

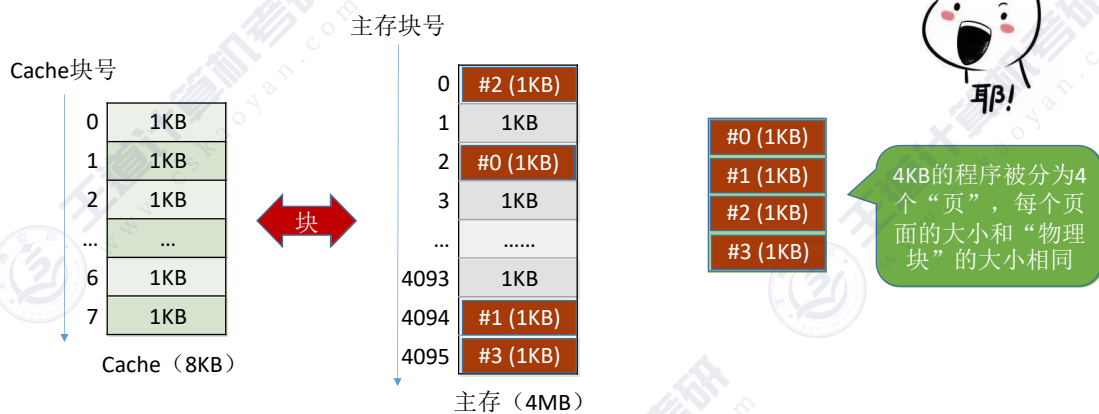
本节内容

# 页式 存储器

王道考研/CSKAOYAN.COM

1

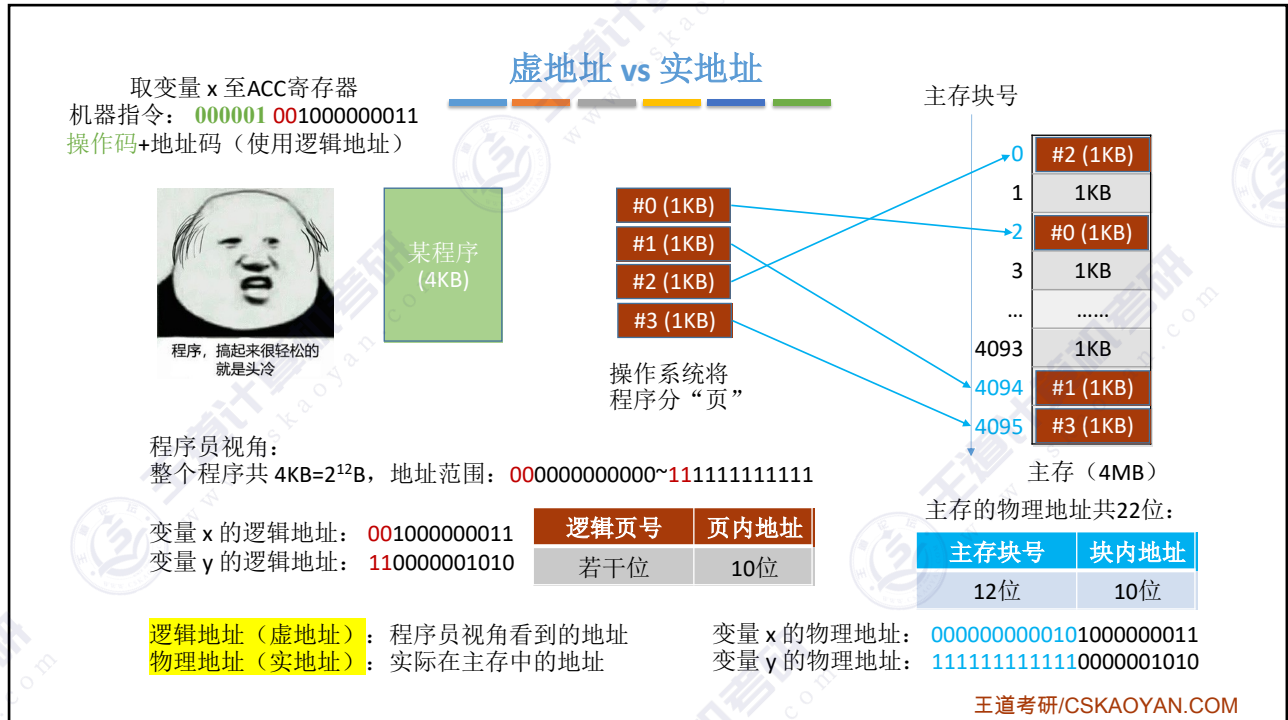
## 页式存储



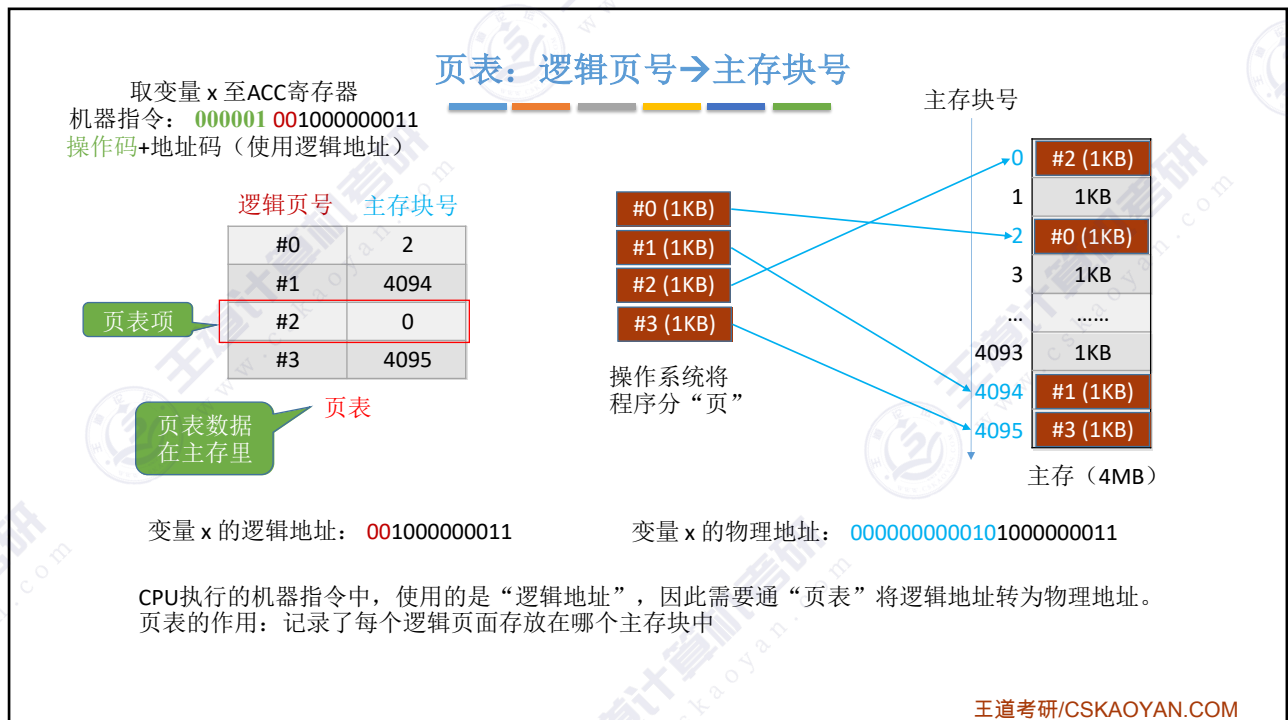
页式存储系统：一个程序(进程)在逻辑上被分为若干个大小相等的“**页面**”，“**页面**”大小与“**块**”的大小相同。每个页面可以离散地放入不同的主存块中。

王道考研/CSKAOYAN.COM

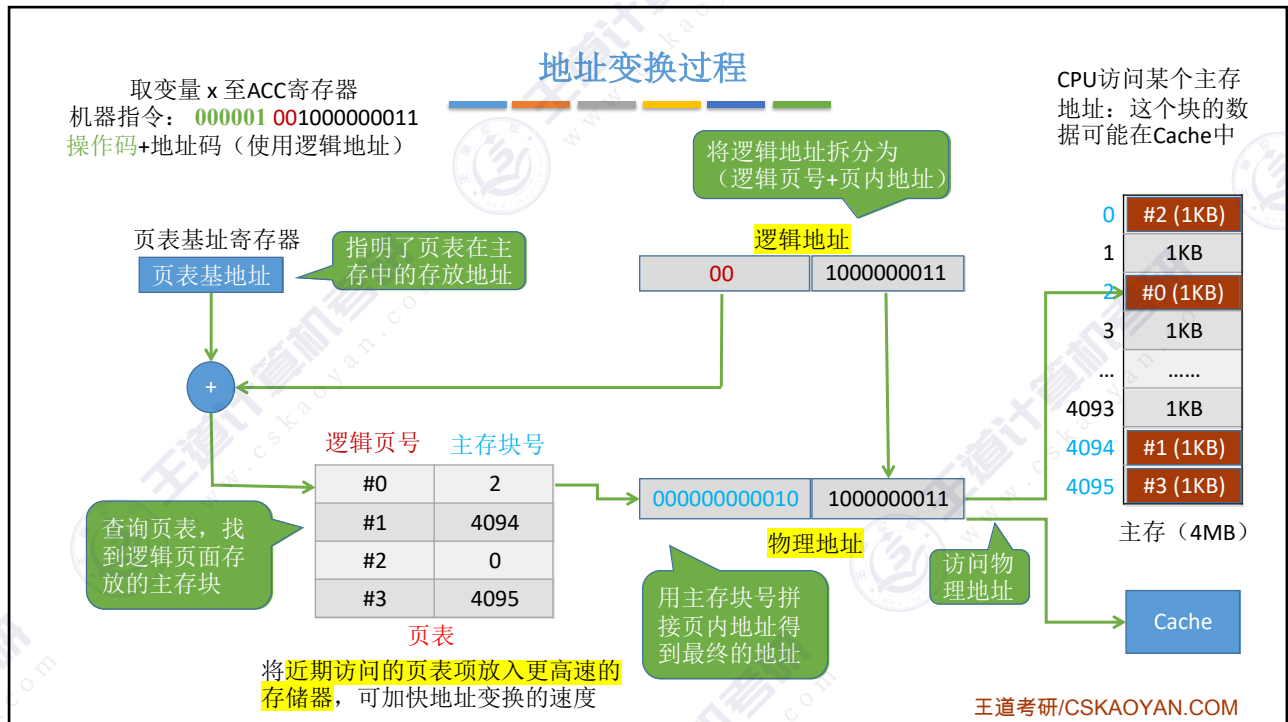
2



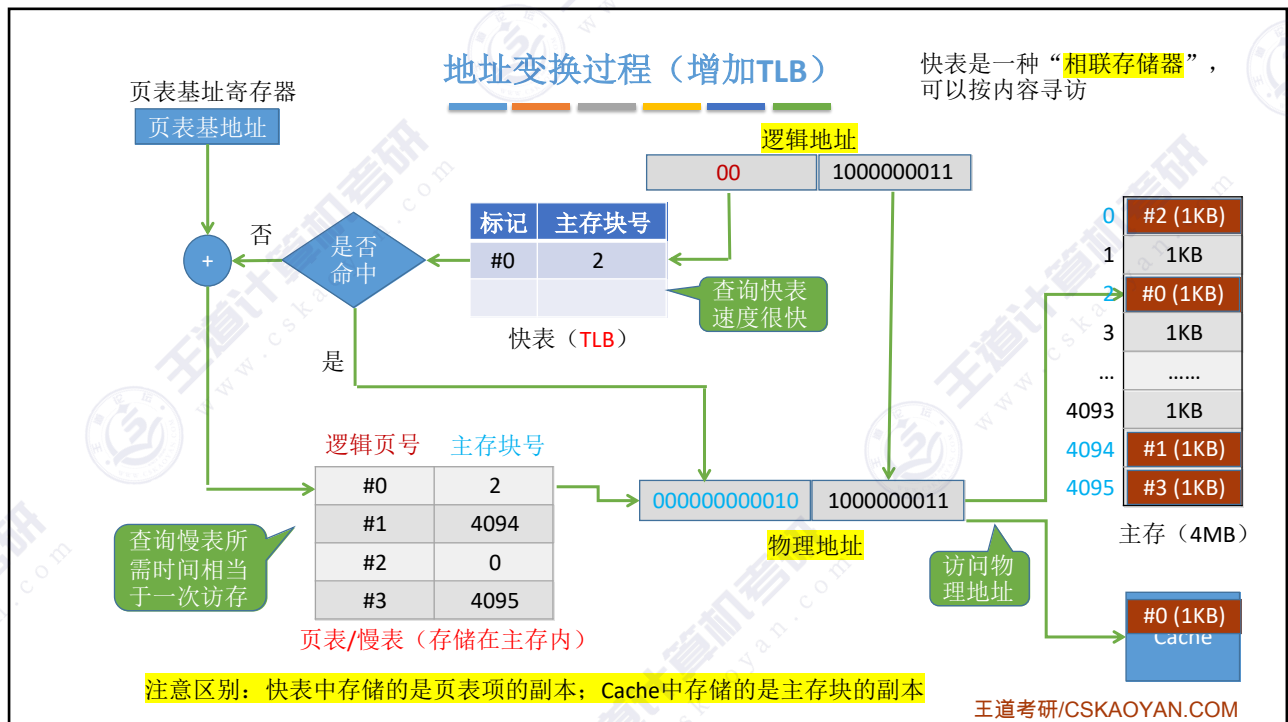
3



4



5

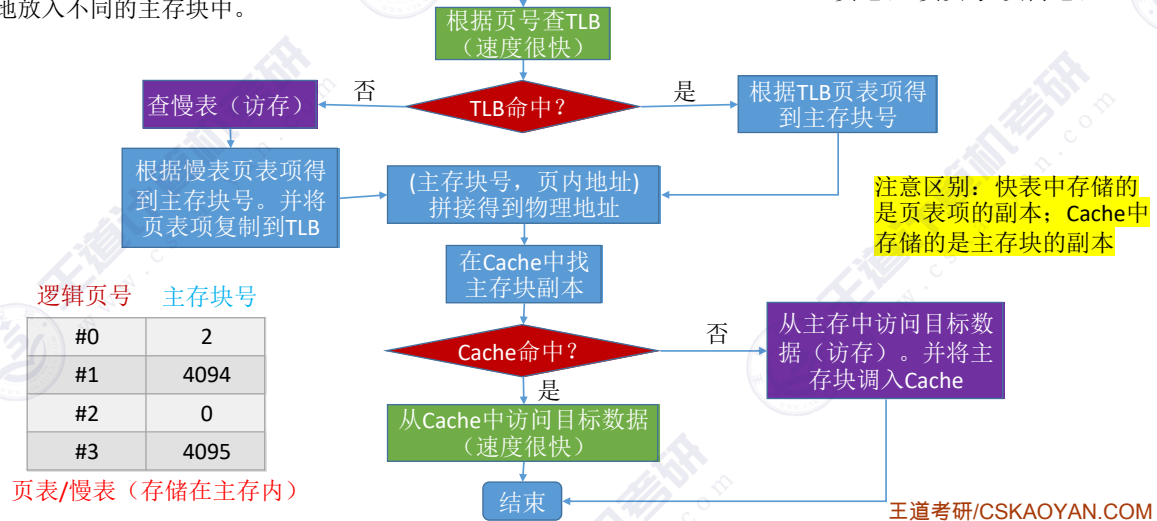


6

## 知识回顾

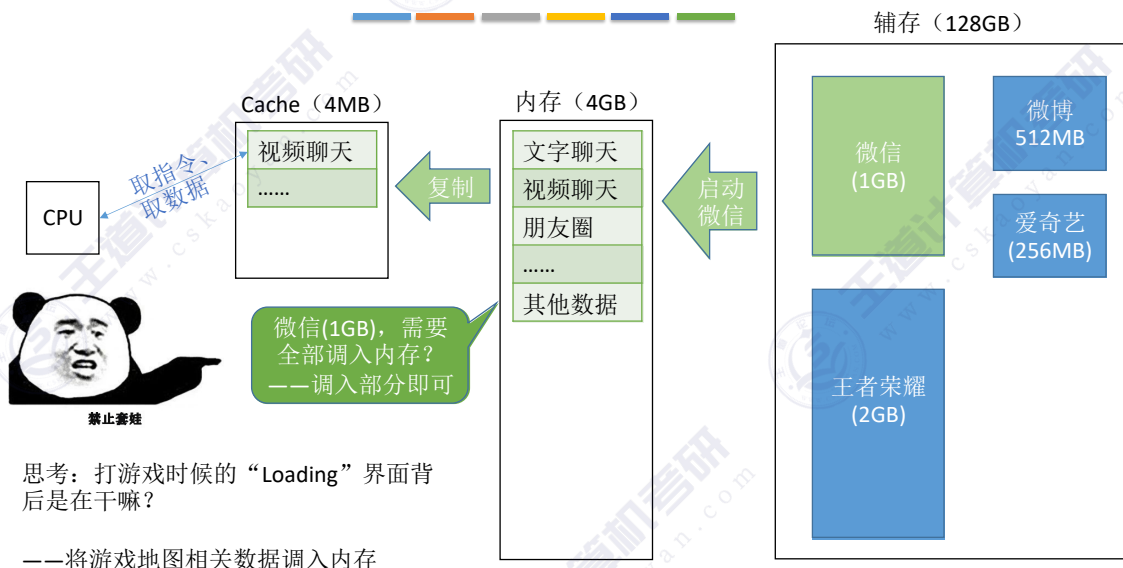
页式存储系统：一个程序在逻辑上被分为若干个大小相等的“页面”，“页面”大小与“块”的大小相同。每个页面可以离散地放入不同的主存块中。

逻辑地址 = 逻辑页号 + 页内地址  
(虚地址 = 虚页号 + 页内地址)  
物理地址 = 主存块号 + 页内地址  
(实地址 = 实页号 + 页内地址)

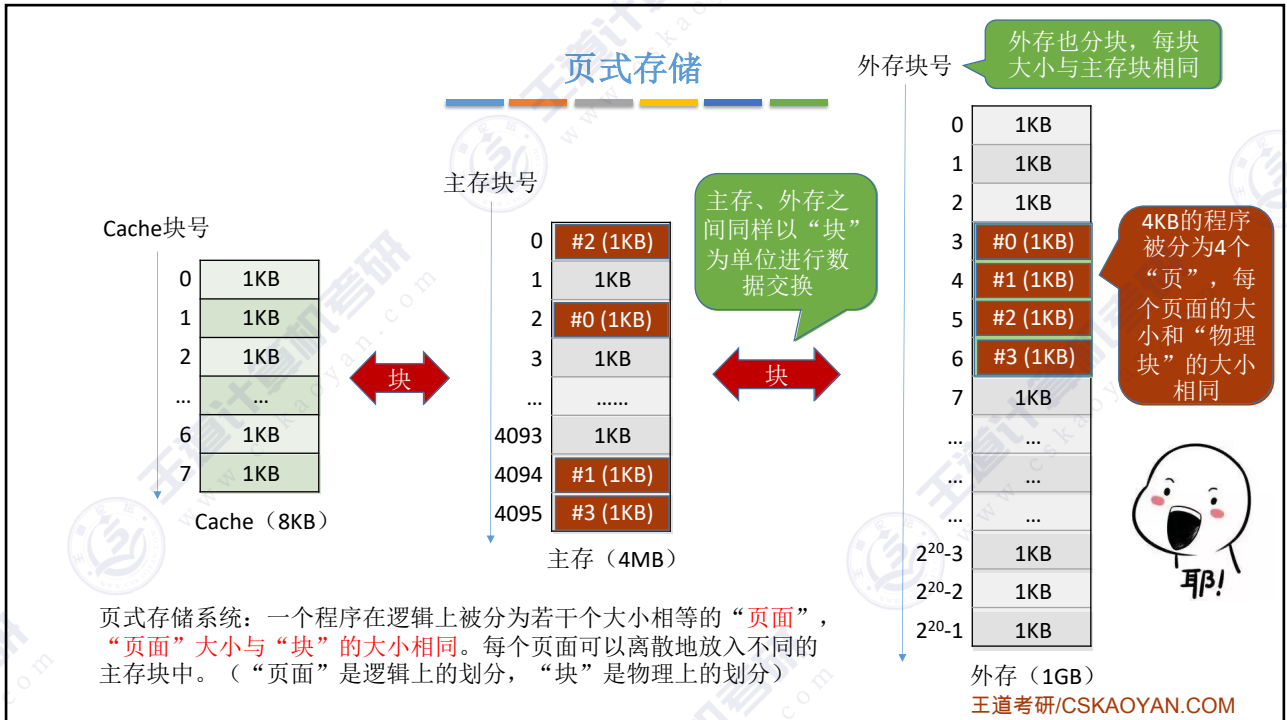


7

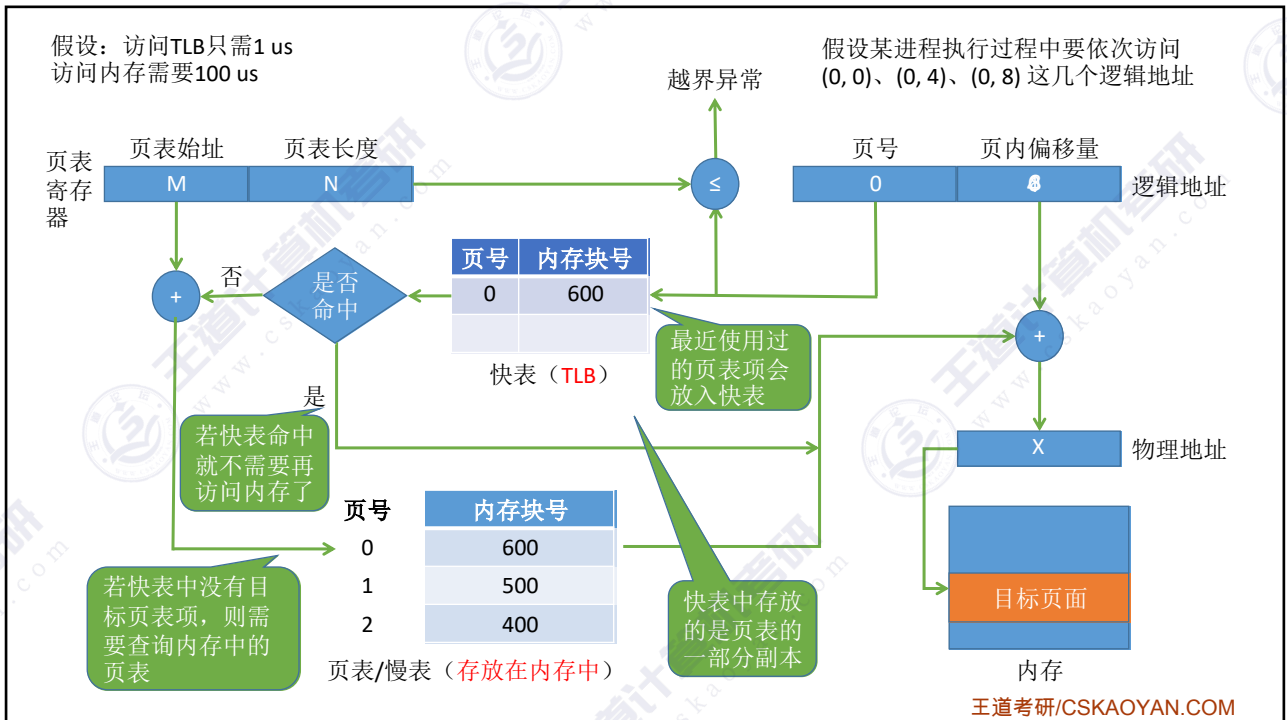
## 套娃警告：虚拟存储系统



8

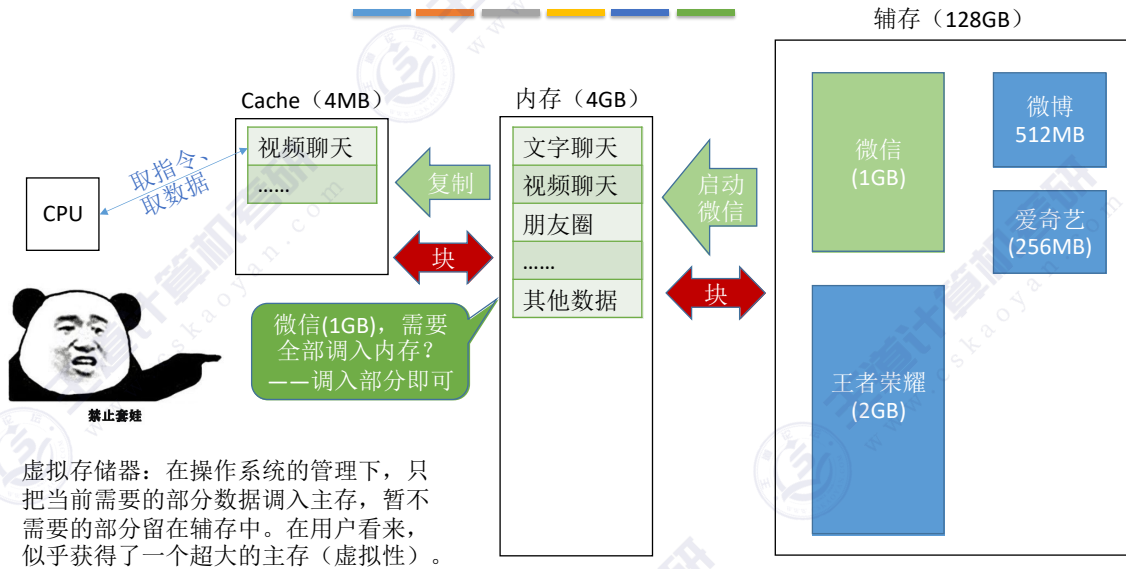


9



10

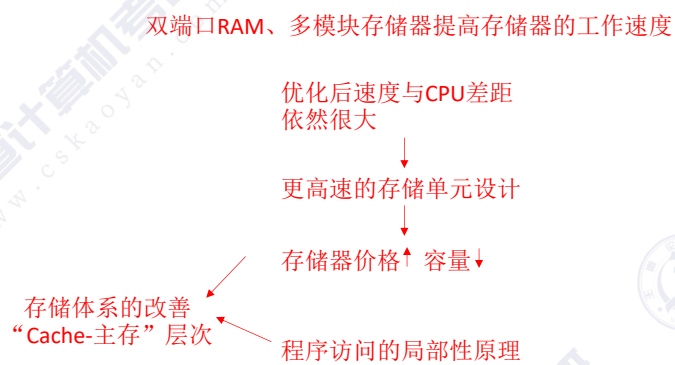
## 套娃警告：虚拟存储系统



王道考研/CSKAOYAN.COM

11

## 存储系统存在的问题

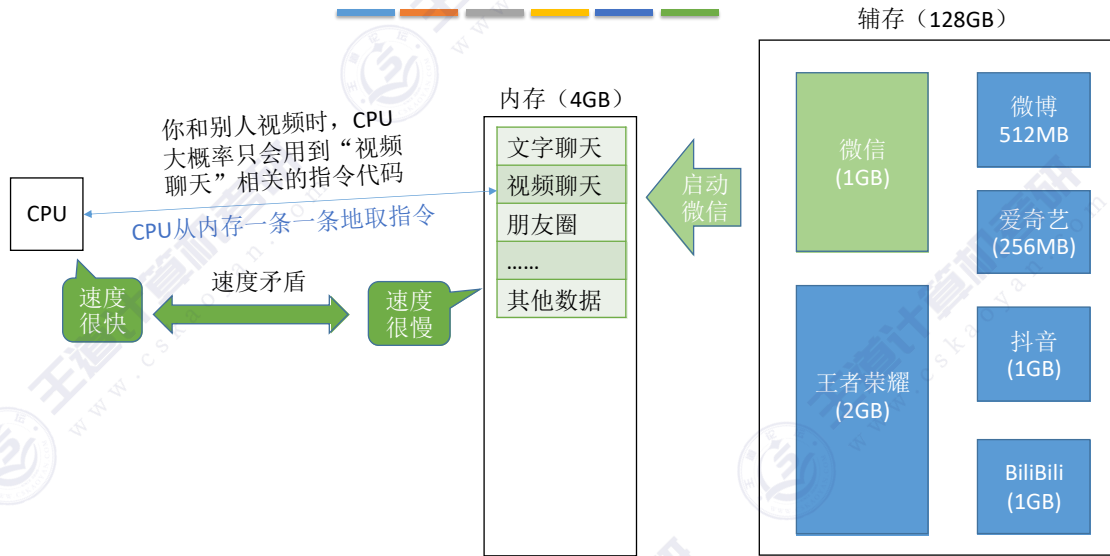


王道考研/CSKAOYAN.COM

12



## Cache的工作原理



王道考研/CSKAOYAN.COM

13

## 局部性原理

程序A:

```

1  int sumarrayrows(int a[M][N])
2  {
3      int i, j, sum = 0;
4      for (i = 0; i < M; i++)
5          for (j = 0; j < N; j++)
6              sum += a[i][j];
7      return sum;
8  }

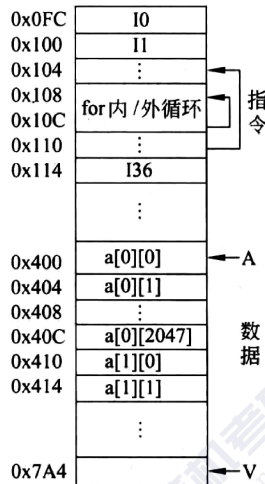
```

程序B:

```

1  int sumarraycols(int a[M][N])
2  {
3      int i, j, sum = 0;
4      for (j = 0; j < N; j++)
5          for (i = 0; i < M; i++)
6              sum += a[i][j];
7      return sum;
8  }

```



Eg: 数组元素、顺序执行的指令代码

**空间局部性:** 在最近的未来要用到的信息(指令和数据), 很可能与现在正在使用的信息在存储空间上是邻近的

**时间局部性:** 在最近的未来要用到的信息, 很可能是现在正在使用的信息

Eg: 循环结构的指令代码

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

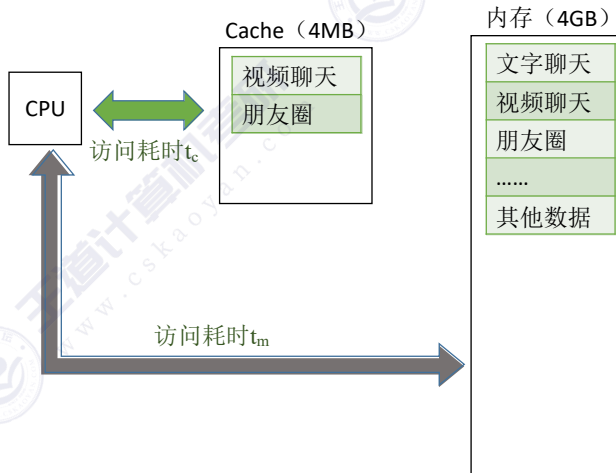
程序B按“列优先”访问二维数组, 空间局部性更差

指令和数据在内存中的存储

王道考研/CSKAOYAN.COM

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

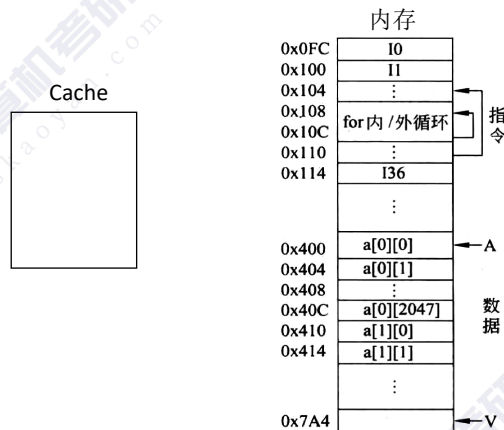
16



### 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



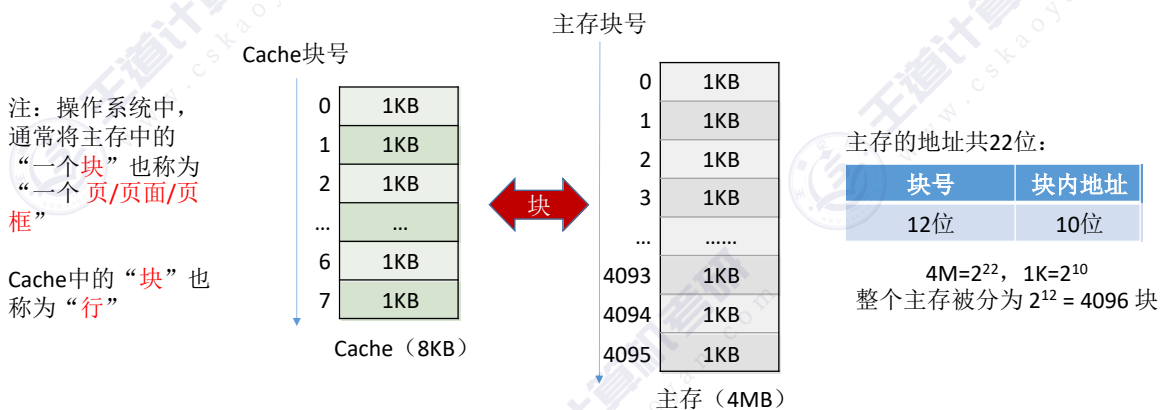
王道考研/CSKAOYAN.COM

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)

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

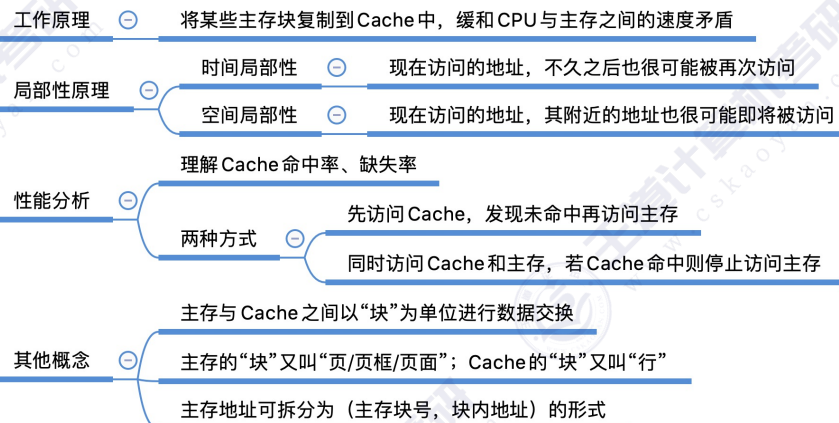
$4M=2^{22}$ ,  $1K=2^{10}$   
整个主存被分为  $2^{12} = 4096$  块

- 如何区分 Cache 与 主存 的数据块对应关系？ ——Cache和主存的映射方式
- Cache 很小，主存很大。如果Cache满了怎么办？ ——替换算法
- 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写策略

王道考研/CSKAOYAN.COM

19

## 知识回顾

高速缓冲存储器  
Cache

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和主存的映射方式
- Cache 很小，主存很大。如果Cache满了怎么办？ ——替换算法
- 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写策略

王道考研/CSKAOYAN.COM

20

## 你还可以在这里找到我们

快速获取第一手计算机考研信息&资料



购买2024考研全程班/领学班/定向班  
可扫码加微信咨询



微博: @王道计算机考研教育



B站: @王道计算机教育



小红书: @王道计算机考研



知乎: @王道计算机考研



抖音: @王道计算机考研



淘宝: @王道论坛书店