

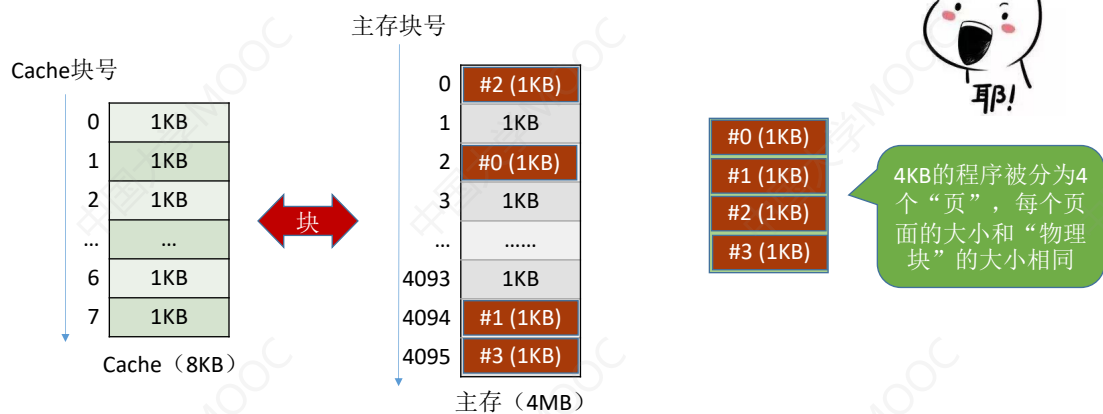
本节内容

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)

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

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

程序员视角:

整个程序共 $4KB=2^{12}B$, 地址范围: **000000000000~111111111111**

变量 x 的逻辑地址: **001000000011**

变量 y 的逻辑地址: **110000001010**

逻辑页号

页内地址

若干位

10位

逻辑地址 (虚地址): 程序员视角看到的地址

物理地址 (实地址): 实际在主存中的地址

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

变量 y 的物理地址: **1111111111100000001010**

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 主存块号

#0	2
#1	4094
#2	0
#3	4095

页表项

页表数据
在主存里

页表

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

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)

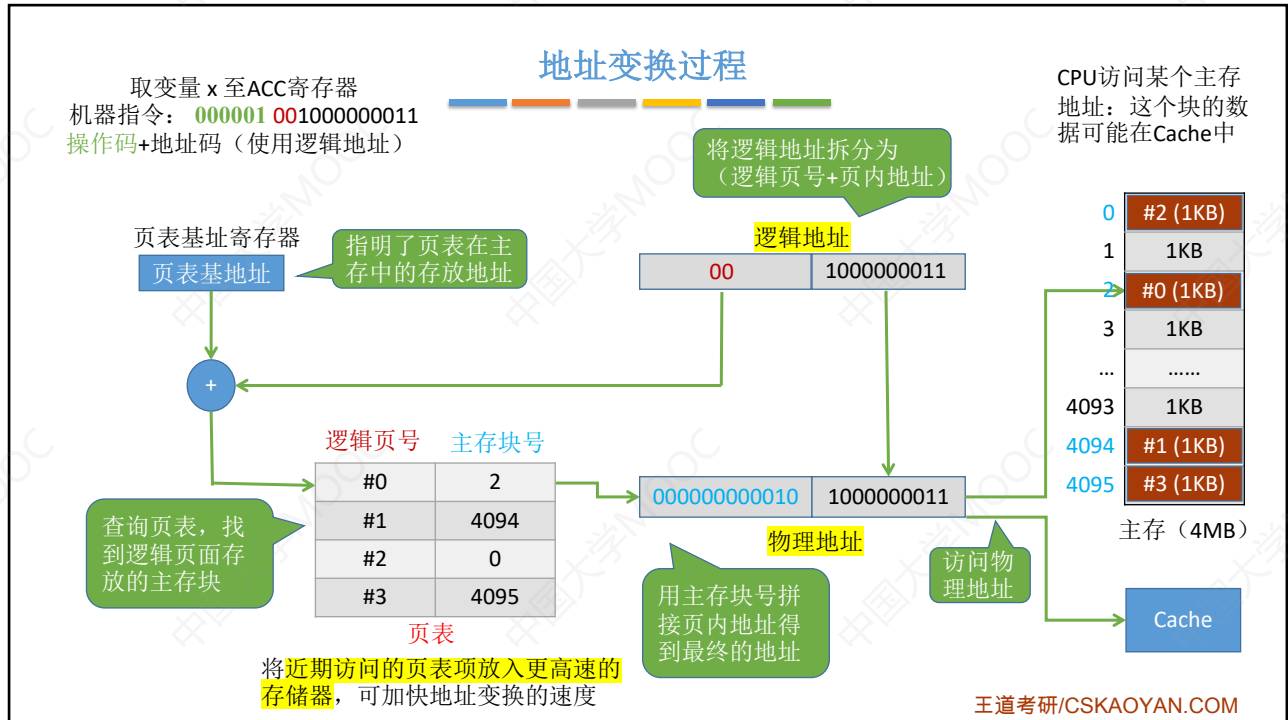
变量 x 的逻辑地址: **001000000011**

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

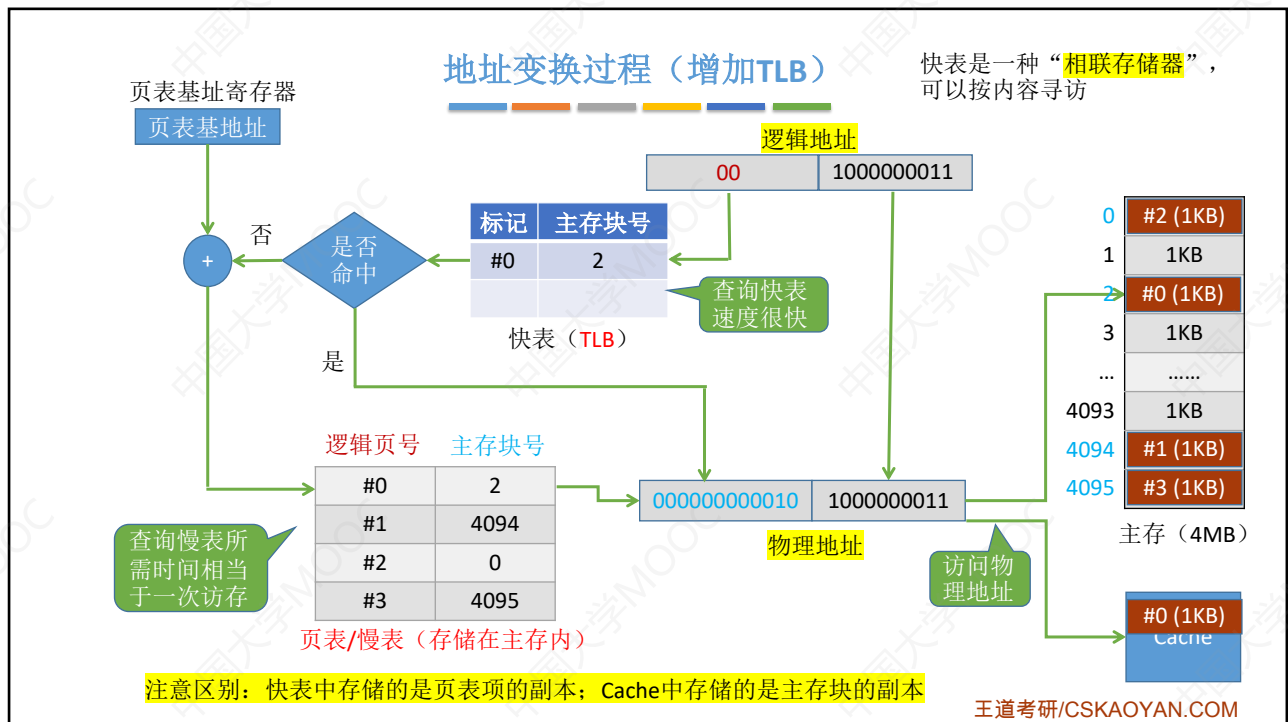
CPU 执行的机器指令中, 使用的是“逻辑地址”, 因此需要通“页表”将逻辑地址转为物理地址。
 页表的作用: 记录了每个逻辑页面存放在哪个主存块中

王道考研/CSKAOYAN.COM

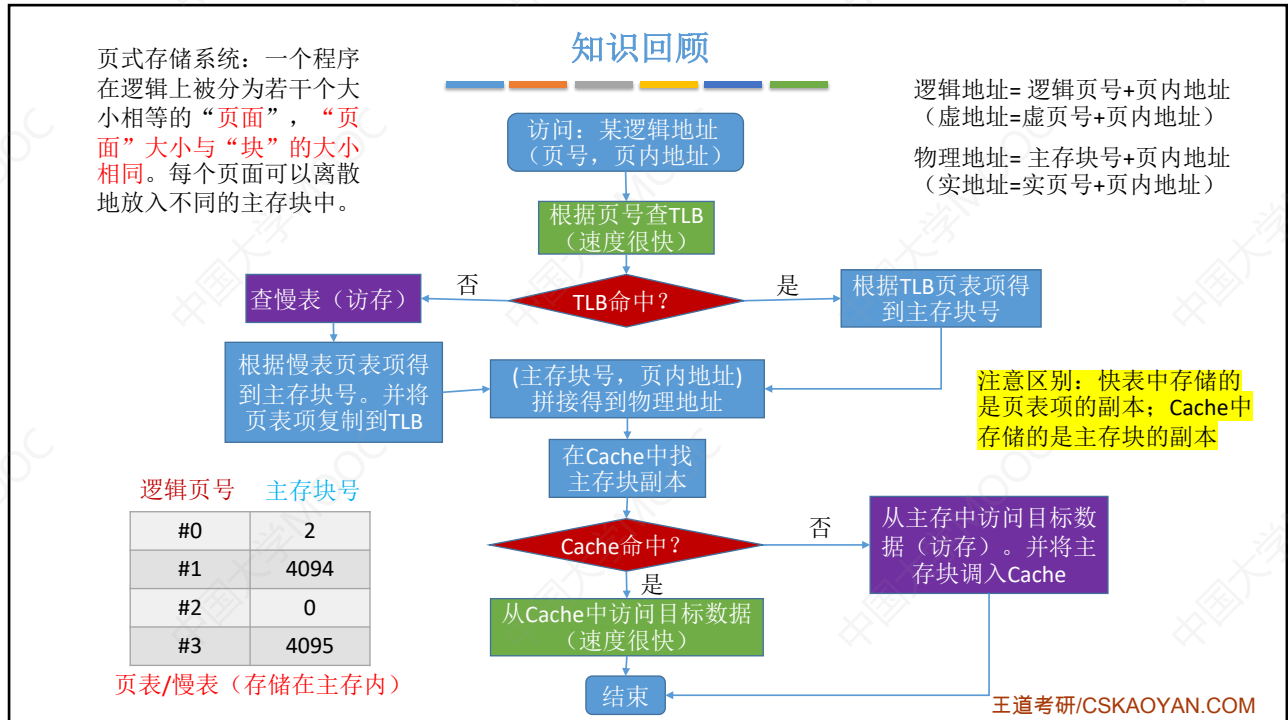
4



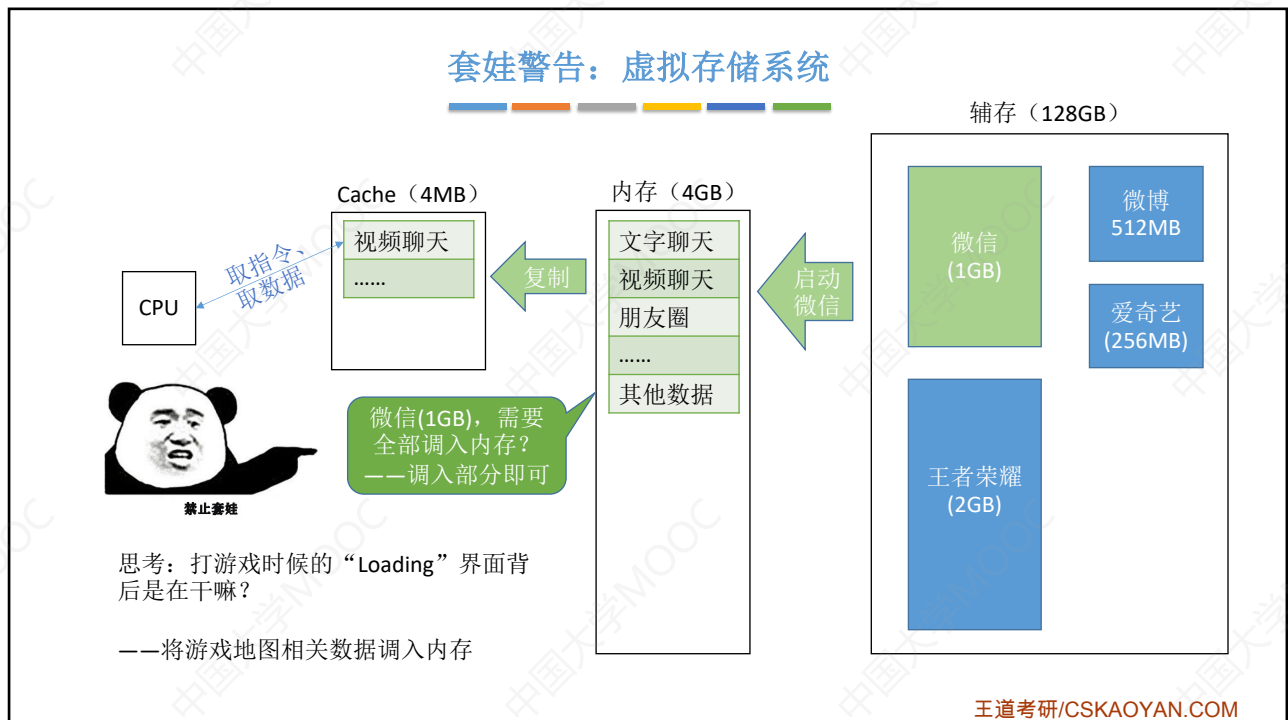
5



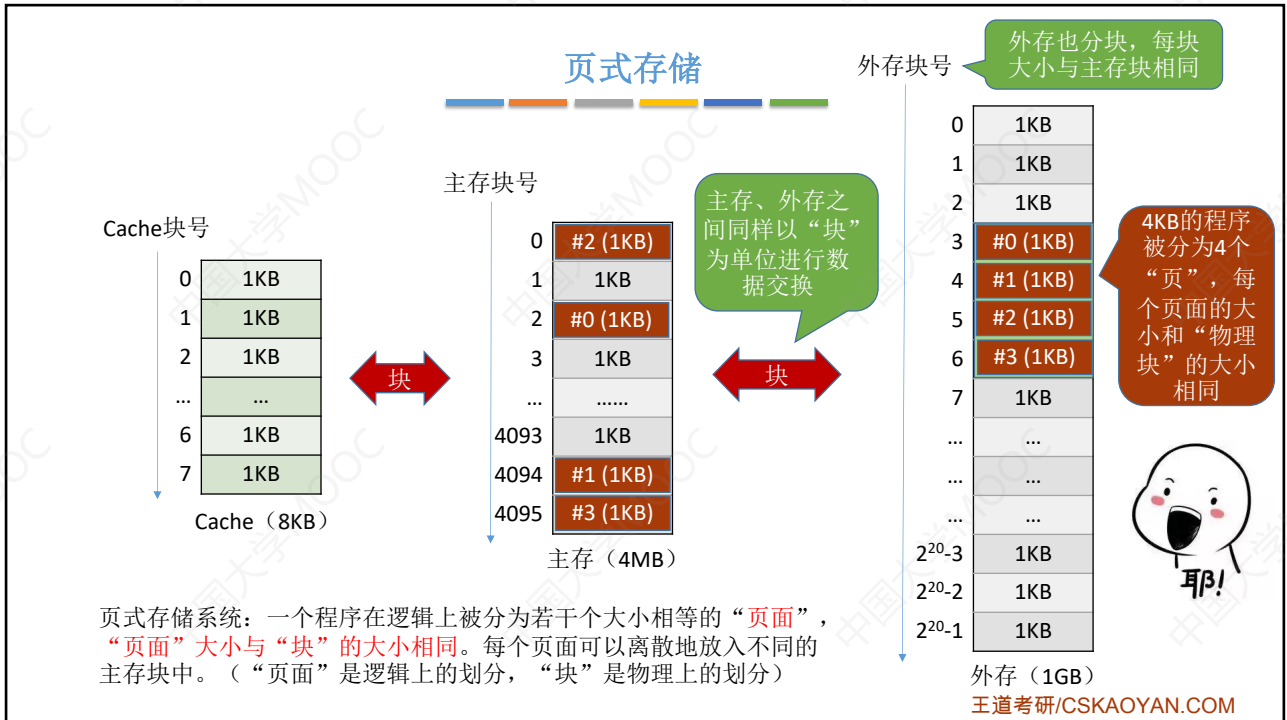
6



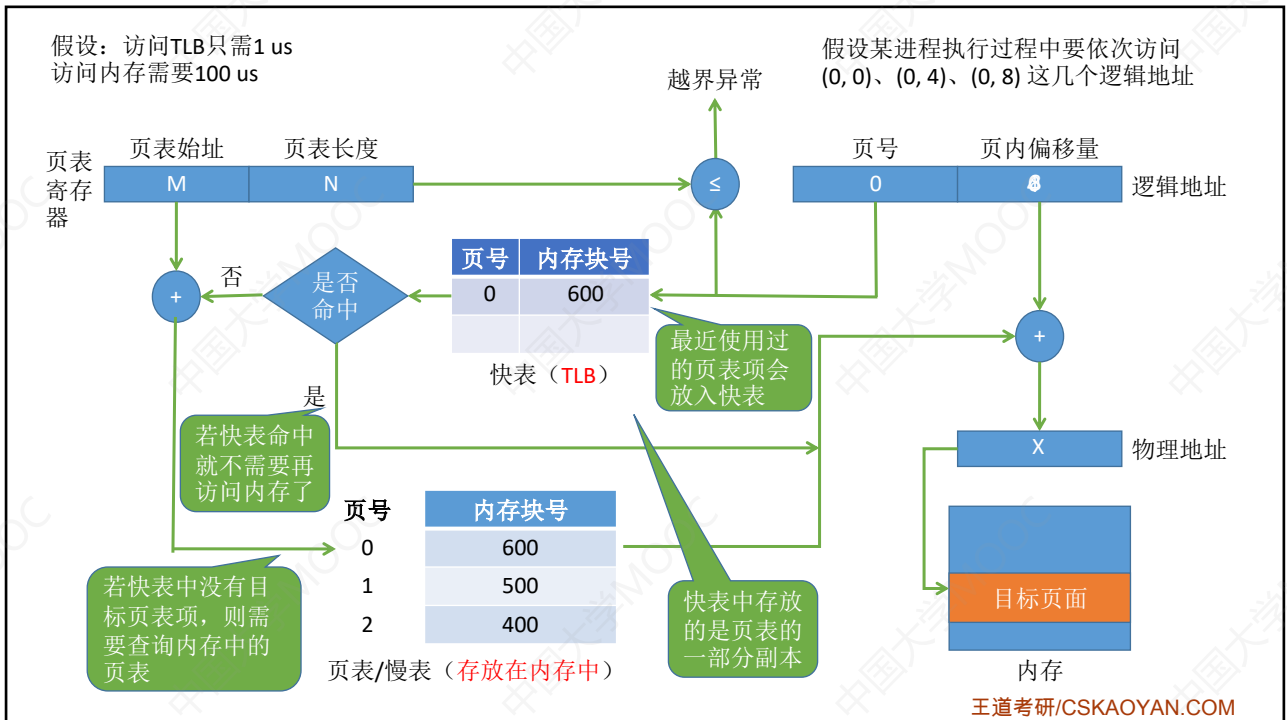
7



8

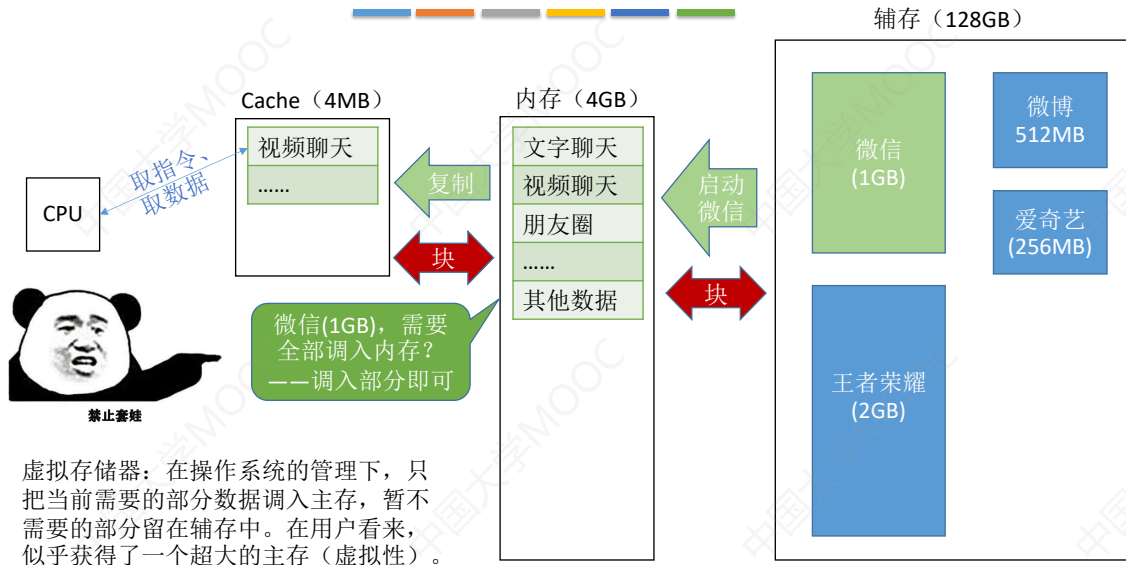


9



10

套娃警告：虚拟存储系统

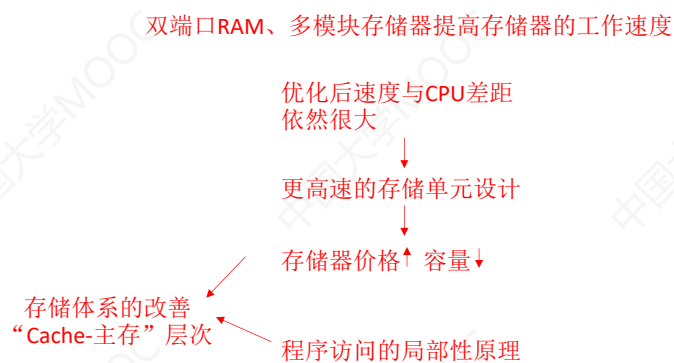


虚拟存储器：在操作系统的管理下，只把当前需要的部分数据调入主存，暂不需要的部分留在辅存中。在用户看来，似乎获得了一个超大的主存（虚拟性）。

王道考研/CSKAOYAN.COM

11

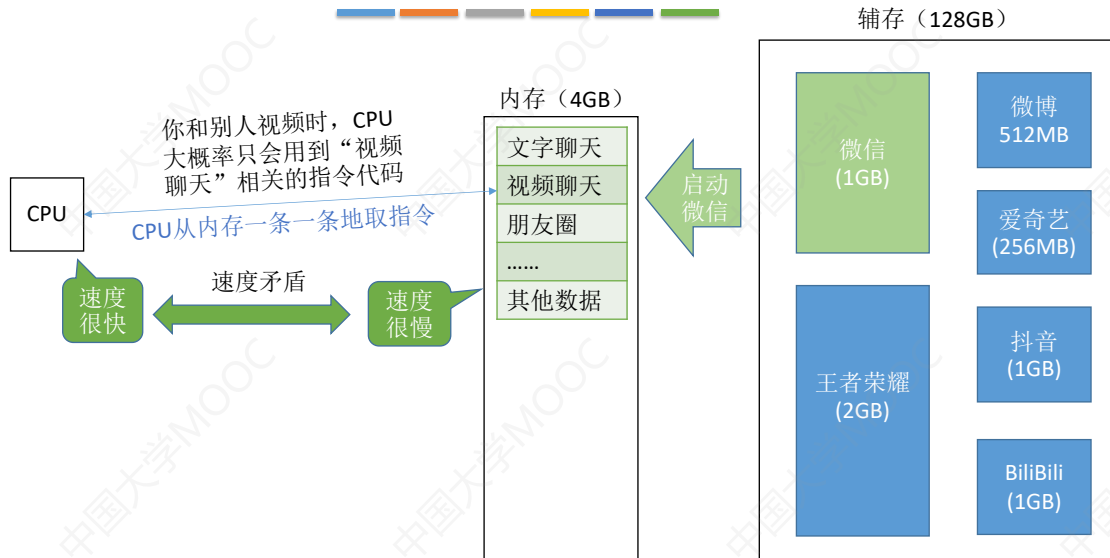
存储系统存在的问题



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  }

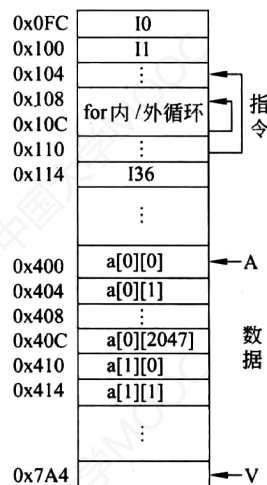
```

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

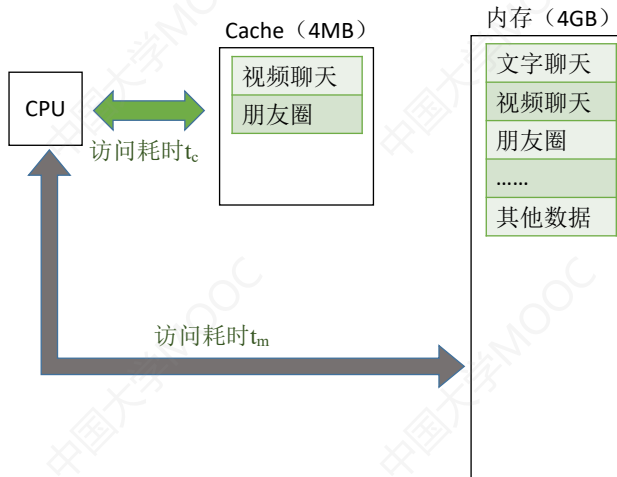
程序B按“列优先”访问二维数组, 空间局部性更差

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

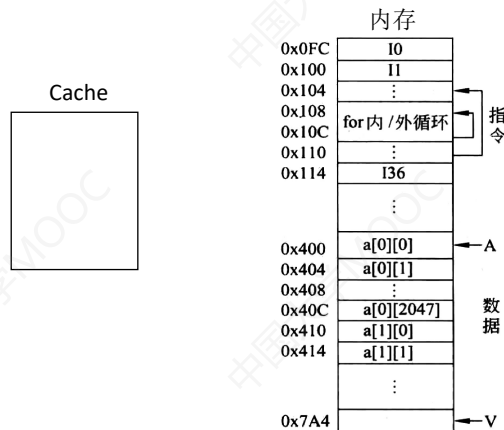
王道考研/CSKAOYAN.COM

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



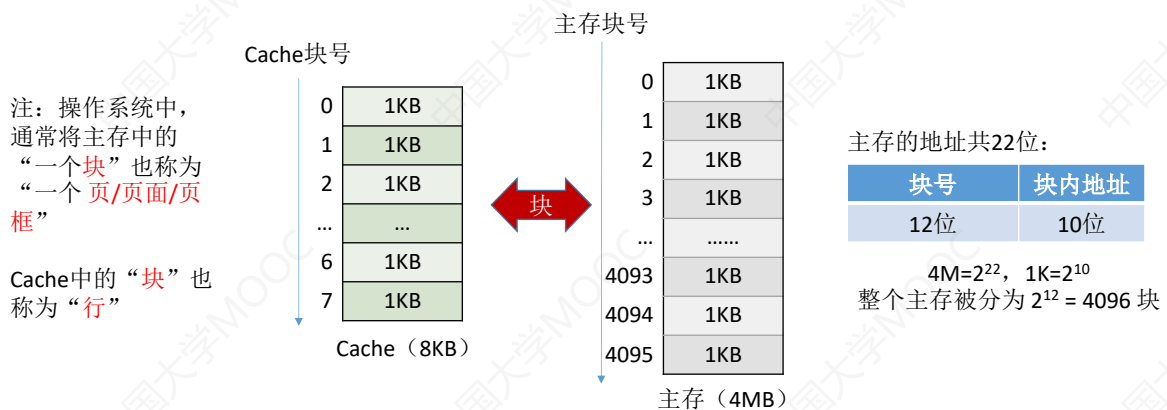
王道考研/CSKAOYAN.COM

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”？

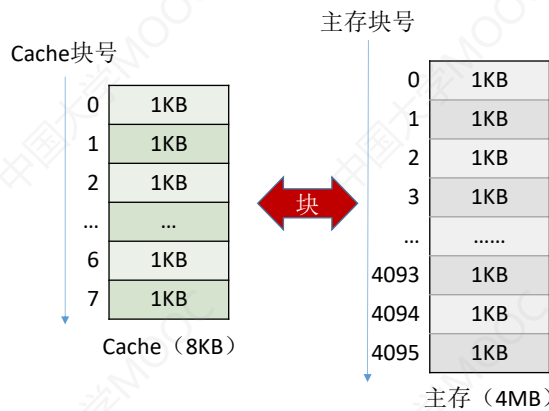
将主存的存储空间“分块”，如：每 1KB 为一块。主存与Cache之间以“块”为单位进行数据交换



王道考研/CSKAOYAN.COM

18

有待解决的问题



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

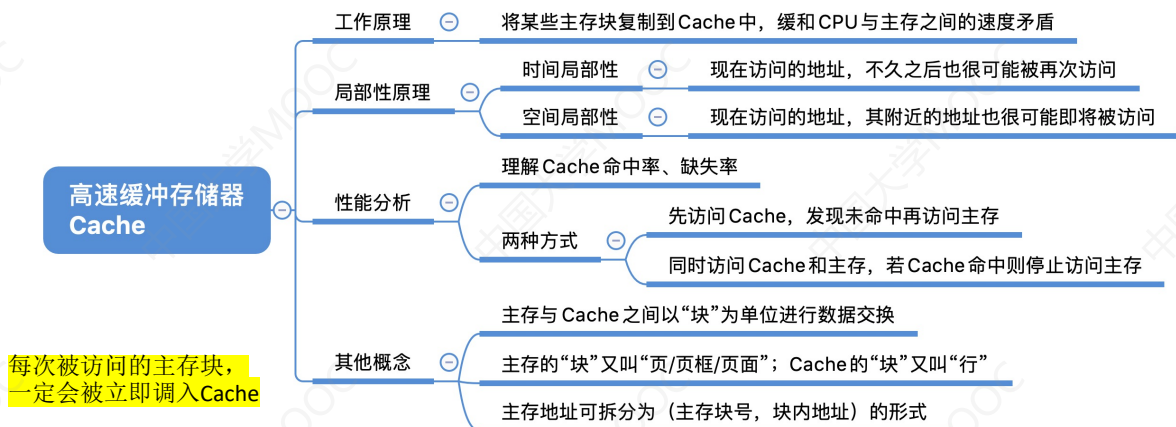
$4M=2^{22}$, $1K=2^{10}$
整个主存被分为 $2^{12} = 4096$ 块

- 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



- 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



@王道论坛



@王道计算机考研备考



@王道咸鱼老师-计算机考研

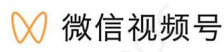
@王道楼楼老师-计算机考研



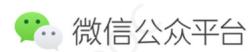
@王道计算机考研



@王道计算机考研



@王道计算机考研



@王道在线