第三章

存储管理

方　钰

主 要 内 容

3.1 存储管理的主要任务

3.2 连续分配方式

3.3 页式存储管理

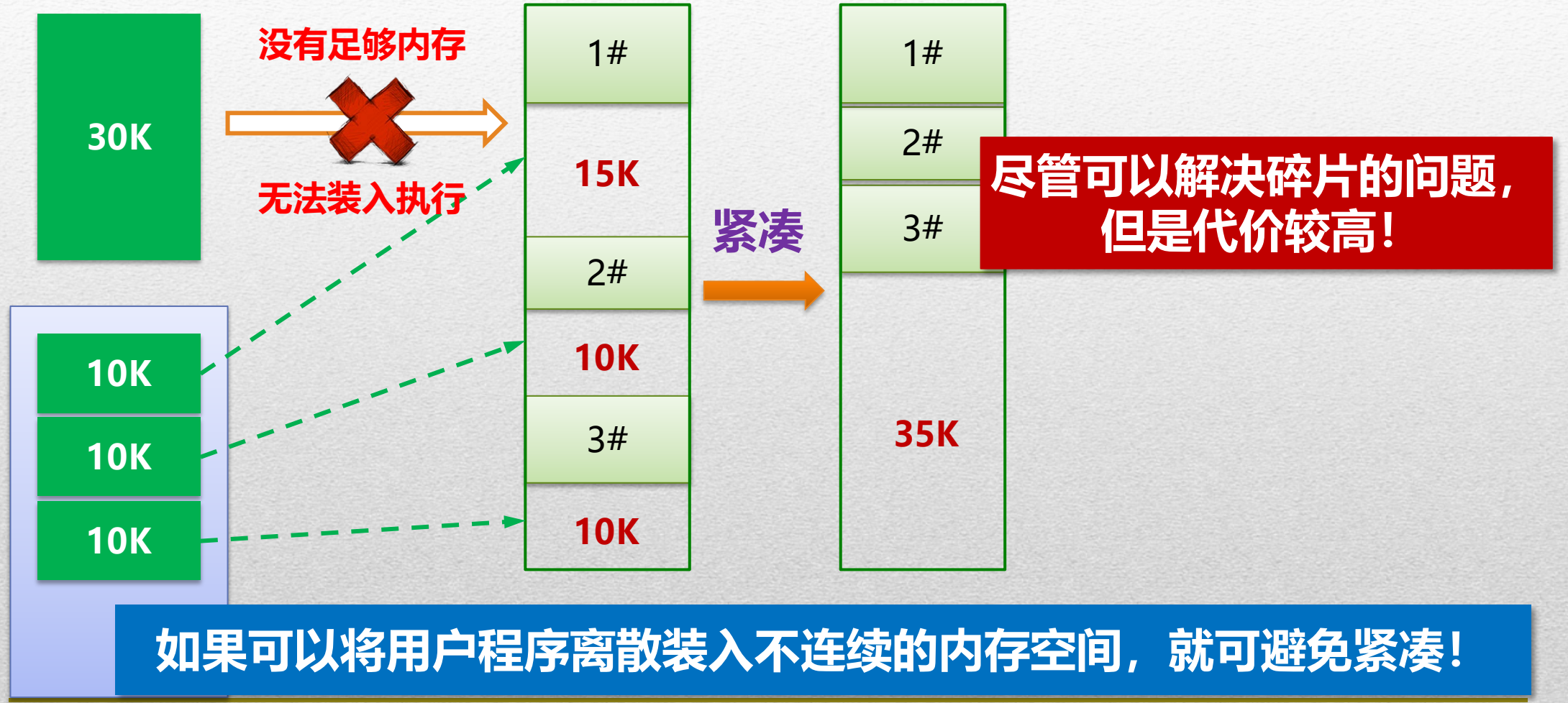
3.4 段式与段页式存储管理**

3.5 UNIX 存储管理

连续分配方式

一个用户程序占用一个连续的内存空间

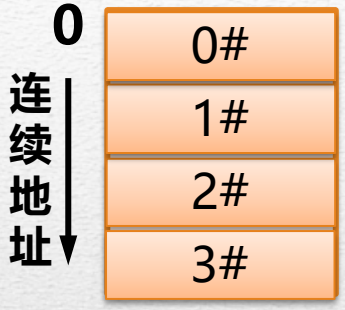
实现简单，可用于单道或多道程序，但……



如果可以将用户程序离散装入不连续的内存空间，就可避免紧凑！

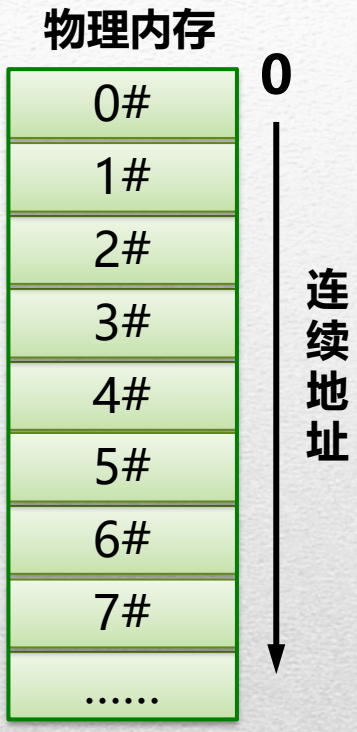
分页存储管理

程序逻辑地址空间



把每个程序的连续逻辑地址空间划分成若干大小相等的页

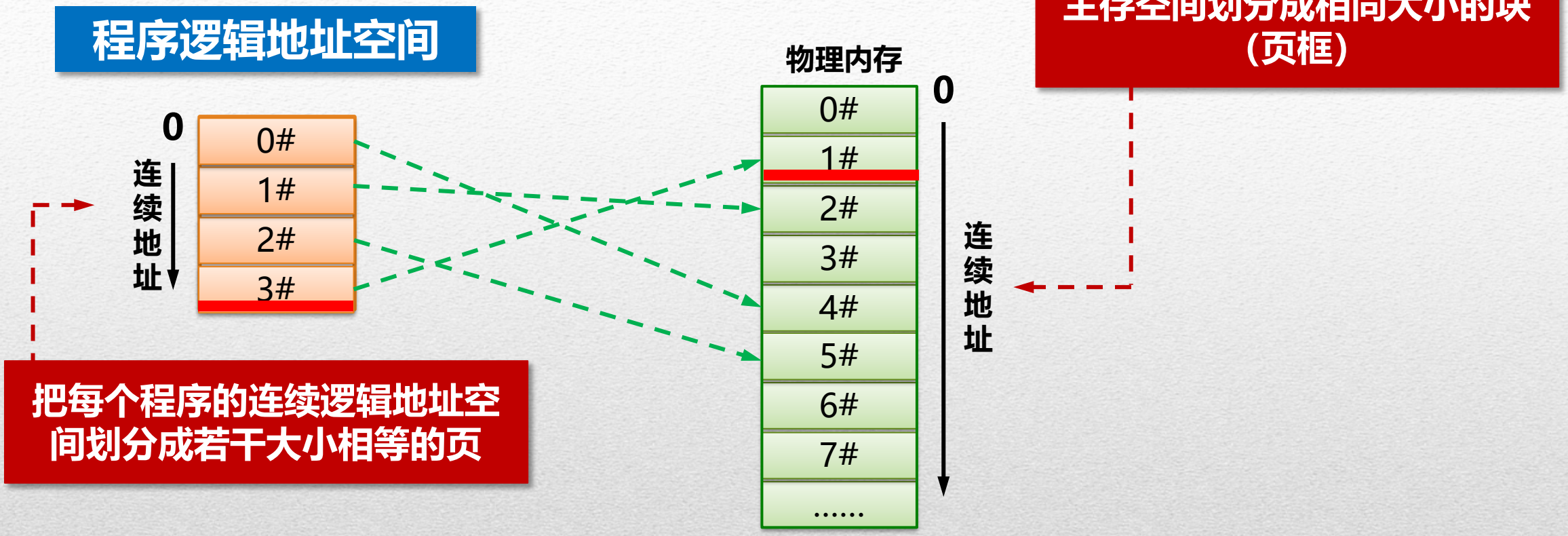
主存空间划分成相同大小的块（页框）



分配存储空间以页为单位
每页分别装入一个页框

不需要页面连续

分页存储管理



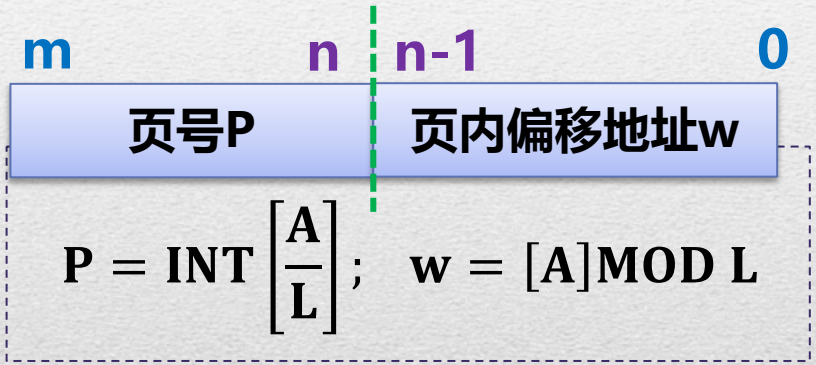
分配存储空间以页为单位
每页分别装入一个页框

不需要页面连续

分页存储管理

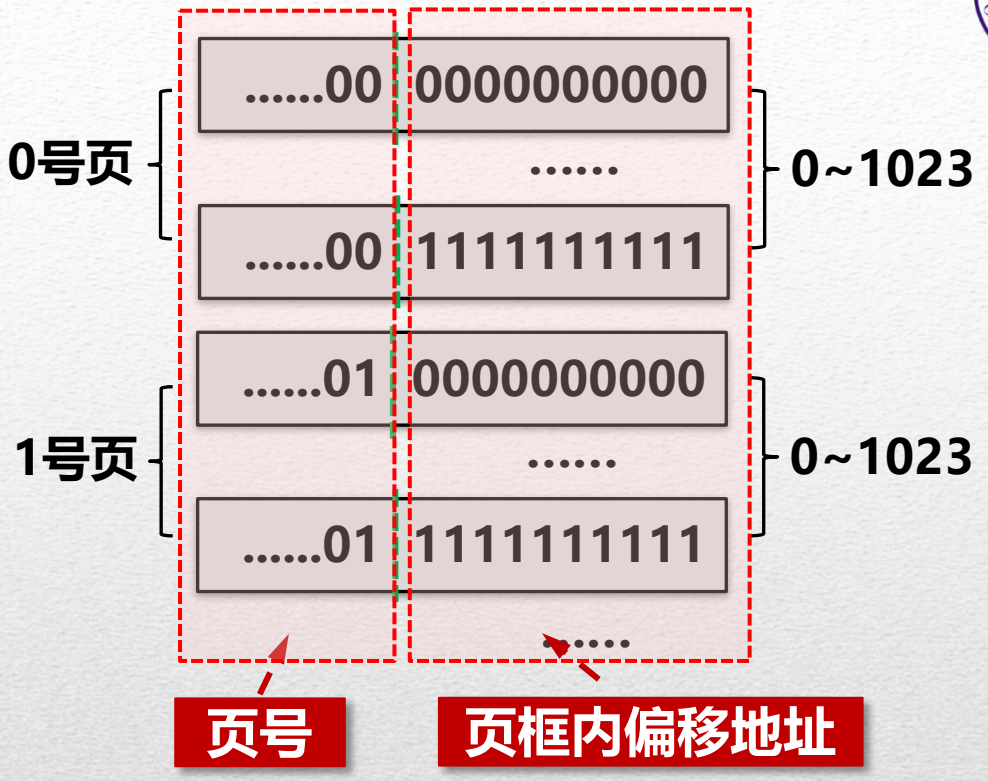
程序逻辑地址空间

地址A，页长：$L=2^n$
程序地址分成两个部分：

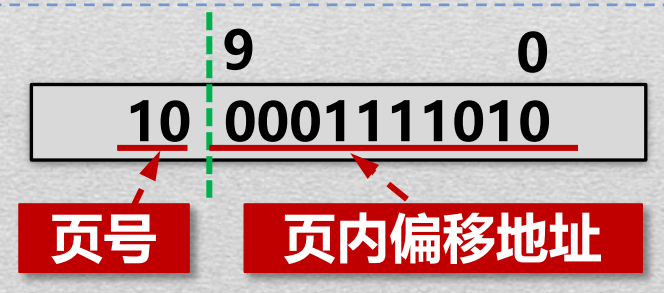


从0开始顺序编址的程序地址，称为**线性地址**
（一维的，连续的）

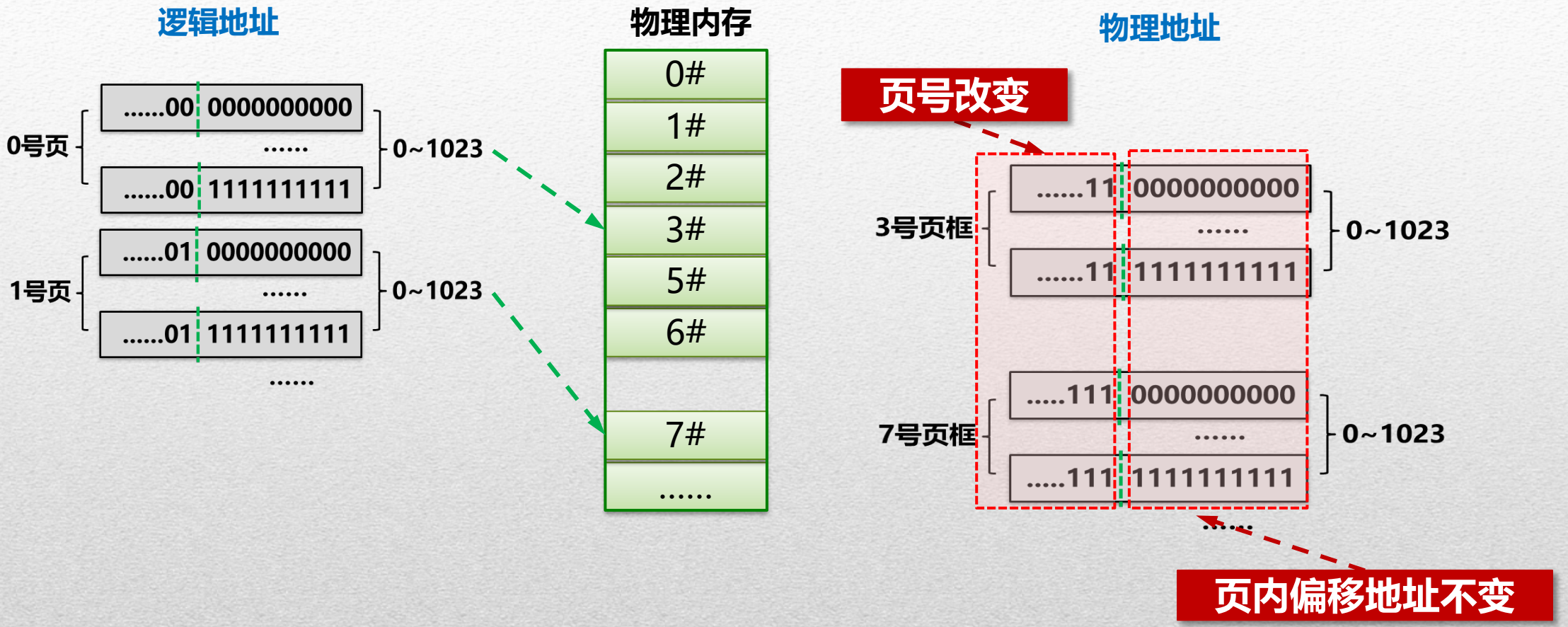
例如，某系统的逻辑页面大小1KB



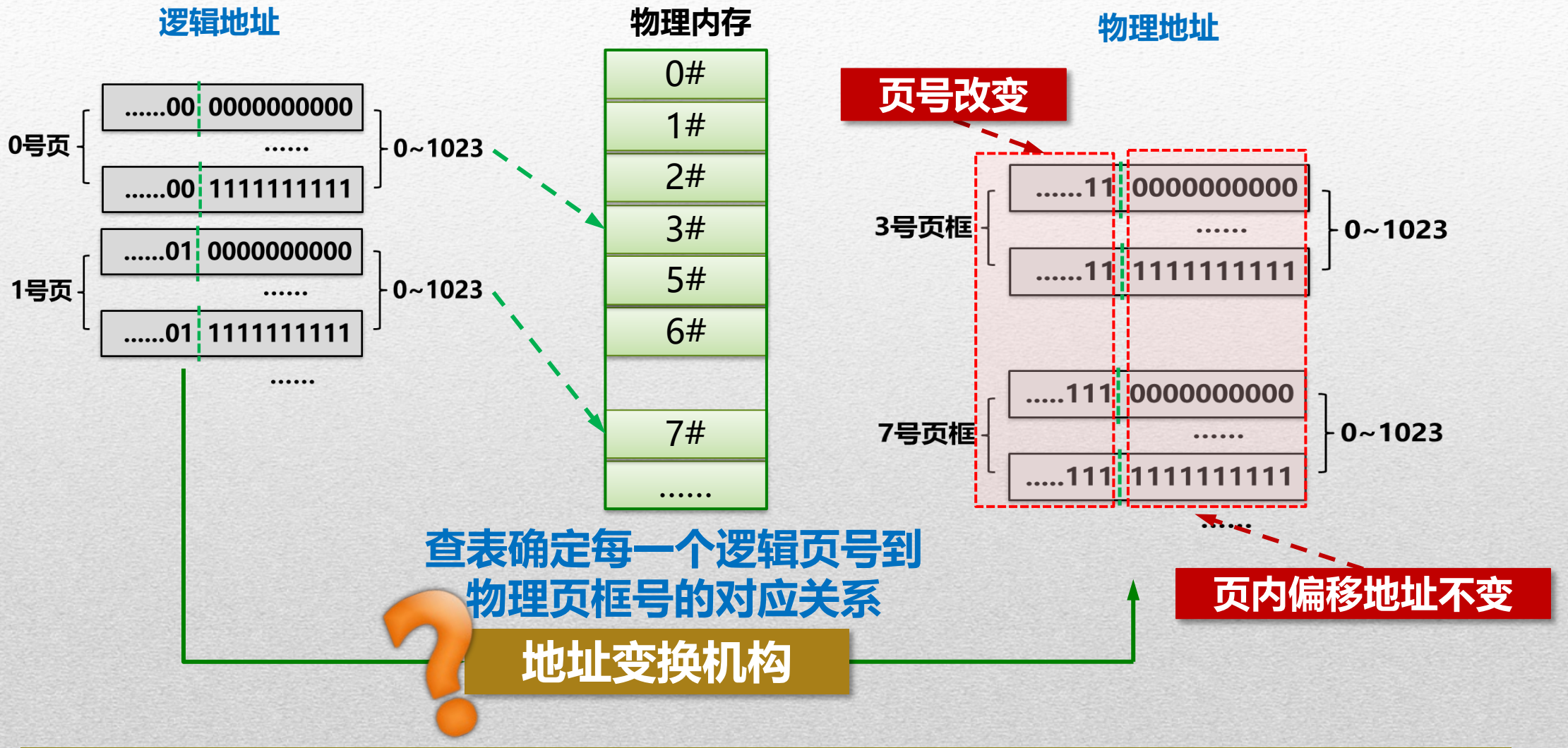
上例中，若A＝2170，则P=2，W=122



分页存储管理

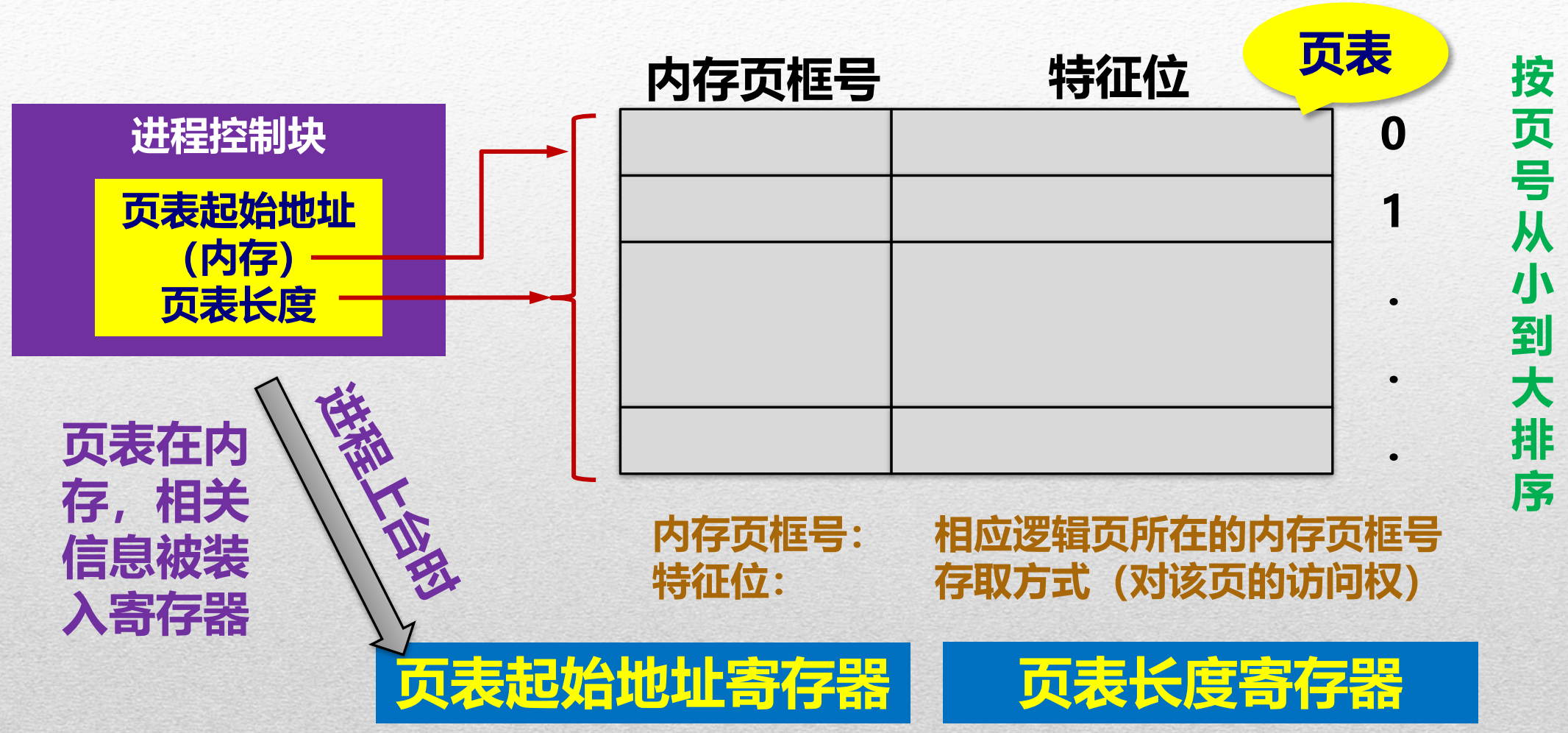

逻辑地址
物理内存
物理地址
0号页
1号页
.....00 0000000000
.....00 1111111111
.....01 0000000000
.....01 1111111111
0~1023
0~1023
0#
1#
2#
3#
5#
6#
7#
页号改变
3号页框
7号页框
.....11 0000000000
.....11 1111111111
.....111 0000000000
.....111 1111111111
0~1023
0~1023
页内偏移地址不变

分页存储管理


逻辑地址
物理内存
物理地址
0号页
1号页
.....00 0000000000
.....00 1111111111
.....01 0000000000
.....01 1111111111
0~1023
0~1023
0#
1#
2#
3#
5#
6#
7#
页号改变
3号页框
7号页框
.....11 0000000000
.....11 1111111111
.....111 0000000000
.....111 1111111111
0~1023
0~1023
查表确定每一个逻辑页号到
物理页框号的对应关系
地址变换机构
页内偏移地址不变

分页存储管理

利用页表实现页号到物理页框号的映射



分页存储管理

利用页表实现页号到物理页框号的映射

CPU内的两个寄存器

页表长度　　页表始址

页表

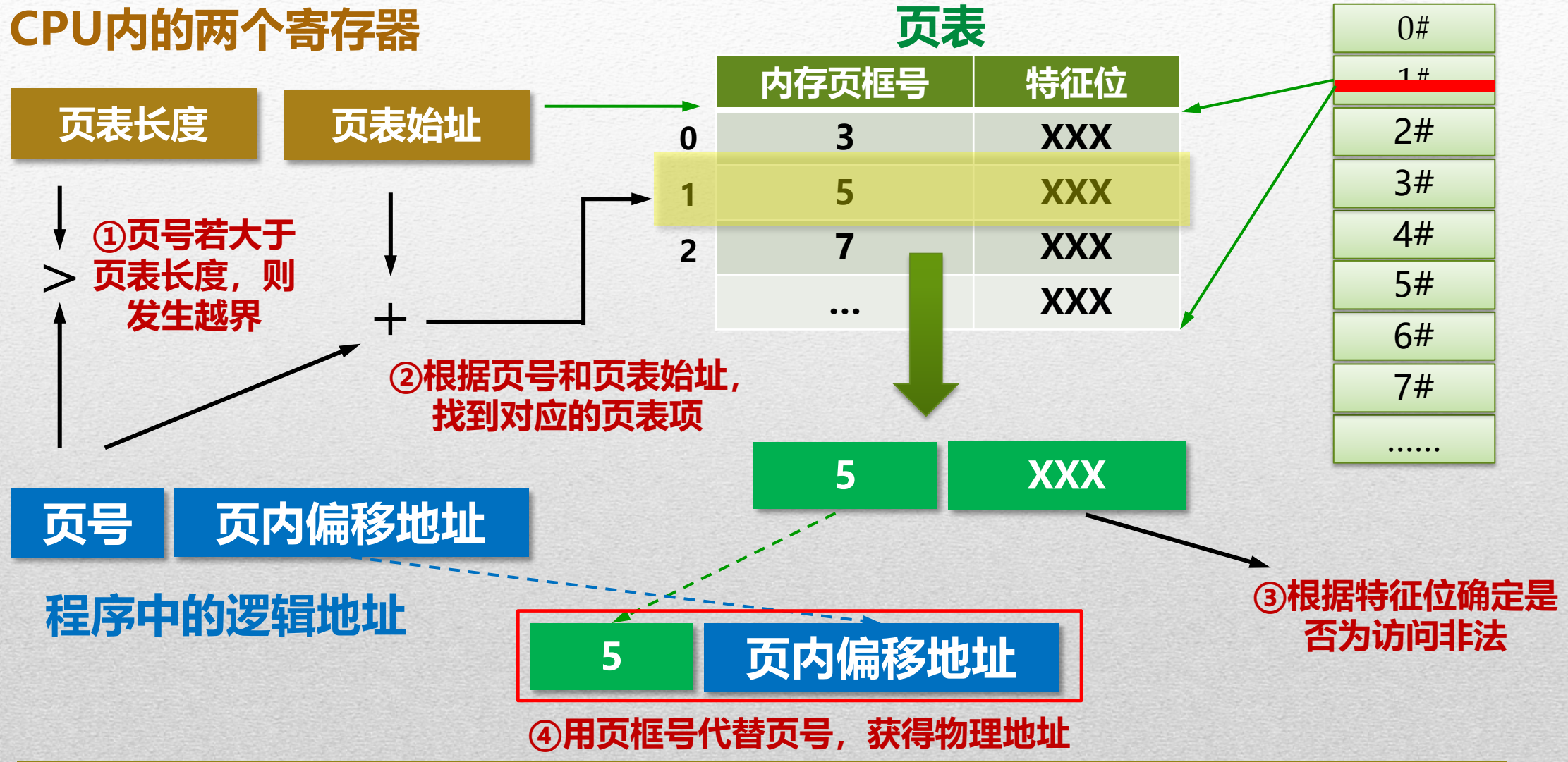
	内存页框号	特征位
0	3	XXX
1	5	XXX
2	7	XXX
	…	XXX

页号　　页内偏移地址

程序中的逻辑地址

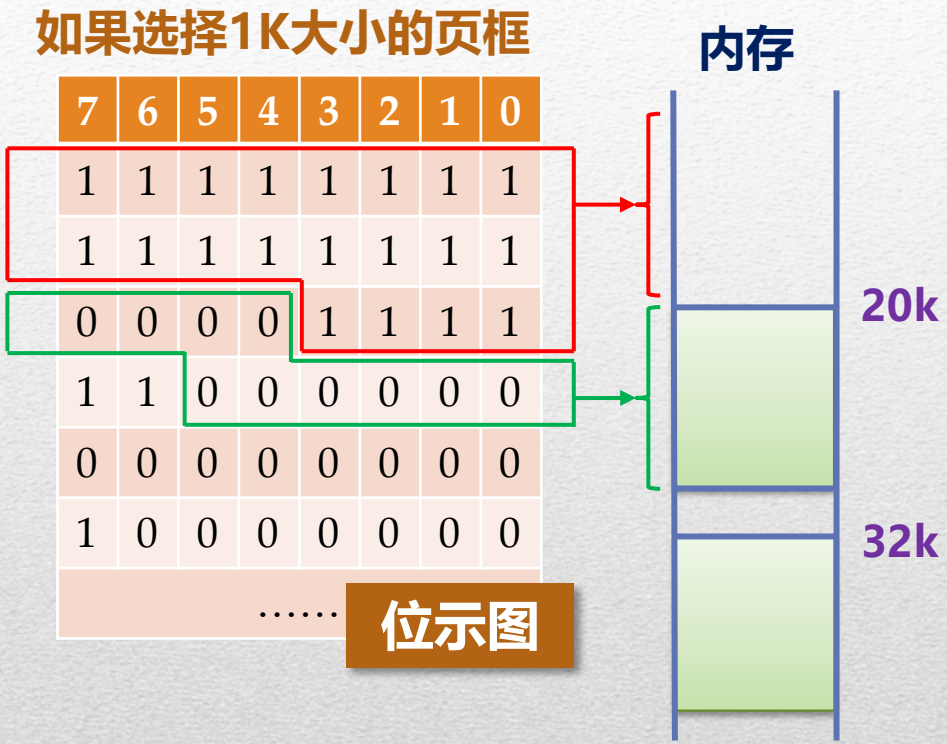
分页存储管理

利用页表实现页号到物理页框号的映射

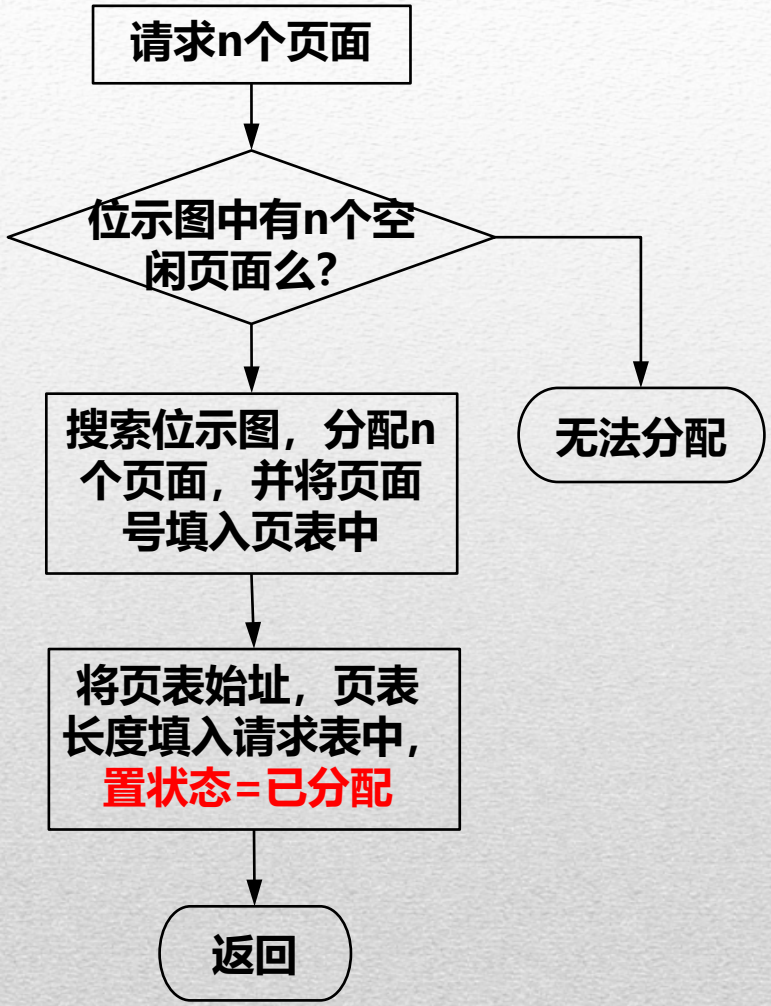


分页存储管理

空闲页面的管理



采用位示图方式管理



分配算法

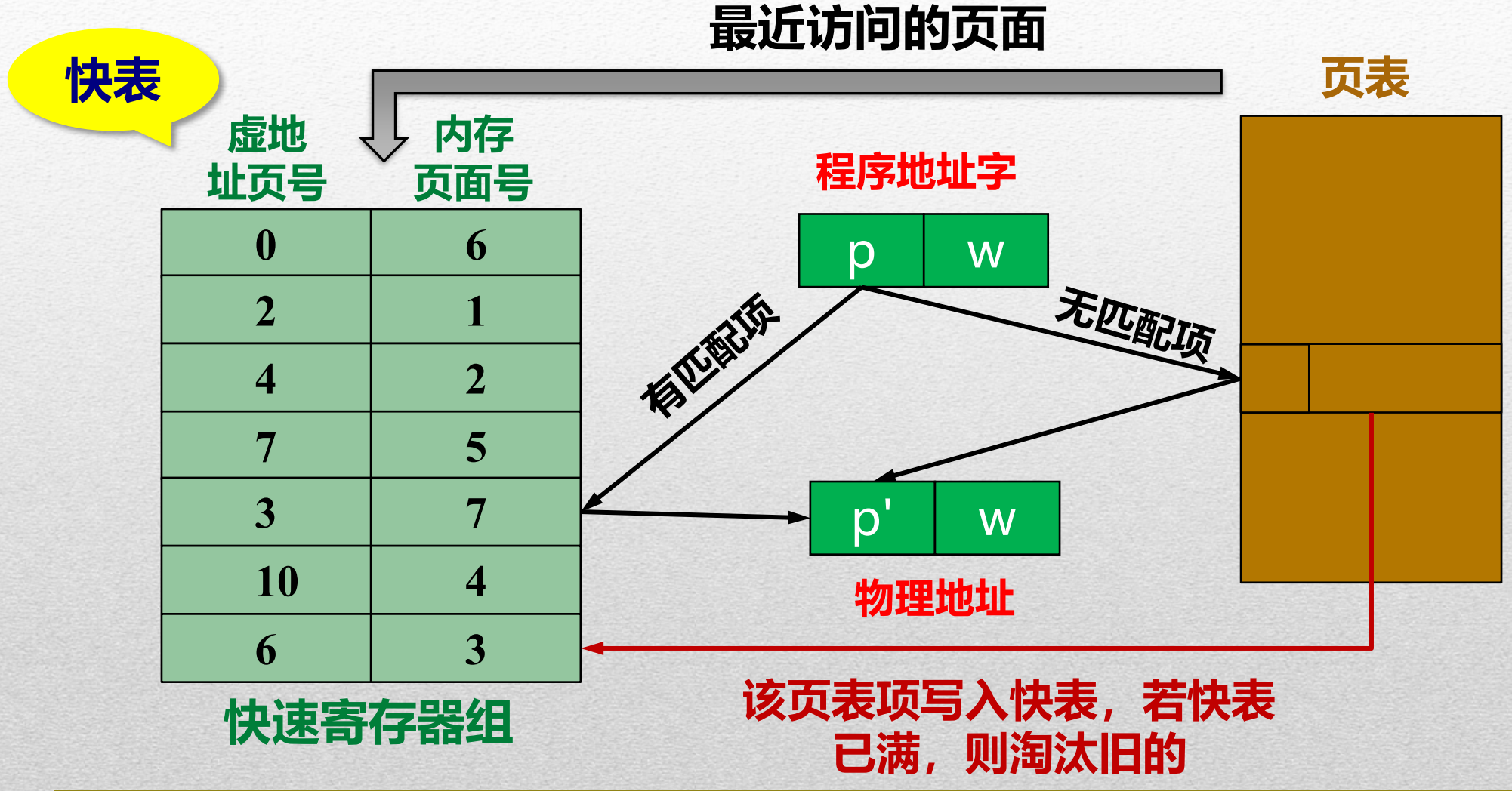
分页存储管理有缺点么?

访问内存页表速度慢?

—— 快表

分页存储管理

页表访问慢怎么办？？？



分页存储管理

分页存储管理有缺点么?

访问内存页表速度慢?

—— 快表

物理内存不够大?

——虚拟内存

虚拟存储器

利用请求分页实现虚拟存储器

具有**请求调页功能**和**页置换功能**，能从逻辑上对内存容量加以扩充的一种存储器系统。**其逻辑容量由内存容量和外存容量之和决定，速度接近于内存速度，成本接近于外存**。

程序运行前，不必全部装入内存，只需将当前要运行的少数页面先装入内存便可运行。

程序运行时，如果要访问的页已在内存，则继续执行。否则，利用请求调页功能，将它们调入内存。

若此时内存已满，则利用页置换功能，将内存中暂时不用的页调出，腾出空间后，再将需要的页调入。

利用请求分页实现虚拟存储器

具有请求调页功能和页置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和决定，速度接近于内存速度，成本接近于外存。

建立在基本分页存储管理方式基础之上：

- 扩展的页表
- 地址变换机构
- 缺页中断机构
- 页面置换算法

利用请求分页实现虚拟存储器

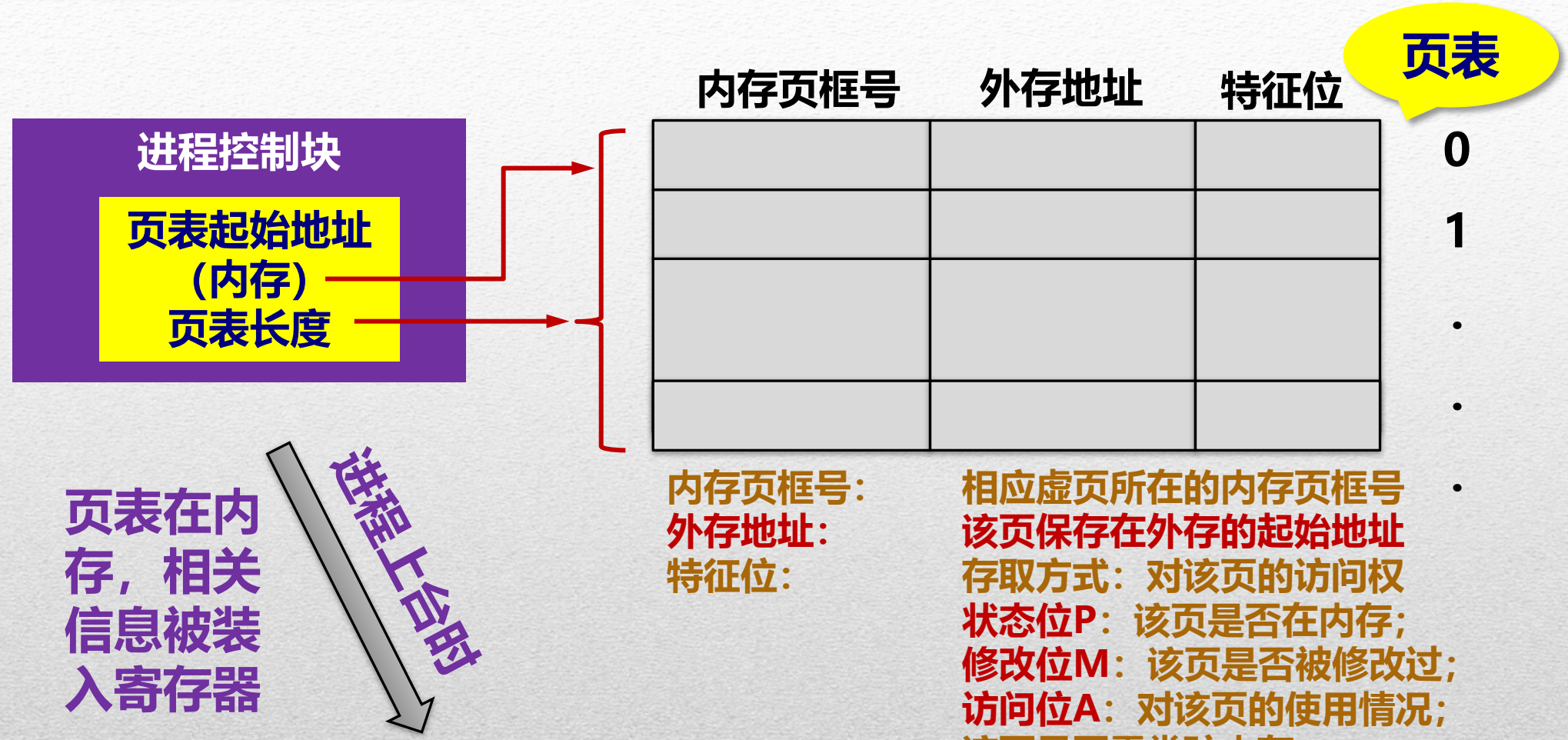
具有请求调页功能和页置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和决定，速度接近于内存速度，成本接近于外存。

建立在基本分页存储管理方式基础之上：

- 扩展的页表
- 地址变换机构
- 缺页中断机构
- 页面置换算法

虚拟存储器

利用请求分页实现虚拟存储器



利用请求分页实现虚拟存储器

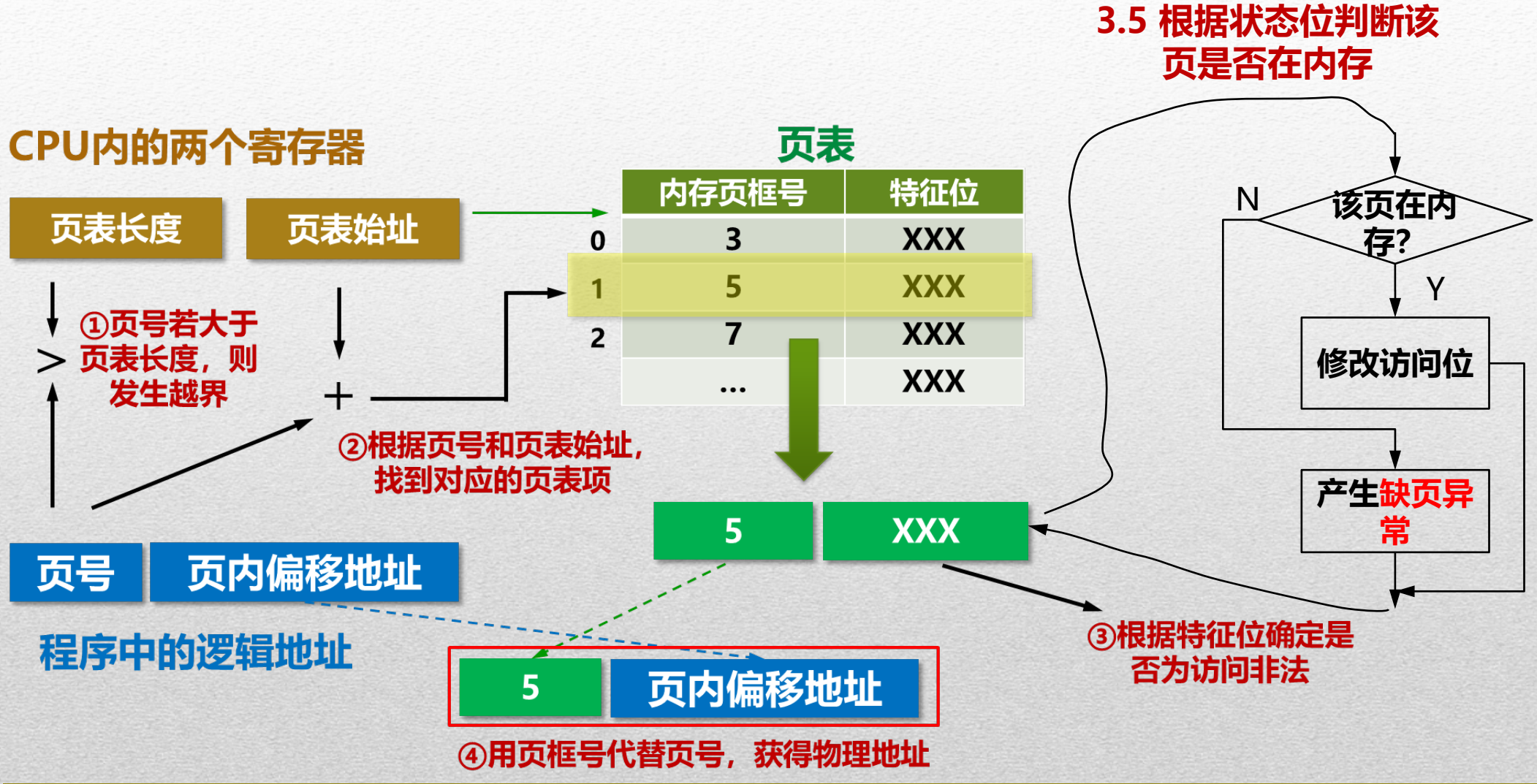
具有请求调页功能和页置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和决定，速度接近于内存速度，成本接近于外存。

建立在基本分页存储管理方式基础之上：

- 扩展的页表
- 地址变换机构
- 缺页中断机构
- 页面置换算法

虚拟存储器

利用请求分页实现虚拟存储器



利用请求分页实现虚拟存储器

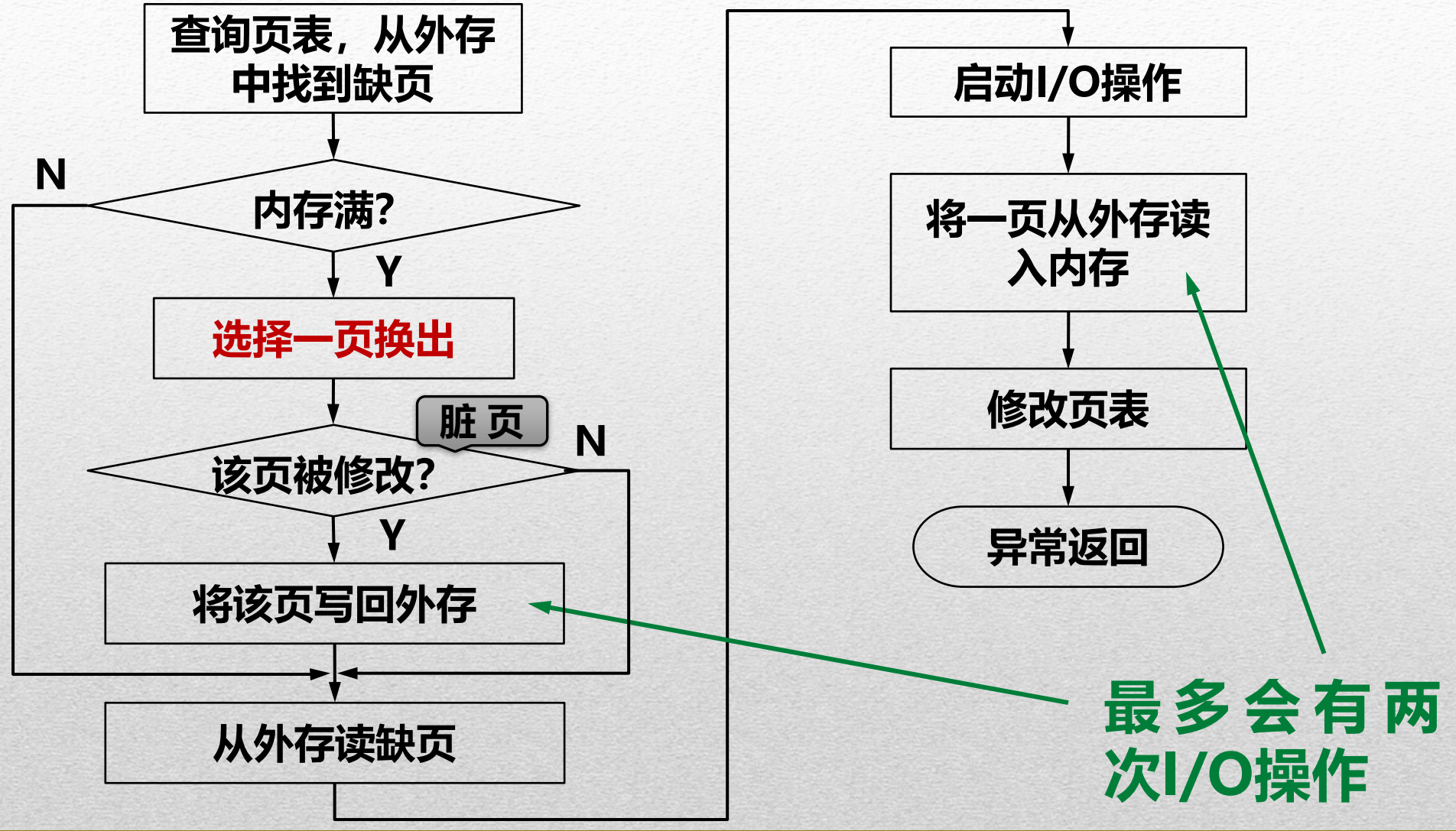
具有请求调页功能和页置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和决定，速度接近于内存速度，成本接近于外存。

建立在基本分页存储管理方式基础之上：

- 扩展的页表
- 地址变换机构
- 缺页异常机构
- 页面置换算法

虚拟存储器

利用请求分页实现虚拟存储器



利用请求分页实现虚拟存储器

具有请求调页功能和页置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和决定，速度接近于内存速度，成本接近于外存。

建立在基本分页存储管理方式基础之上：

- 扩展的页表
- 地址变换机构
- 缺页异常机构
- 页面置换算法

虚拟存储器

利用请求分页实现虚拟存储器

最佳置换算法——理论上的最佳算法

其所选择的被淘汰页面，将是以后永不使用的，或许在未来最长时间内不再被访问的页面。

假定系统为某进程分配了3个页框，并考虑以下的页号引用串：

7，0，1，2，0，3，0，4，2，3，0，3，2，1，2，0，1，7，0，1

7	7	7	2	2	2	2	2	7
	0	0	0	0	4	0	0	0
		1	1	3	3	3	1	1

共发生9次缺页异常，6次页面置换

最低的缺页率，但无法预知哪一页未来最长时间内不再访问

虚拟存储器

利用请求分页实现虚拟存储器

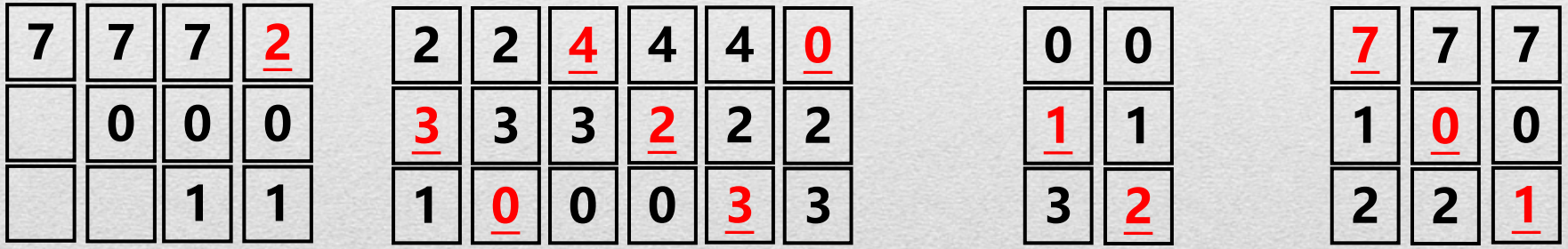
先进先出（FIFO）置换算法

其总是淘汰最先进入内存的页面。

简单，但不能保证经常访问的页面不被淘汰。

假定系统为某进程分配了3个页框，并考虑以下的页号引用串：

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1



共发生15次缺页异常，12次页面置换

利用请求分页实现虚拟存储器

最近最久未使用（LRU）置换算法

其总是淘汰最近最久未使用的页面。

利用“最近的过去”作为“最近的将来”的近似。

假定系统为某进程分配了3个页框，并考虑以下的页号引用串：

7	0	1	2	0	3	0	4	2	3	0	3	2	1	2	0	1	7	0	1
7	7	7	2		2		4	4	4	0			1		1		1		
	0	0	0		0		0	0	3	3			3		0		0		
		1	1		3		3	2	2	2			2		2		7		

共发生12次缺页异常，9次页面置换

需要辅助的硬件支持

分页存储管理

分页存储管理有缺点么?

访问内存页表速度慢?

—— 快表

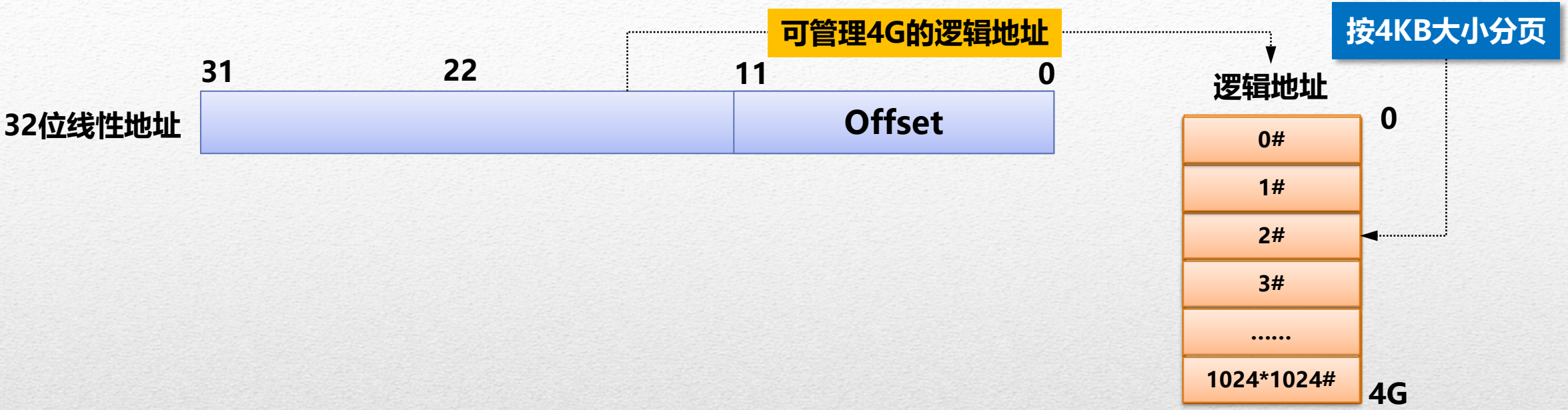
物理内存不够大?

——虚拟内存

页表太大?

i386线性地址空间

使用二级页表管理进程的线性地址

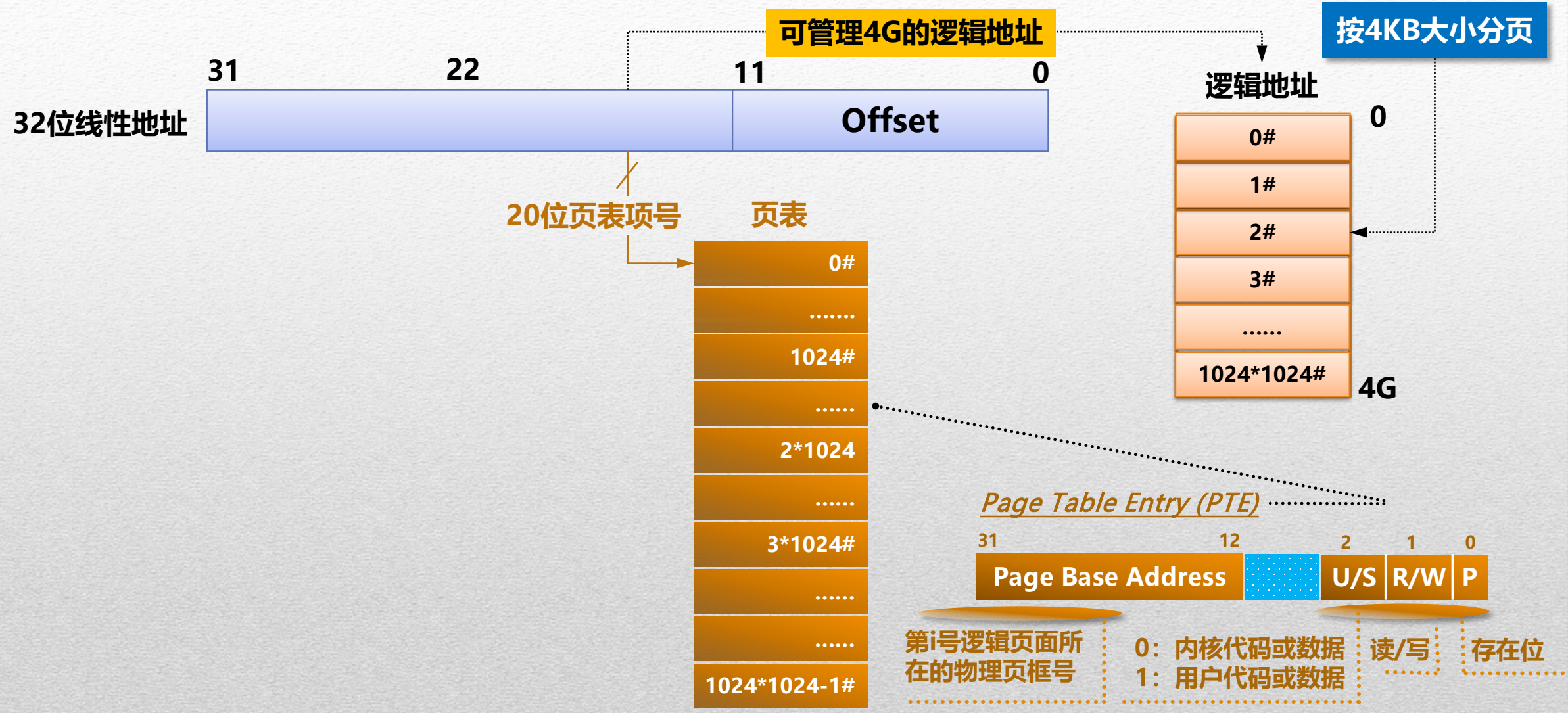


i386线性地址空间

使用二级页表管理进程的线性地址

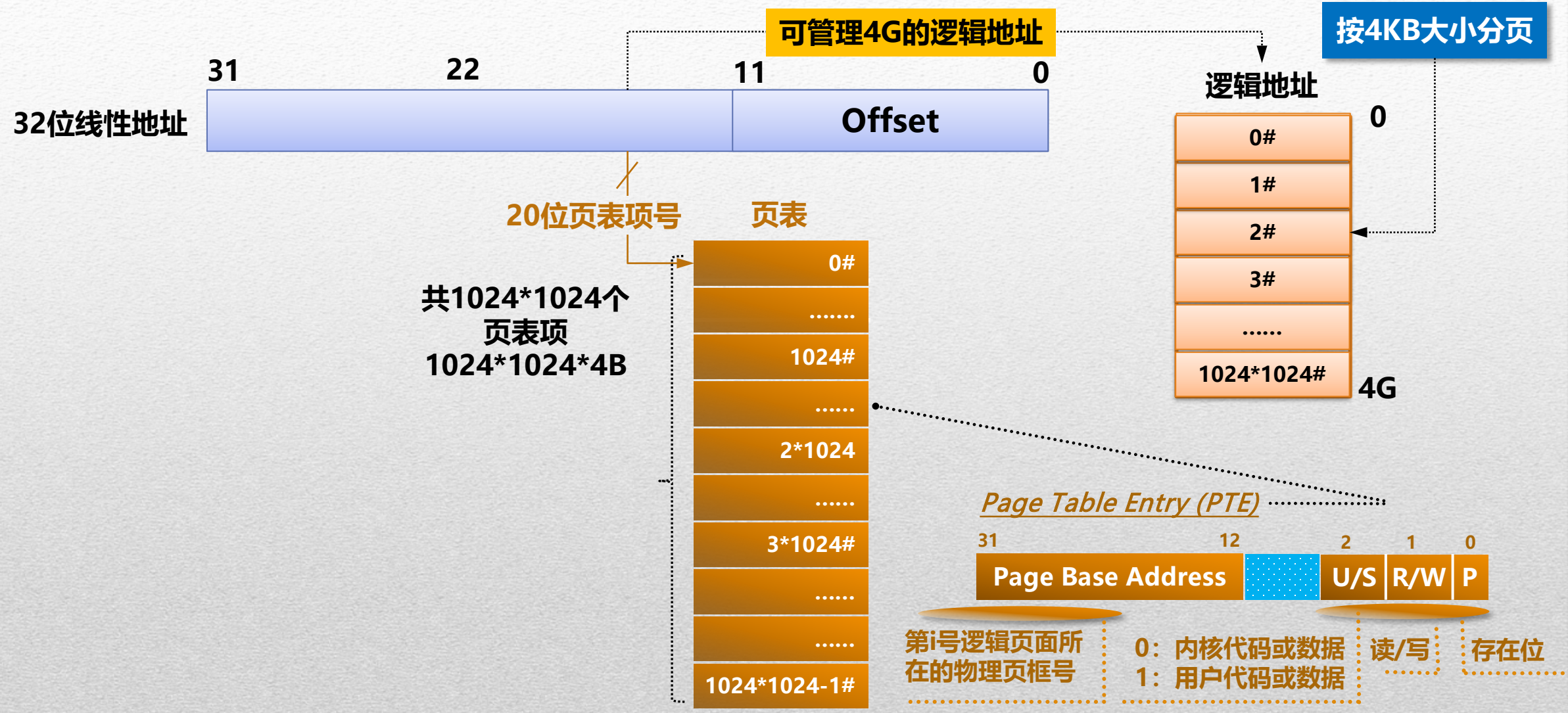
i386线性地址空间

使用二级页表管理进程的线性地址



i386线性地址空间

使用二级页表管理进程的线性地址



分页存储管理

分页存储管理有缺点么?

访问内存页表速度慢?

—— 快表

物理内存不够大?

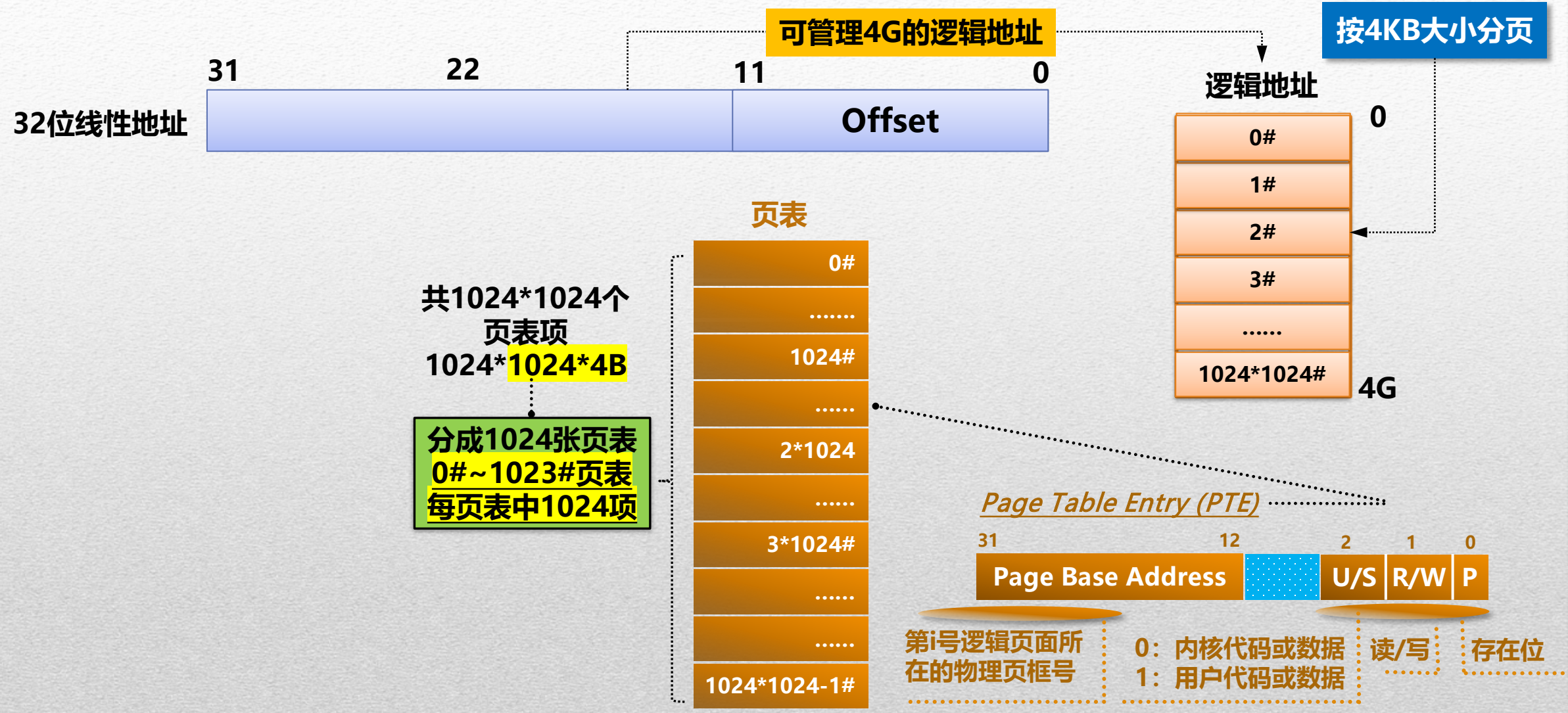
——虚拟内存

页表太大?

—— 多级页表

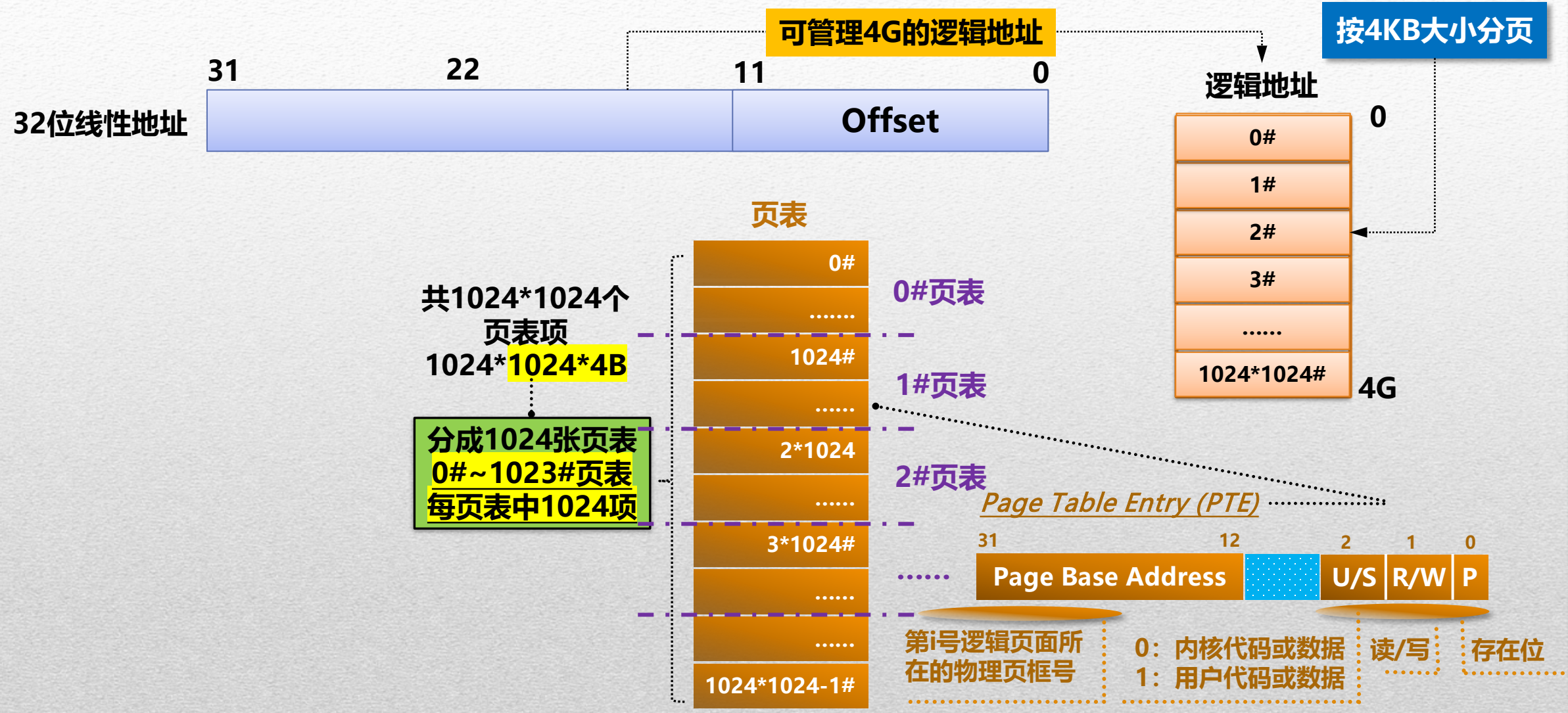
i386线性地址空间

使用二级页表管理进程的线性地址



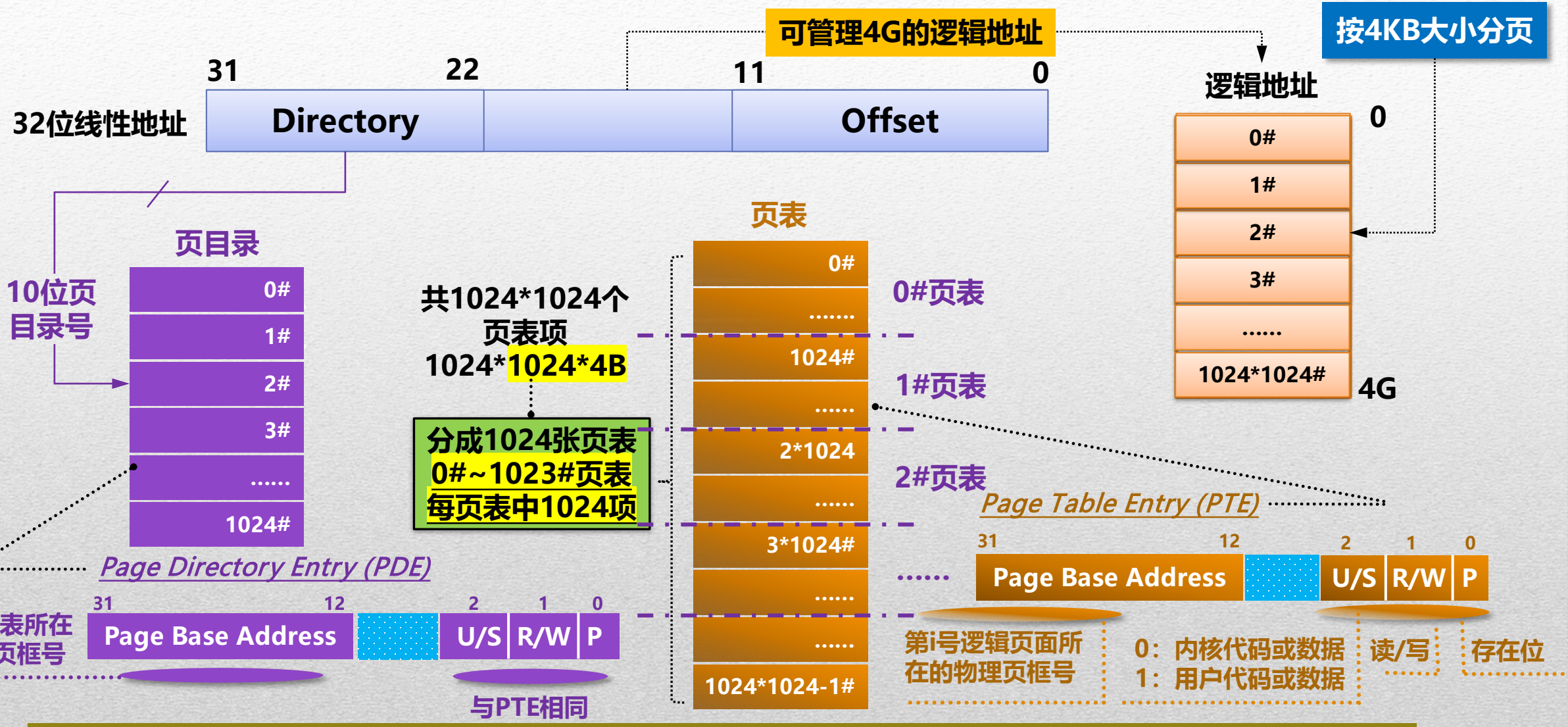
i386线性地址空间

使用二级页表管理进程的线性地址



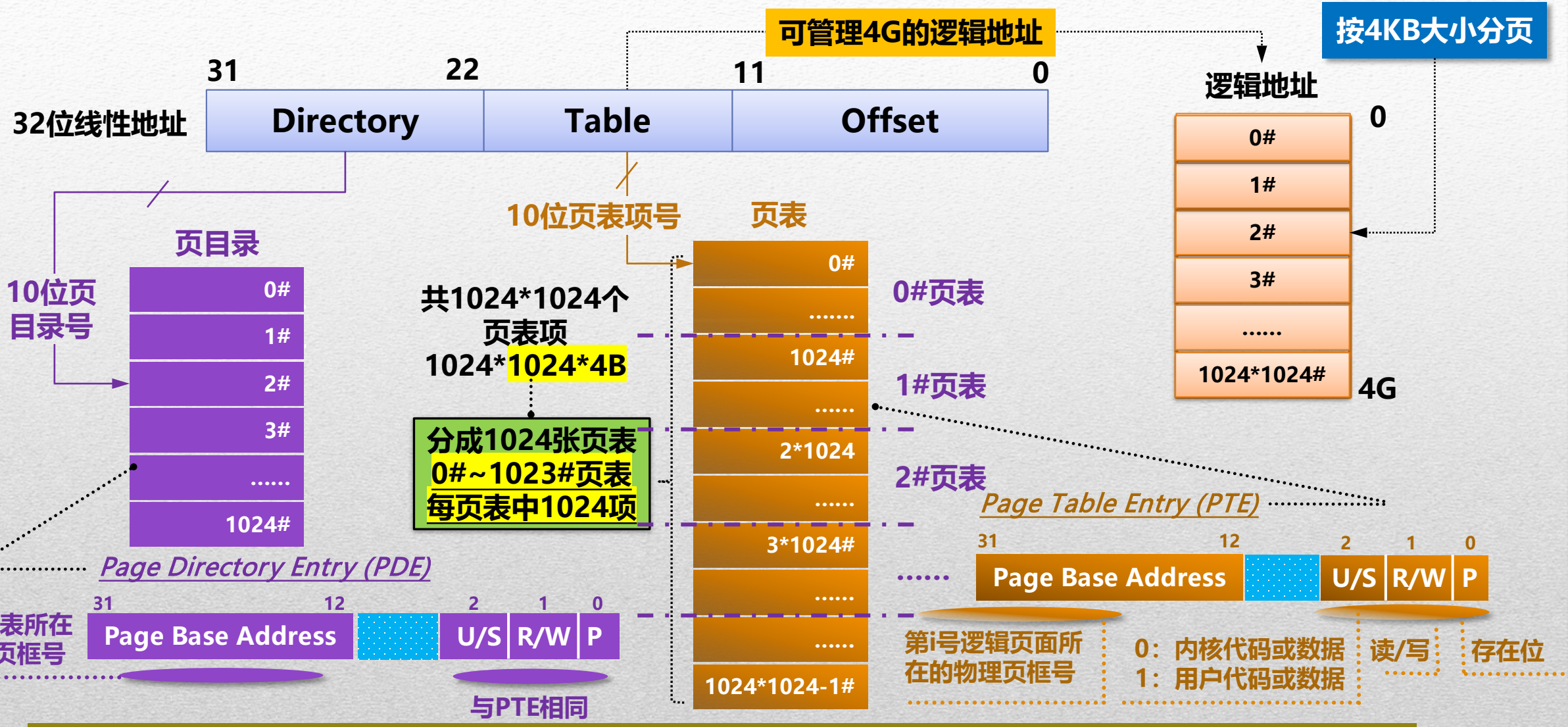
i386线性地址空间

使用二级页表管理进程的线性地址



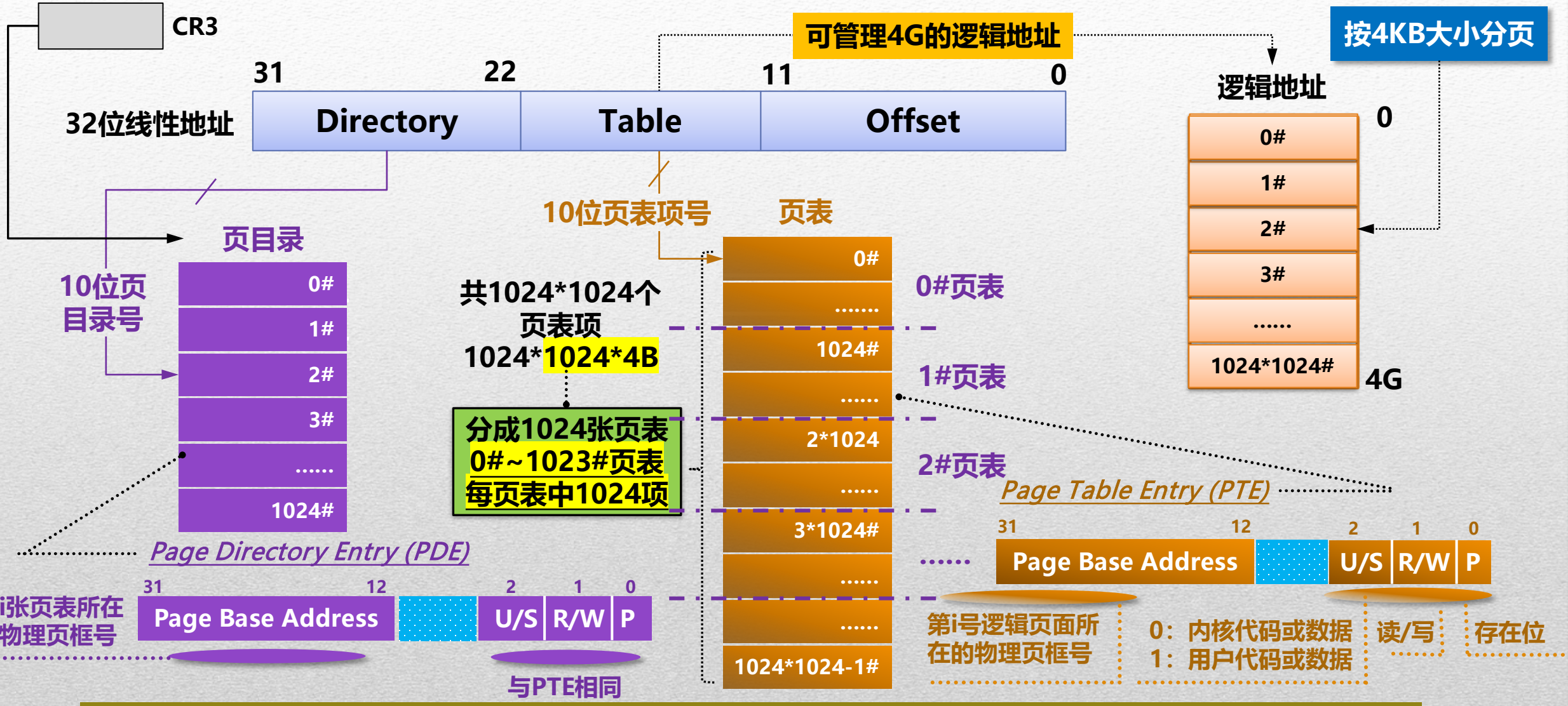
i386线性地址空间

使用二级页表管理进程的线性地址



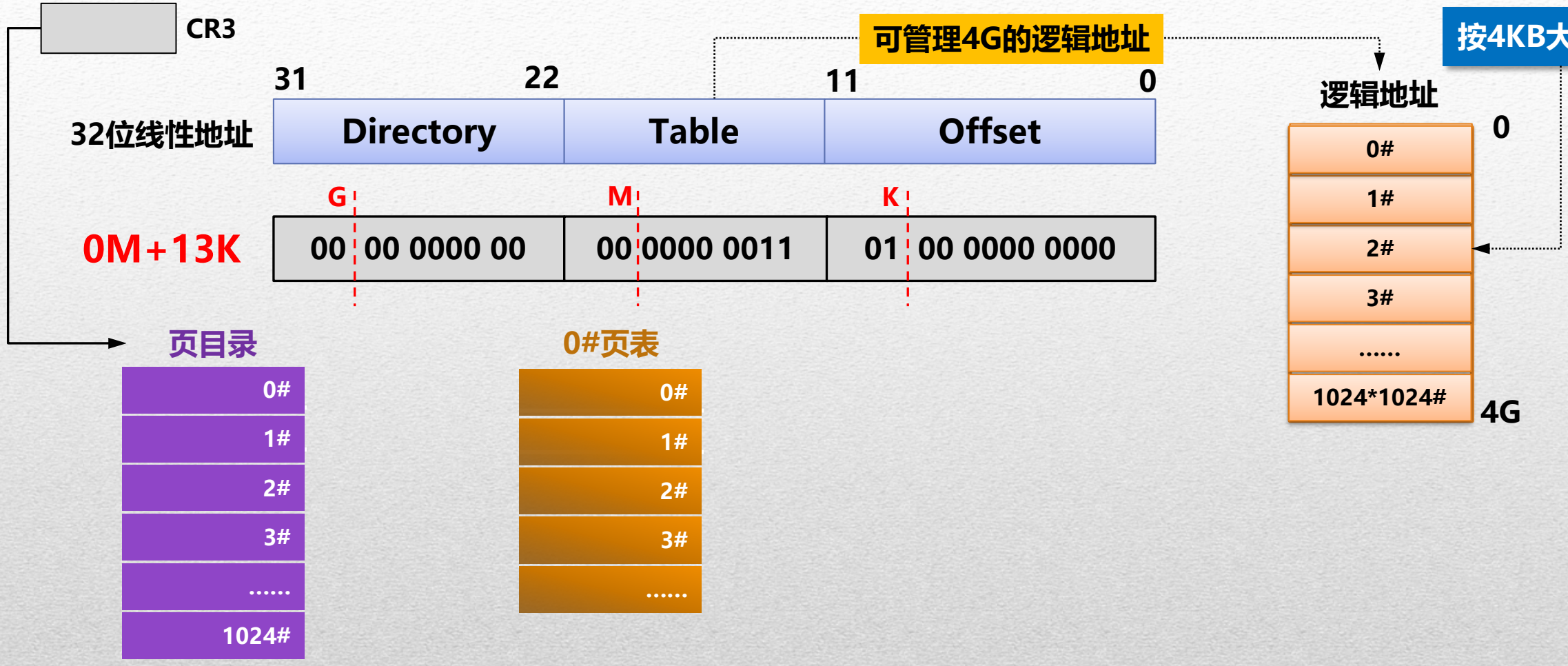
i386线性地址空间

使用二级页表管理进程的线性地址



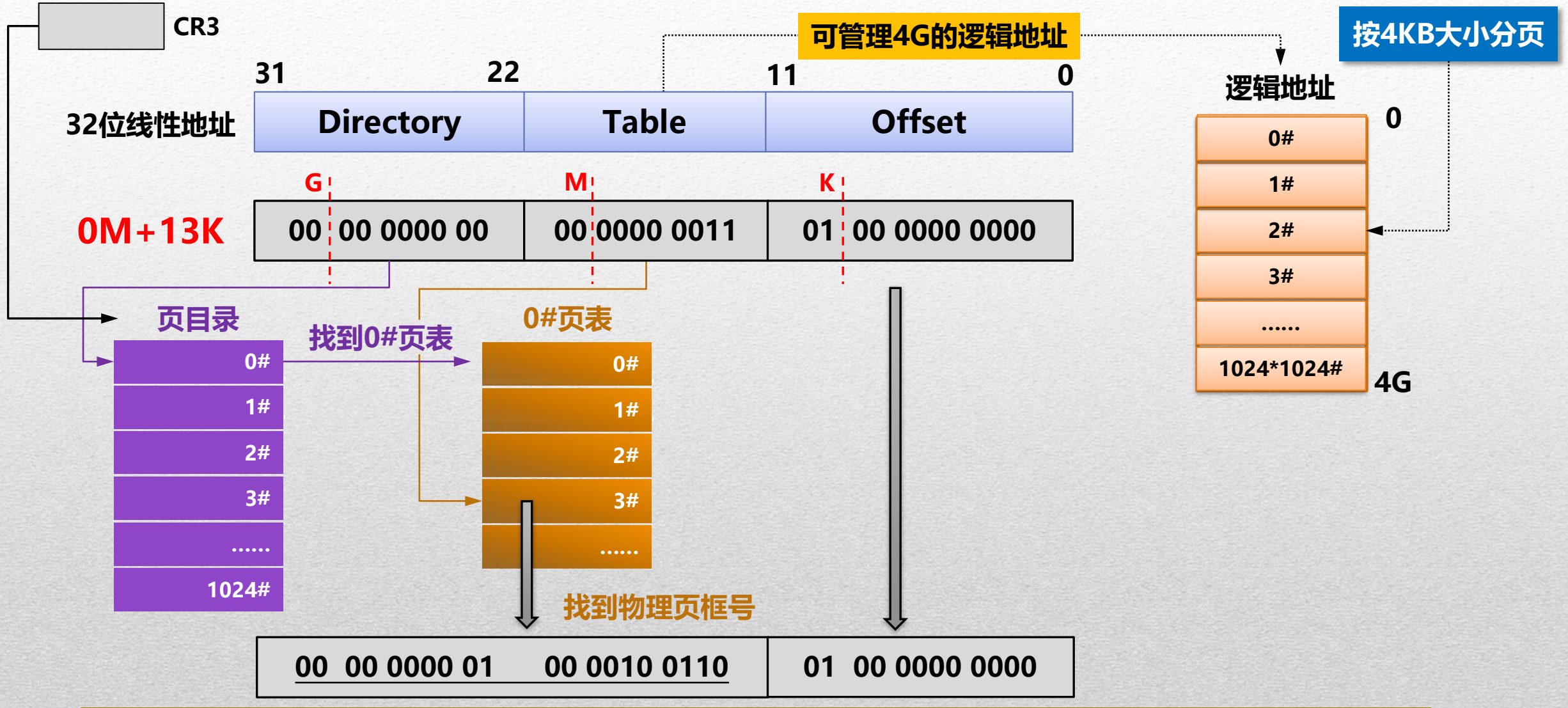
i386线性地址空间

使用二级页表管理进程的线性地址



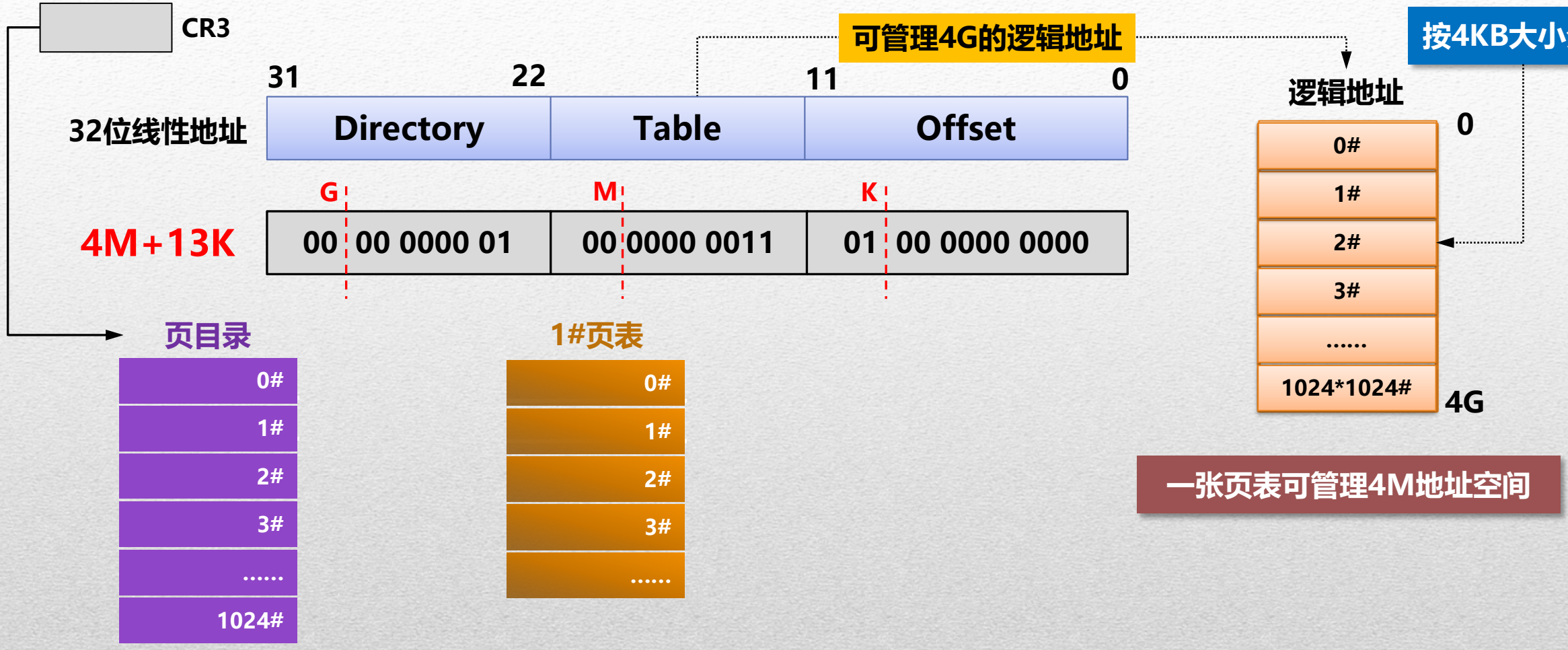
i386线性地址空间

使用二级页表管理进程的线性地址



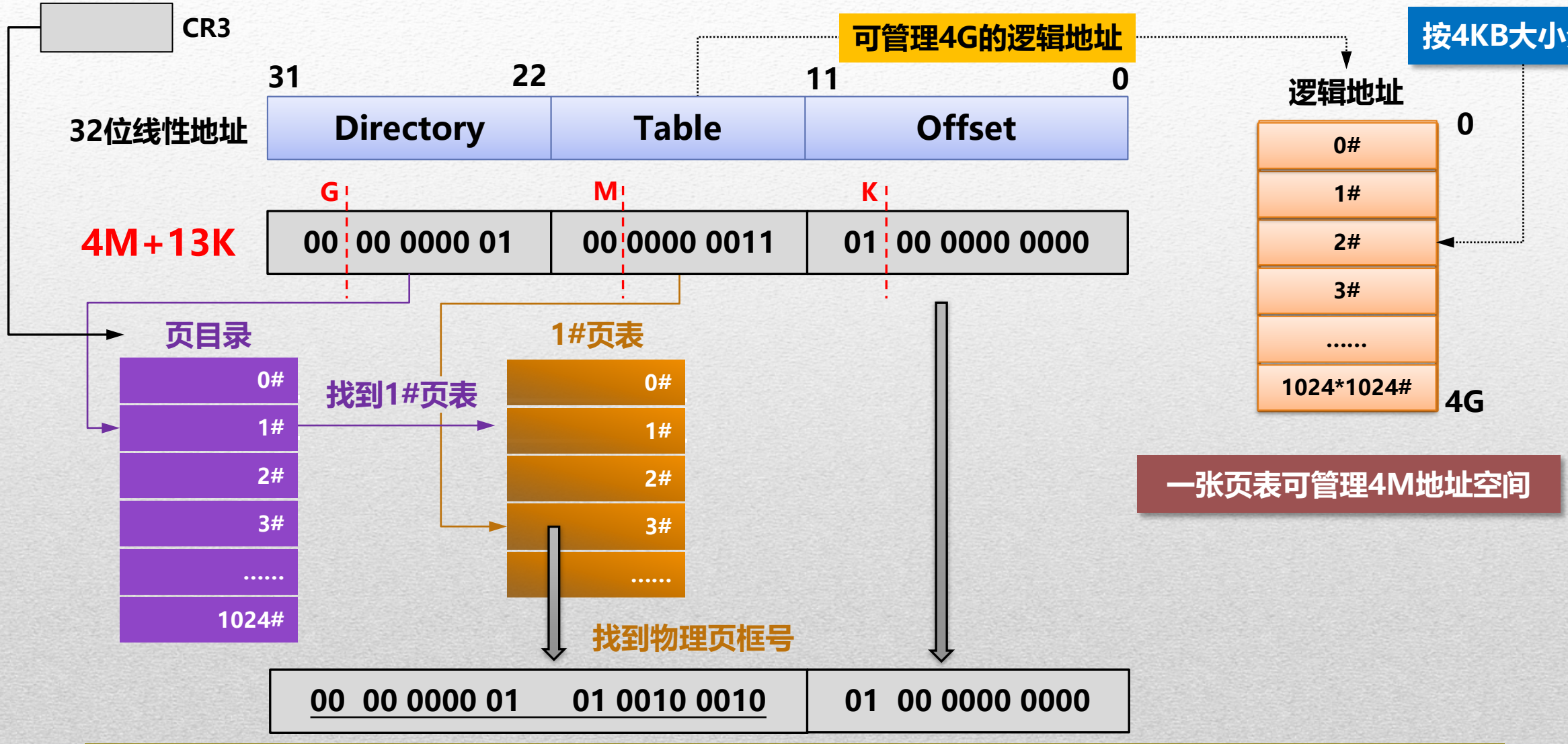
i386线性地址空间

使用二级页表管理进程的线性地址



i386线性地址空间

使用二级页表管理进程的线性地址

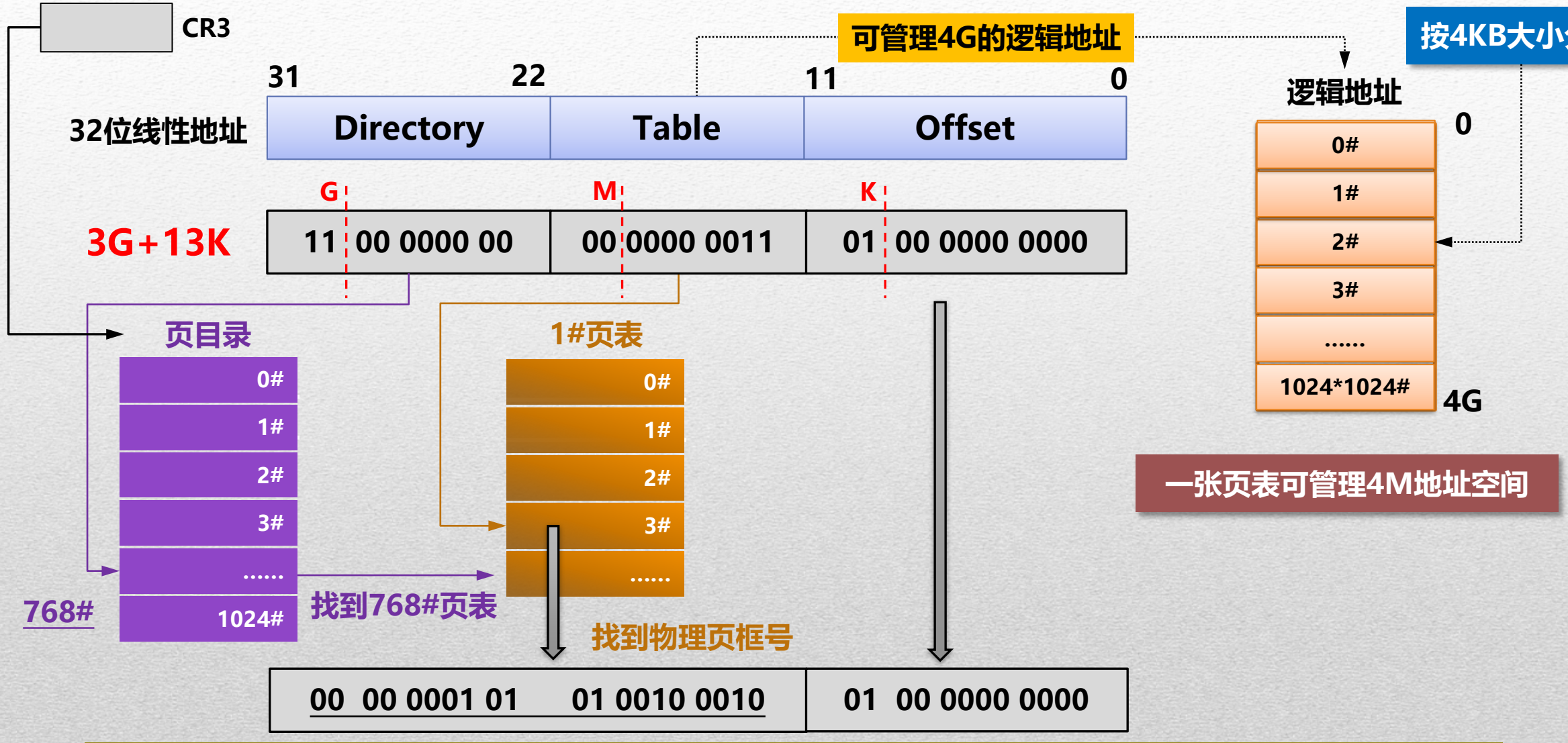


i386线性地址空间

使用二级页表管理进程的线性地址

i386线性地址空间

使用二级页表管理进程的线性地址



本节小结：

1 页式存储管理的基本思想

2 利用页表实现的地址变换过程

3 虚拟存储器

阅读讲义159~172页

E04：存储管理（连续分配、页式分配与虚拟存储器）