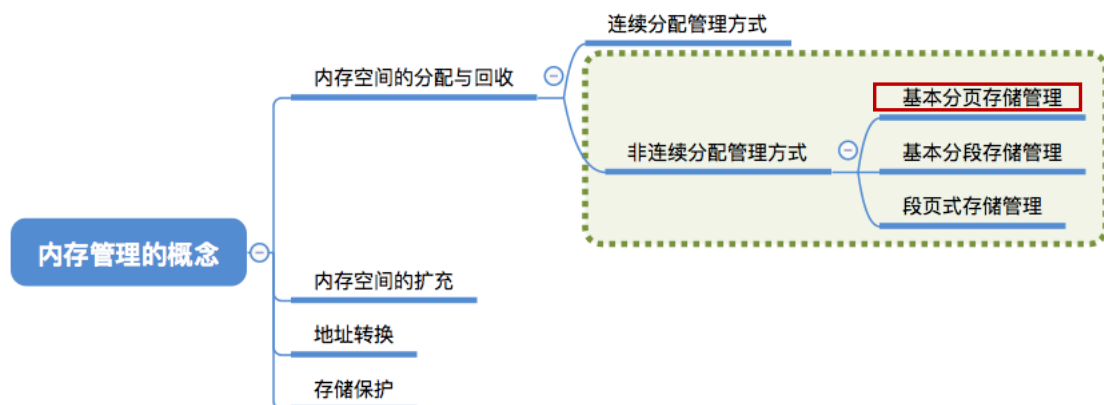


本节内容

# 基本地址变换机构

王道考研/CSKAOYAN.COM

## 知识总览



结合上一小节理解基本地址变换机构（用于实现逻辑地址到物理地址转换的一组硬件机构）的原理和流程

王道考研/CSKAOYAN.COM

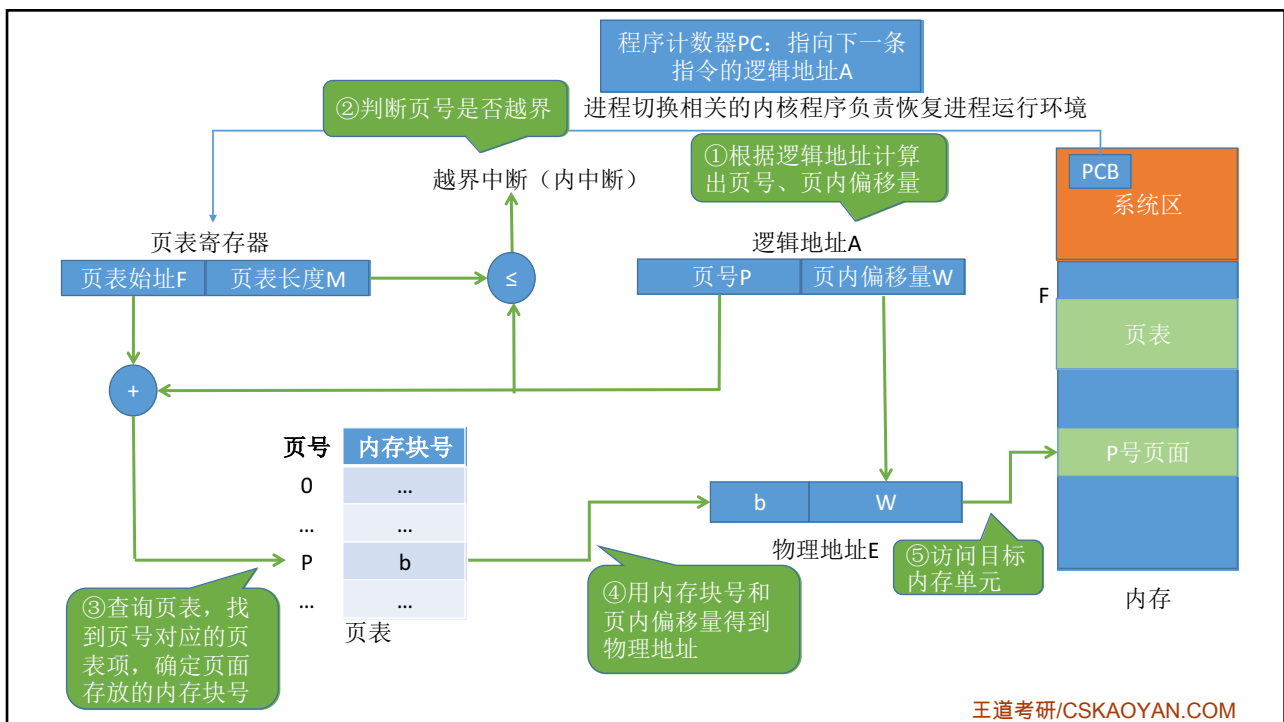
## 基本地址变换机构

基本地址变换机构可以借助进程的页表将逻辑地址转换为物理地址。  
 通常会在系统中设置一个**页表寄存器（PTR）**，存放**页表在内存中的起始地址F**和**页表长度M**。  
 进程未执行时，页表的始址和页表长度**放在进程控制块（PCB）中**，当进程被调度时，操作系统内核会把它放到页表寄存器中。

注意：**页面大小是2的整数幂**

设页面大小为L，逻辑地址A到物理地址E的变换过程如下：

王道考研/CSKAOYAN.COM





## 对页表项大小的进一步探讨

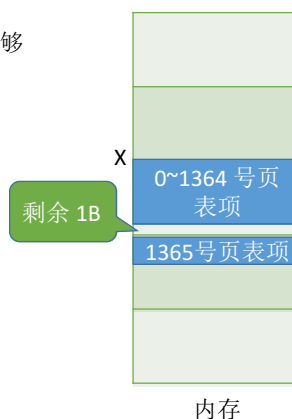
每个页表项的长度是相同的，页号是“隐含”的

Eg: 假设某系统物理内存大小为 4GB，页面大小为 4KB，的内存总共会被分为  $2^{32} / 2^{12} = 2^{20}$  个内存块，因此内存块号的范围应该是  $0 \sim 2^{20} - 1$   
因此至少要 20 个二进制位才能表示这么多的内存块号，因此至少要 3 个字节才够  
(每个字节 8 个二进制位，3 个字节共 24 个二进制位)

页号	块号
0	3 字节
1	3 字节
.....	3 字节
n	3 字节

页表

各页表项会按顺序连续地存放在内存中  
如果该页表在内存中存放的起始地址为 X，则  
M 号页对应的页表项是存放在内存地址为  $X + 3 * M$   
一个页面为 4KB，则每个页框可以存放  $4096 / 3 = 1365$  个页表项，但是这个页框会剩余  $4096 \% 3 = 1 \text{ B}$  页内碎片  
因此，1365 号页表项存放的地址为  $X + 3 * 1365 + 1$   
如果每个页表项占 4 字节，则每个页框刚好可存放 1024 个页表项



王道考研/CSKAOYAN.COM

## 对页表项大小的进一步探讨

每个页表项的长度是相同的，页号是“隐含”的

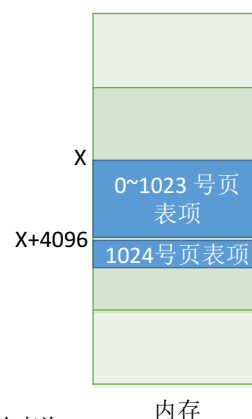
Eg: 假设某系统物理内存大小为 4GB，页面大小为 4KB，的内存总共会被分为  $2^{32} / 2^{12} = 2^{20}$  个内存块，因此内存块号的范围应该是  $0 \sim 2^{20} - 1$   
因此至少要 20 个二进制位才能表示这么多的内存块号，因此至少要 3 个字节才够  
(每个字节 8 个二进制位，3 个字节共 24 个二进制位)

页号	块号
0	3 字节
1	3 字节
.....	3 字节
n	3 字节

页表

各页表项会按顺序连续地存放在内存中  
如果该页表在内存中存放的起始地址为 X，则  
M 号页对应的页表项是存放在内存地址为  $X + 3 * M$   
一个页面为 4KB，则每个页框可以存放  $4096 / 3 = 1365$  个页表项，但是这个页框会剩余  $4096 \% 3 = 1 \text{ B}$  页内碎片  
因此，1365 号页表项存放的地址为  $X + 3 * 1365 + 1$   
如果每个页表项占 4 字节，则每个页框刚好可存放 1024 个页表项  
1024 号页表项虽然是存放在下一个页框中的，但是它的地址依然可以用  $X + 4 * 1024$  得出

进程页表通常是装在连续的内存块中的



结论: 理论上，页表项长度为 3B 即可表示内存块号的范围，但是，为了方便页表的查询，常常会让一个页表项占更多的字节，使得每个页面恰好可以装得下整数个页表项。

王道考研/CSKAOYAN.COM

