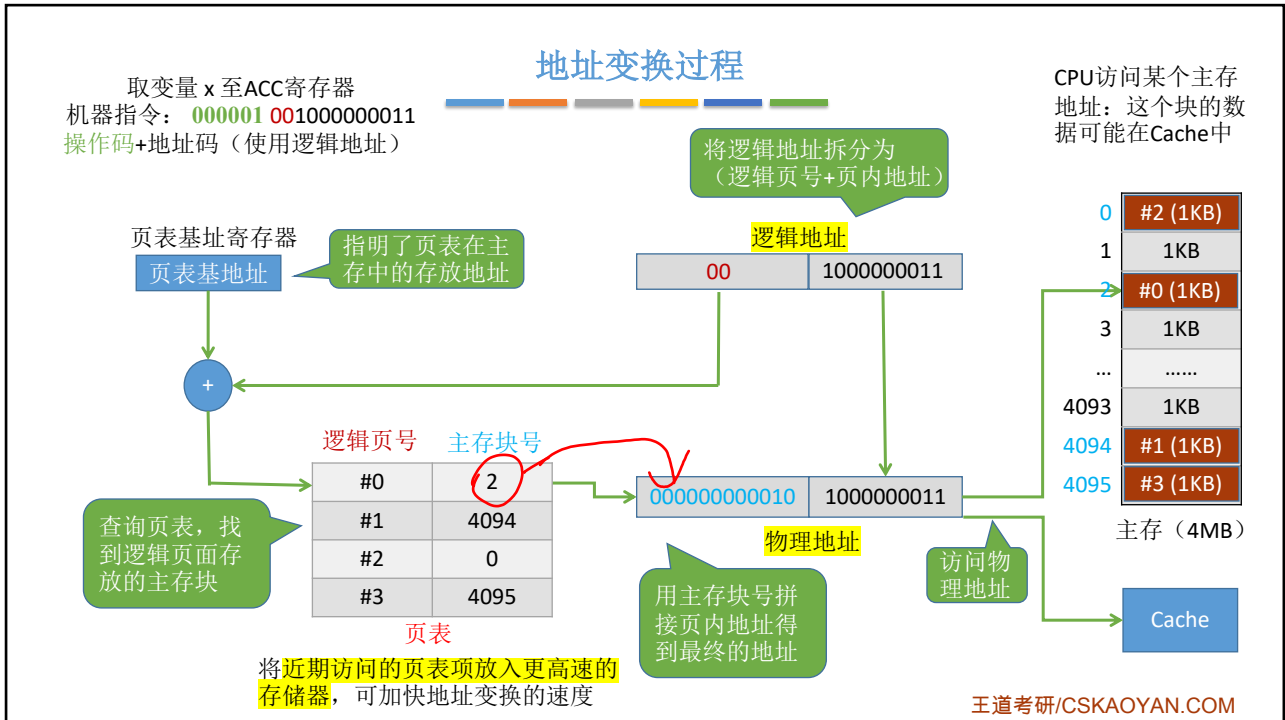
32位物理地址

变量 x 的逻辑地址: **001000000011**

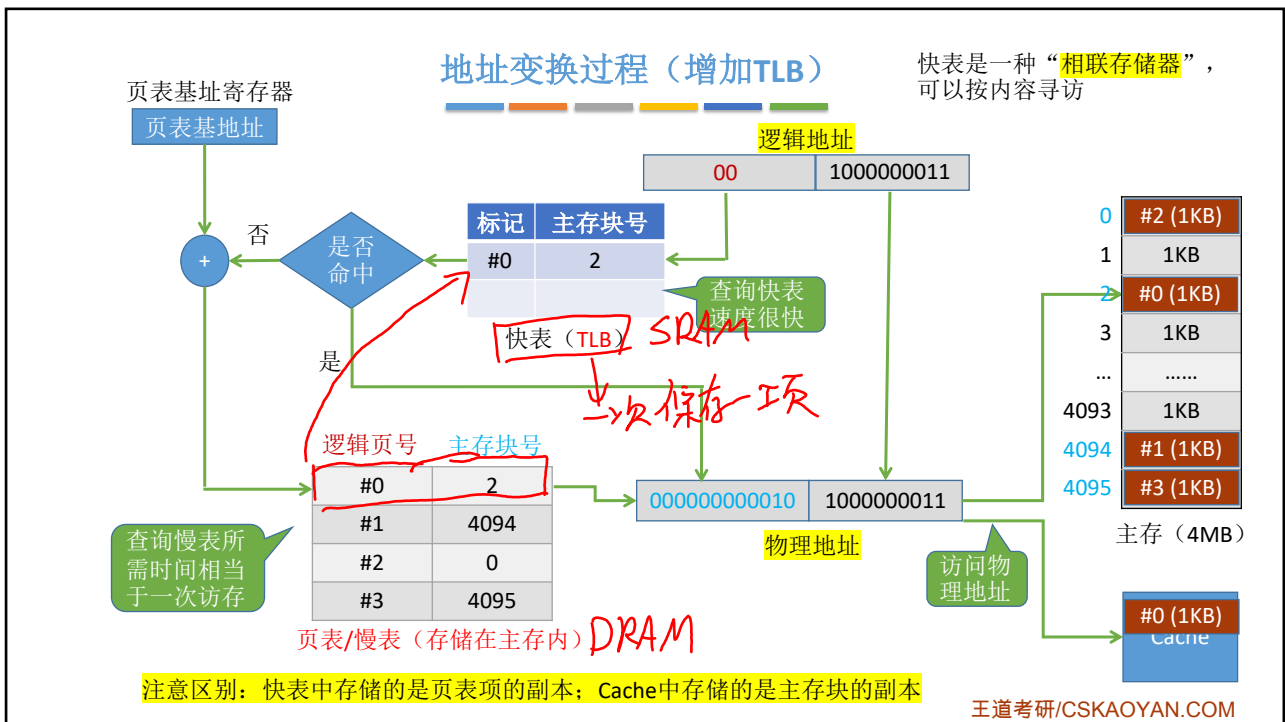
变量 x 的物理地址: **0000000000101000000011**

王道考研/CSKAOYAN.COM

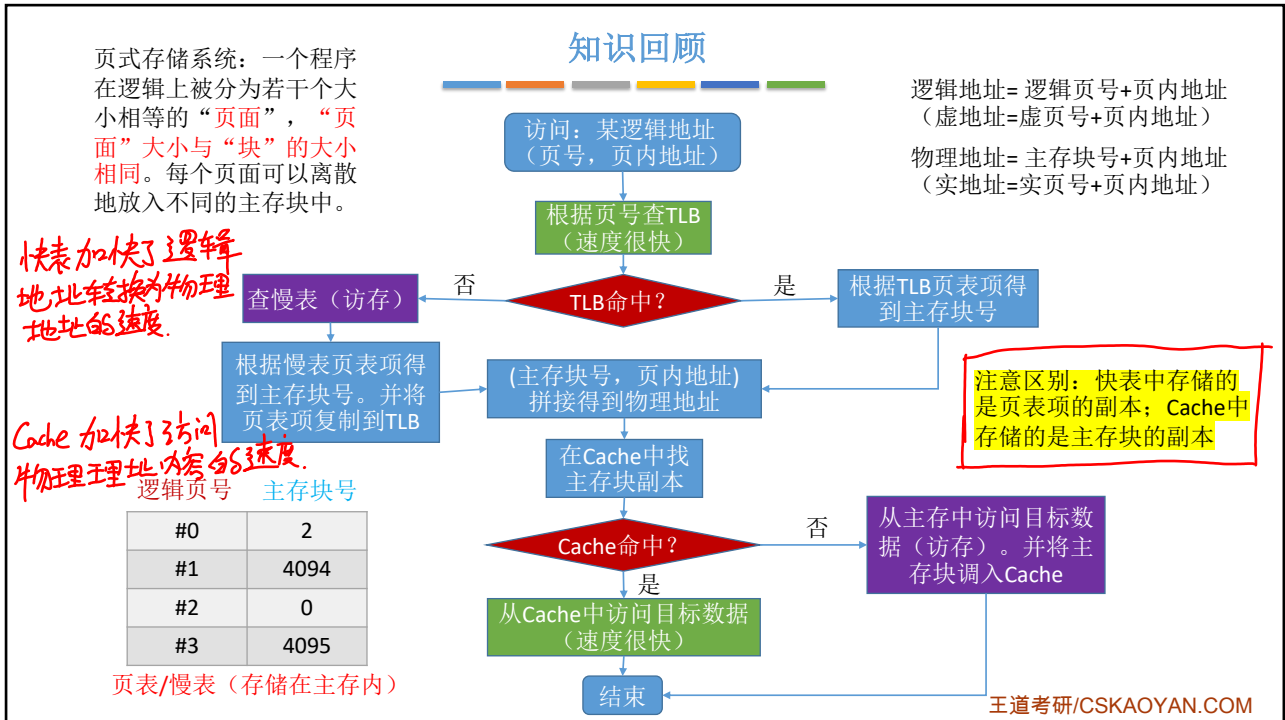
4



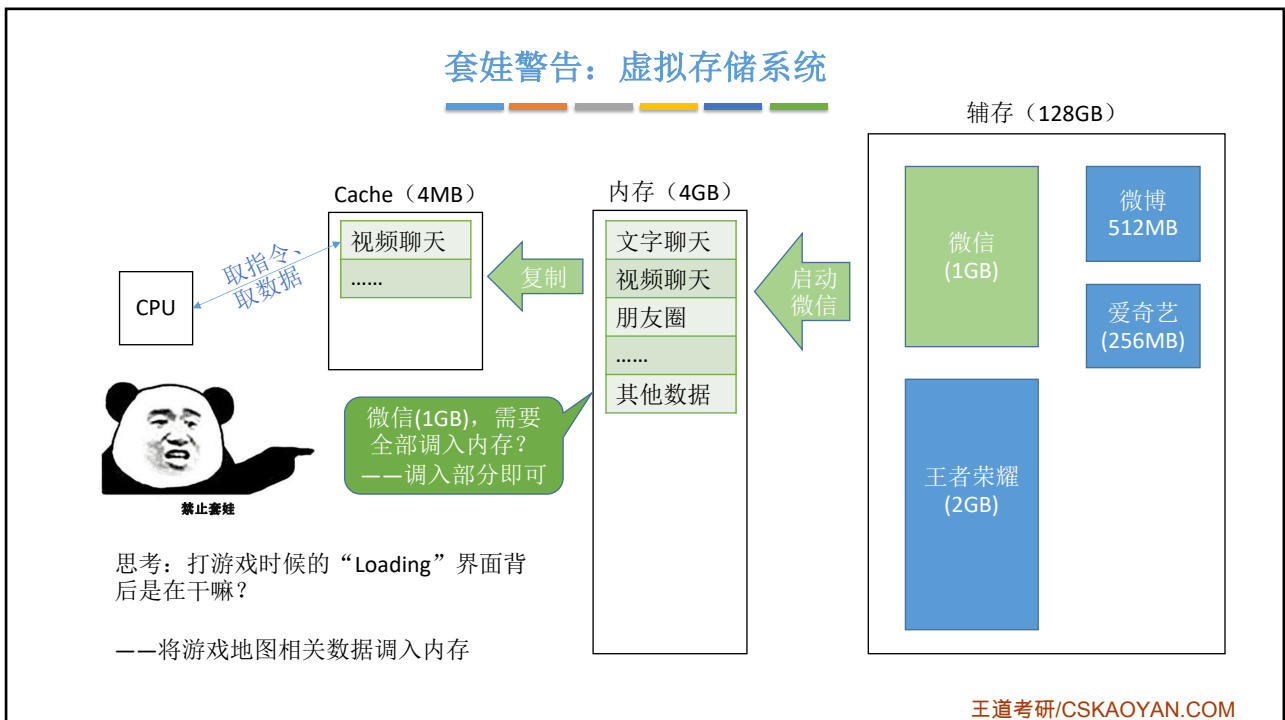
5



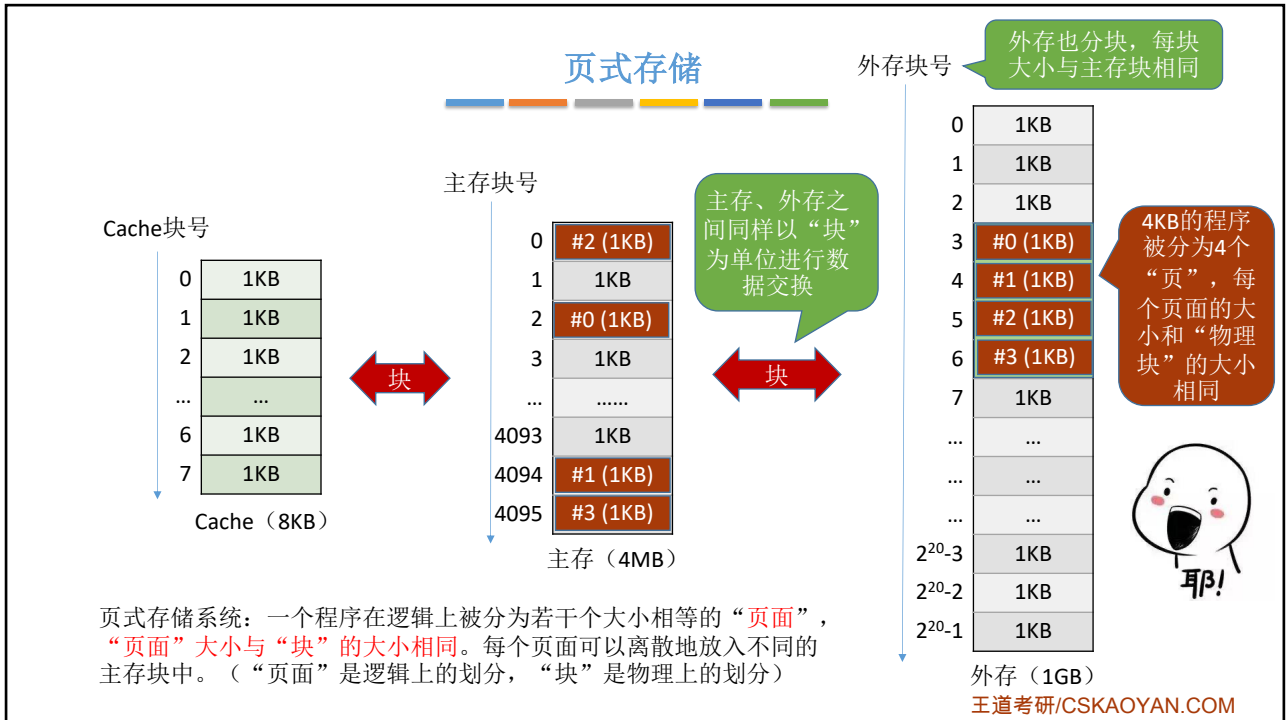
6



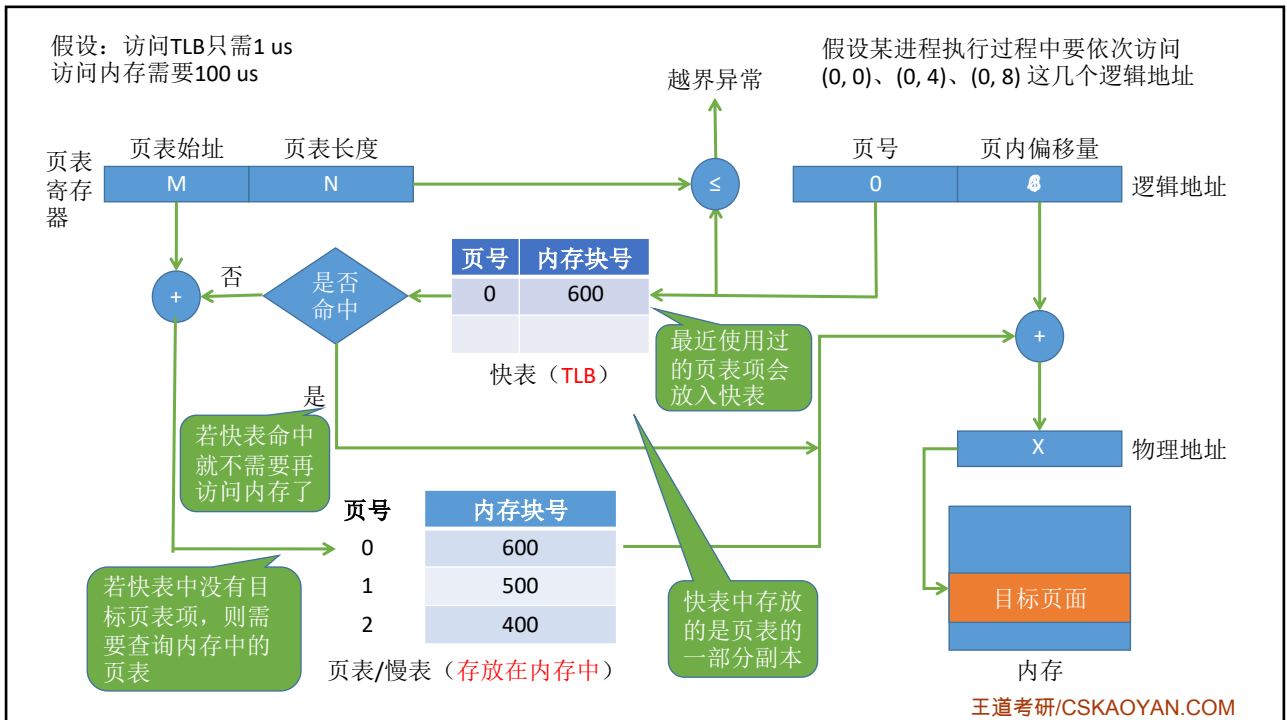
7



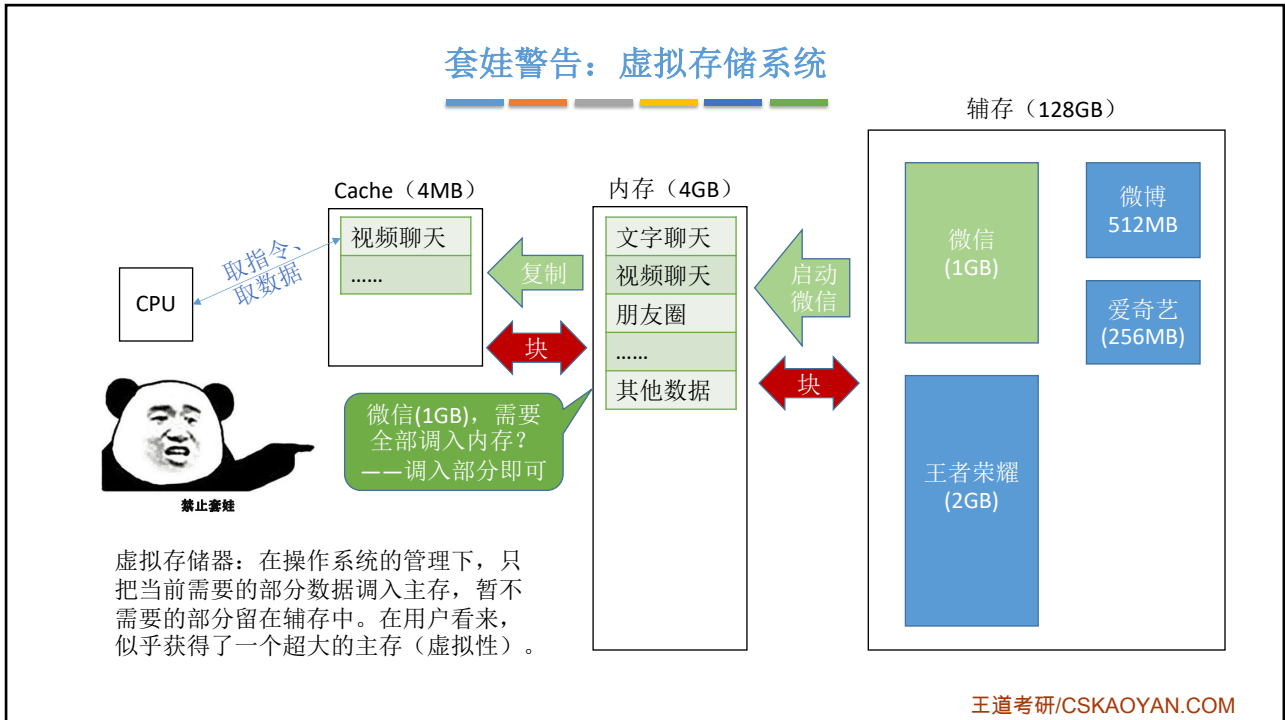
8



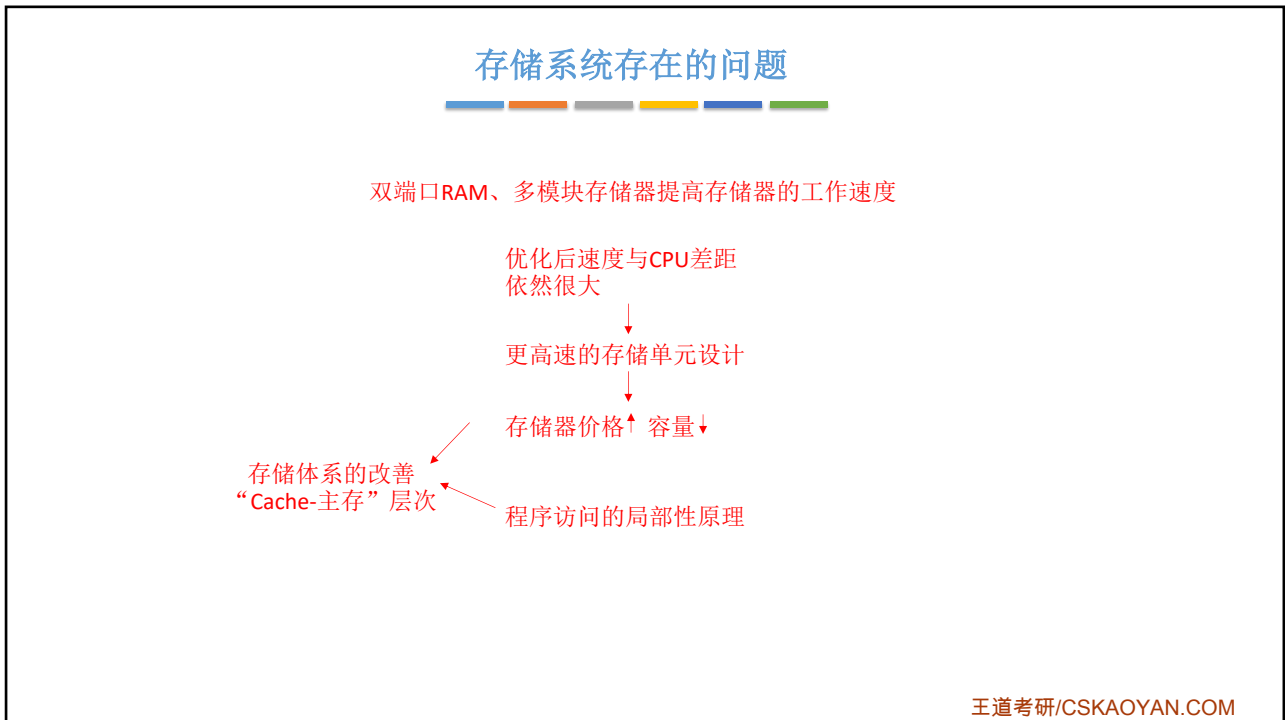
9



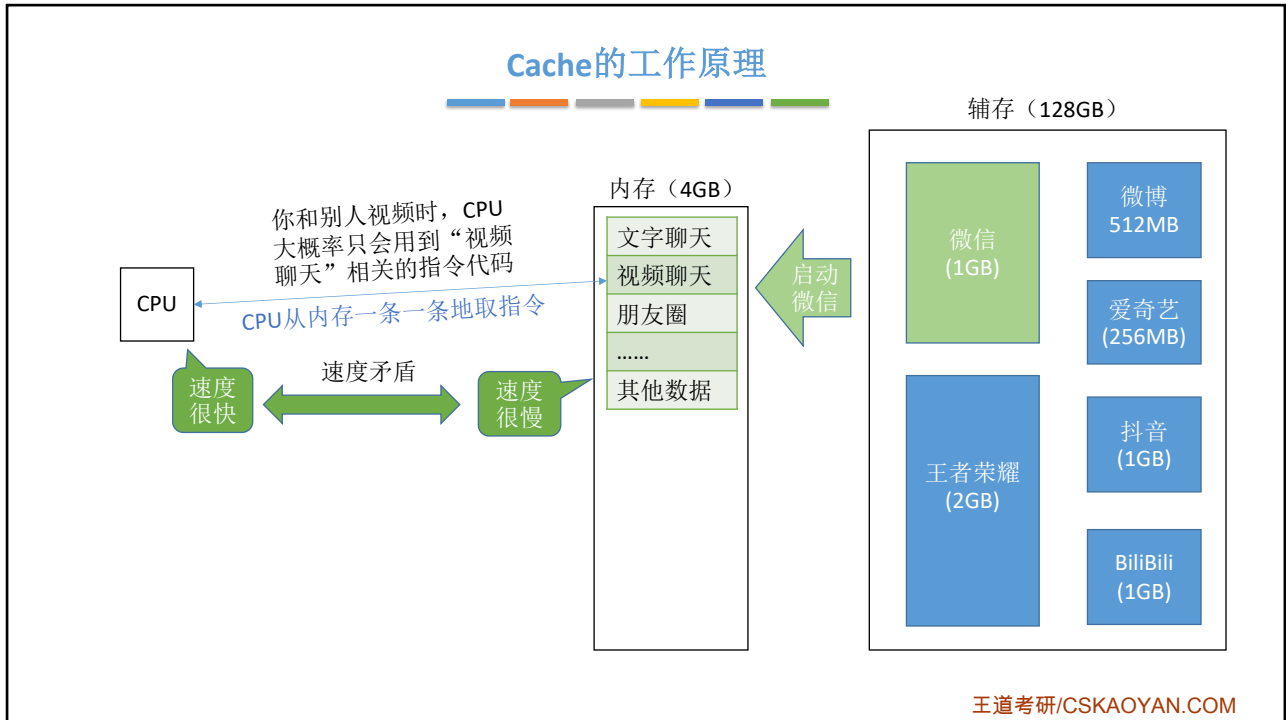
10



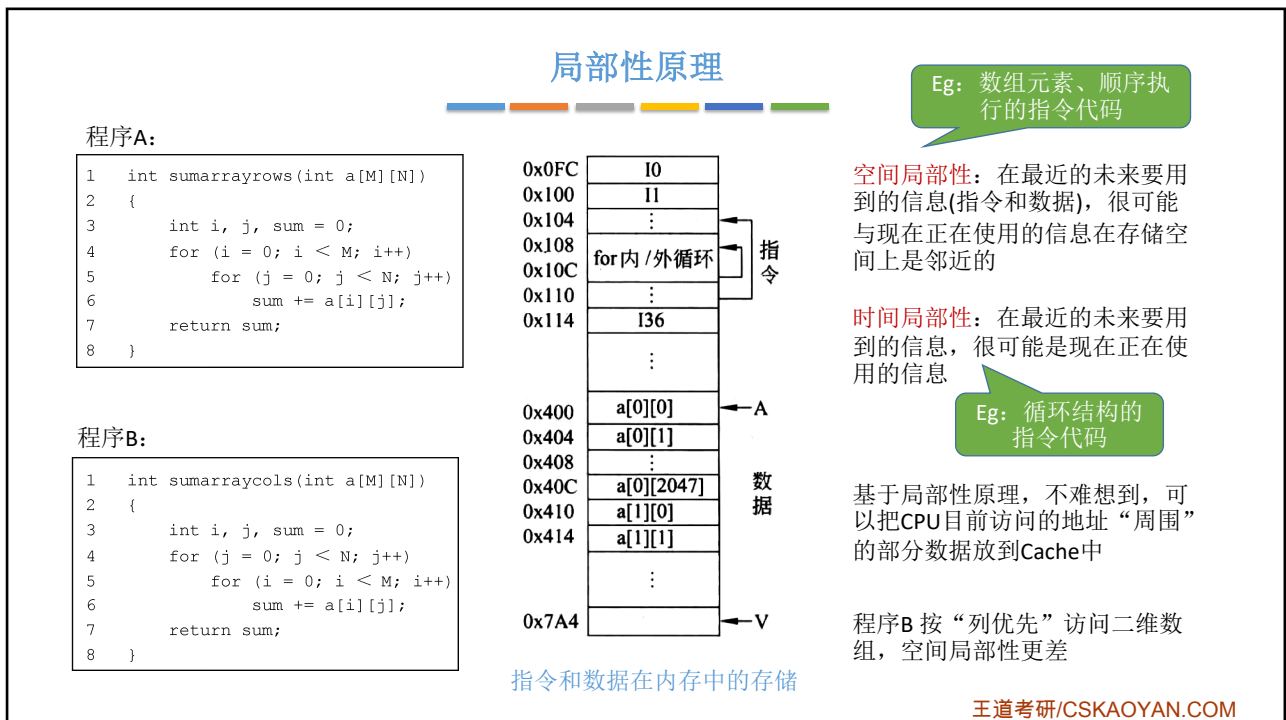
11



12

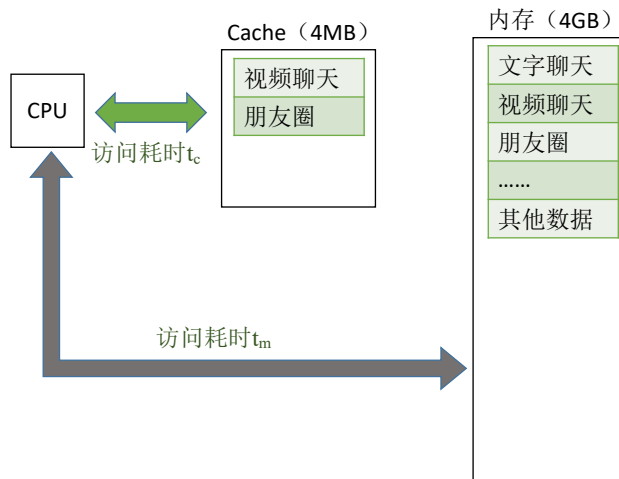


13



14

性能分析



设 t_c 为访问一次Cache所需时间， t_m 为访问一次主存所需时间

命中率 H : CPU欲访问的信息已在Cache中的比率

缺失（未命中）率 $M = 1 - H$

Cache—主存 系统的 **平均访问时间 t** 为

$$t = Ht_c + (1 - H)(t_c + t_m)$$

先访问Cache，若Cache未命中再访问主存

或 $t = Ht_c + (1 - H)t_m$

同时访问 Cache和主存，若Cache命中则立即停止访问主存

王道考研/CSKAOYAN.COM

15

性能分析

【例3-2】 假设Cache的速度是主存的5倍，且Cache的命中率为95%，则采用Cache后，存储器性能提高多少（设Cache和主存同时被访问，若Cache命中则中断访问主存）？

设Cache的存取周期为 t ，则主存的存取周期为 $5t$

若Cache和主存同时访问，命中时访问时间为 t ，未命中时访问时间为 $5t$
 平均访问时间为 $0.95 \times t + 0.05 \times 5t = 1.2t$

故性能为原来的 $\frac{5t}{1.2t} \approx 4.17$ 倍

若先访问Cache再访问主存，命中时访问时间为 t ，未命中时访问时间为 $t + 5t$
 平均访问时间为 $T_a = 0.95 \times t + 0.05 \times 6t = 1.25t$

故性能为原来的 $\frac{5t}{1.25t} = 4$ 倍

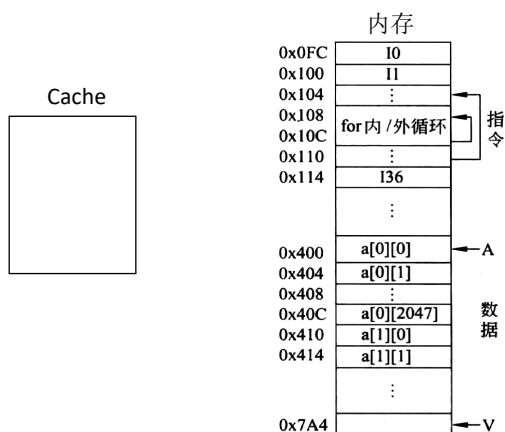
王道考研/CSKAOYAN.COM

16

有待解决的问题

基于局部性原理，不难想到，可以把CPU目前访问的地址“周围”的部分数据放到Cache中。如何界定“周围”？

将主存的存储空间“分块”，如：每 1KB 为一块。主存与Cache之间以“块”为单位进行数据交换



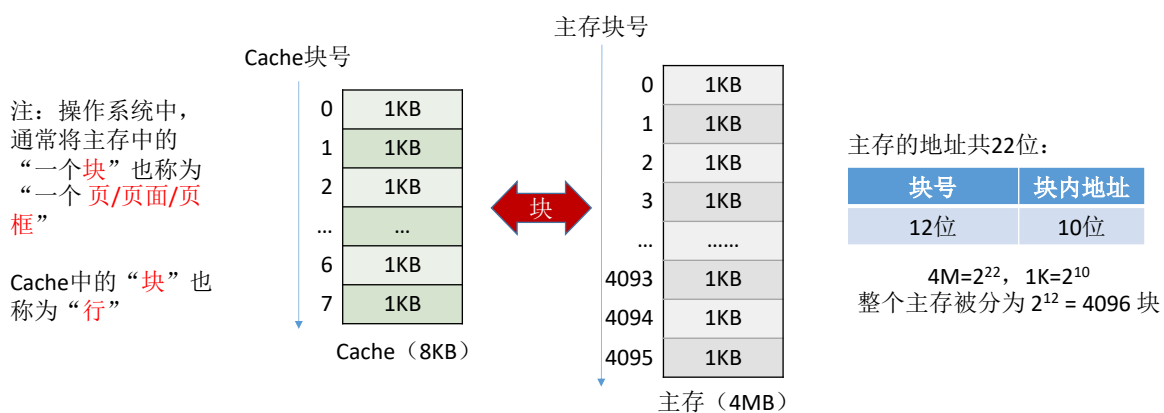
王道考研/CSKAOYAN.COM

17

有待解决的问题

基于局部性原理，不难想到，可以把CPU目前访问的地址“周围”的部分数据放到Cache中。如何界定“周围”？

将主存的存储空间“分块”，如：每 1KB 为一块。主存与Cache之间以“块”为单位进行数据交换



王道考研/CSKAOYAN.COM

18

有待解决的问题

Cache块号

| | |
|-----|-----|
| 0 | 1KB |
| 1 | 1KB |
| 2 | 1KB |
| ... | ... |
| 6 | 1KB |
| 7 | 1KB |

Cache (8KB)

主存块号

| | |
|------|-------|
| 0 | 1KB |
| 1 | 1KB |
| 2 | 1KB |
| 3 | 1KB |
| ... | |
| 4093 | 1KB |
| 4094 | 1KB |
| 4095 | 1KB |

主存 (4MB)

块

注意：每次被访问的主存块，一定会被立即调入Cache

主存的地址共22位：

| 块号 | 块内地址 |
|-----|------|
| 12位 | 10位 |

$4M=2^{22}$, $1K=2^{10}$
整个主存被分为 $2^{12} = 4096$ 块

如何区分 Cache 与 主存 的数据块对应关系？
 Cache 很小，主存很大。如果Cache满了怎么办？
 CPU修改了Cache中的数据副本，如何确保主存中数据母本的一致性？

——Cache和主存的映射方式
 ——替换算法
 ——Cache写策略

王道考研/CSKAOYAN.COM

19

知识回顾

高速缓冲存储器
Cache

- 工作原理 ⊖ 将某些主存块复制到Cache中，缓和CPU与主存之间的速度矛盾
- 局部性原理 ⊖
 - 时间局部性 ⊖ 现在访问的地址，不久之后也很可能被再次访问
 - 空间局部性 ⊖ 现在访问的地址，其附近的地址也很可能即将被访问
- 性能分析 ⊖
 - 理解Cache命中率、缺失率
 - 两种方式 ⊖
 - 先访问Cache，发现未命中再访问主存
 - 同时访问Cache和主存，若Cache命中则停止访问主存
- 其他概念 ⊖
 - 主存与Cache之间以“块”为单位进行数据交换
 - 主存的“块”又叫“页/页框/页面”；Cache的“块”又叫“行”
 - 主存地址可拆分为（主存块号，块内地址）的形式

每次被访问的主存块，一定会被立即调入Cache

如何区分 Cache 与 主存 的数据块对应关系？
 Cache 很小，主存很大。如果Cache满了怎么办？
 CPU修改了Cache中的数据副本，如何确保主存中数据母本的一致性？

——Cache和主存的映射方式
 ——替换算法
 ——Cache写策略

王道考研/CSKAOYAN.COM

20



@王道论坛



等撩

@王道计算机考研备考

@王道咸鱼老师-计算机考研

@王道楼楼老师-计算机考研



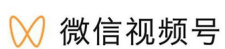
@王道计算机考研



等撩



@王道计算机考研



@王道计算机考研



@王道在线