非连续内存的分配：

为什么要非连续内存分配：

连续内存分配的缺点：**碎片问题，内存利用率较低**

非连续内存分配：更好内存利用和管理，**更好的隔离，支持共享代码与数据，动态加载和动态链接**

如何建立虚拟地址和物理地址之间的转换：

软件方案的开销非常大。

硬件的两种方法：

分段和分页

分段管理机制：

内存空间怎么寻址：

计算机程序有各种段组成，代码段、数据段、栈段、堆段、符号表

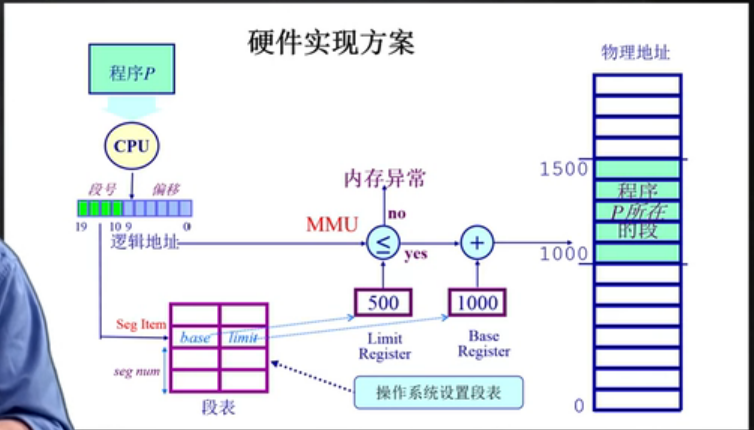
逻辑地址空间是连续的虚拟地址空间但是分段之后可以在物理地址中分离出来，隔离开

分段寻址方案：

将段的地址分成两块，一个段的地址（段号），一个段内偏移

即二元组（段号，段内偏移）

段寄存器存段号



**逻辑地址分成两部分，一部分是段号s，一部分是偏移addr，二元组（s,addr）**

通过**段号查询**进行的**段表（硬件机制）**，**每一个段号对应一个段描述符**，通过段表查询**逻辑地址的段号和物理地址的段号的对应关系**

段表中还存储着**段的起始位置和长度**

**MMU将偏移和长度进行比较，越界则内存访问异常**

**由操作系统建立段表**

分页机制：（主要采取的，分段用的比较少）

分页地址空间，和页寻址方式

和分段一样，也需要**页号和页内的偏移**，但和段不一样的是**段的尺寸是可变的**，但是**页帧的大小是固定的。**

**页帧（frame）的划分一般是2的整数幂，页帧的大小是固定的大小**

**逻辑页Page**也划分**同样的大小**

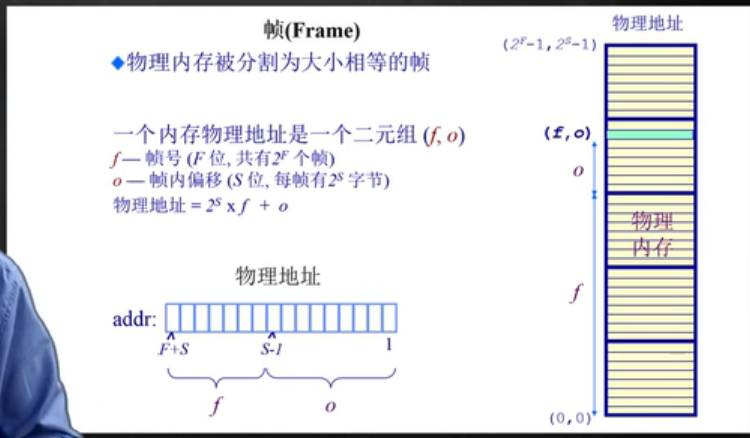
**页帧是物理页，而page是逻辑页**

转换**逻辑地址为物理地址**，用到**页表、MMU和快表TLB，TLB会对页表进行缓存**

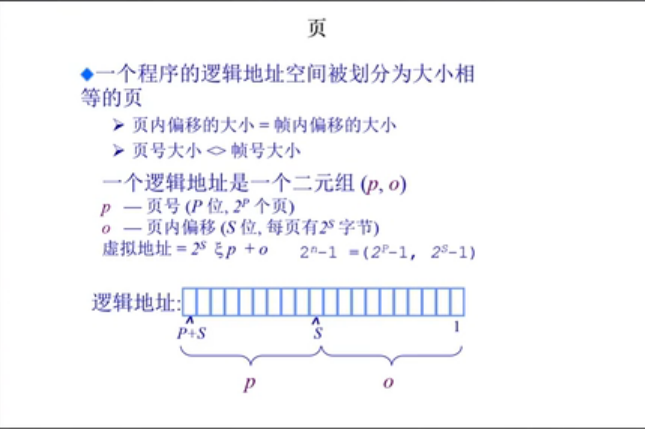
物理页帧的寻址方式：

物理地址，**物理内存被分割成大小相等的帧**

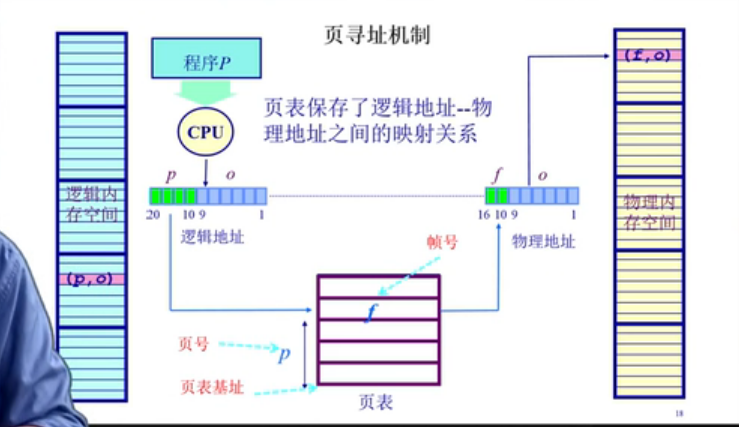
一个**内存物理地址为一个二元组（f,o），f是页帧号，o是帧内偏移**



逻辑页的寻址方式：page



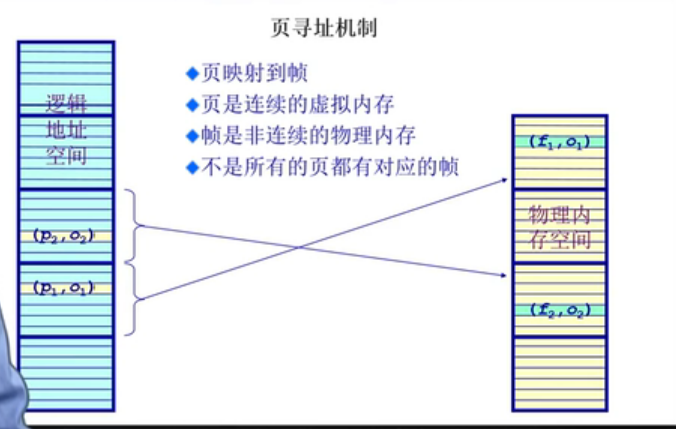
地址转换：



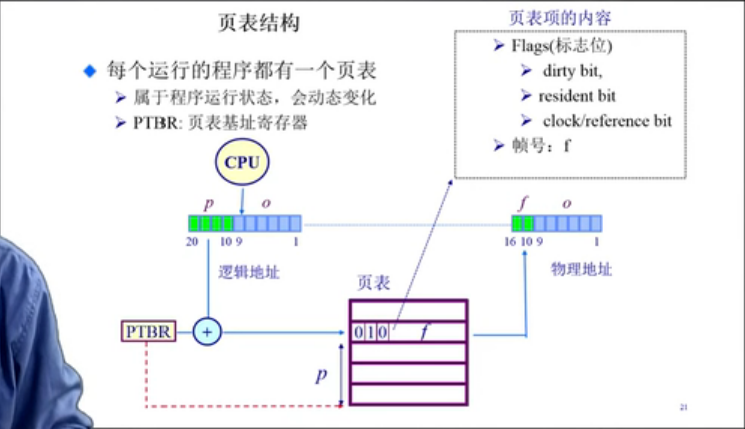
程序运行的时候**cpu会去寻址需要知道逻辑地址/虚拟地址**，根据逻辑地址的**页号P**，去查**page table（页表）**，可以通过**页号查询其对应的物联页帧的帧号**。对于**查询的时候也需要知道**这个页表从什么地方开始查，也就是**页表的基地址**。

由于**逻辑页的偏移和物联页帧的偏移一致**，因此我们找到对应的帧号之后加上偏移，便可以知道对应的物理地址。

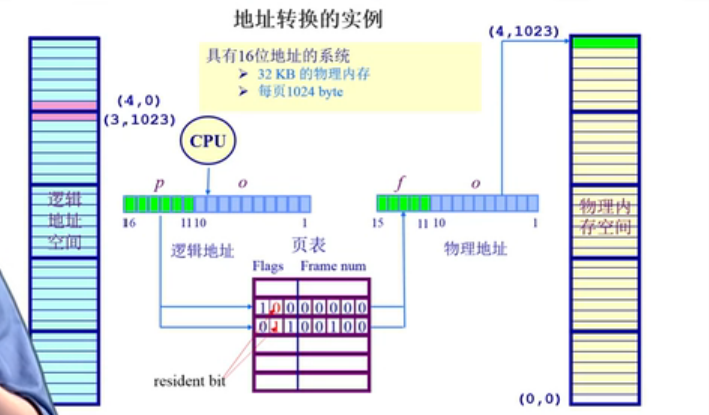
**Page table也是操作系统建立的**。



页表、TLB：



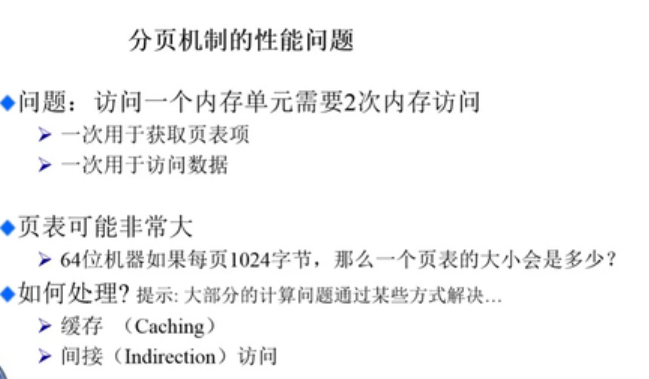
在**页表项中除了帧号F外**，还有**一些标志位，resident可以表示该页表项是否合法存在**，即**可能不存在对应的物理页**（虚拟地址）。



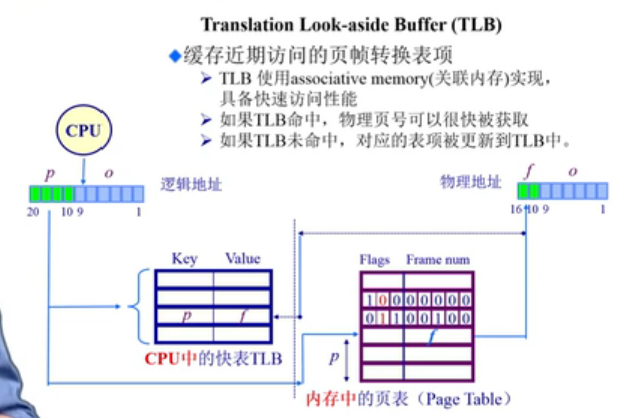
逻辑页有**16bit**，即**64kb地址空间**，**物理地址空间只有32kb**。每个页都是1024.

这里查页表对应的page，从下往上页号增加，从0开始，所以是0、1、1那个是页号为3对应的页表项。

（4,0）对应的**物理页实际不存在，cpu访问地址会触发内存访问异常**



**解决时间问题，MMU是内存管理单元，TLB是缓冲，缓存近期访问的页表的内容。**



**TLB里的表项是p和f，也就是逻辑页号，对应的帧号。Cpu中的TLB容量有限**，相关存储器，**速度很快，并发查找，**存的**经常访问到的页表项放于TLB。**

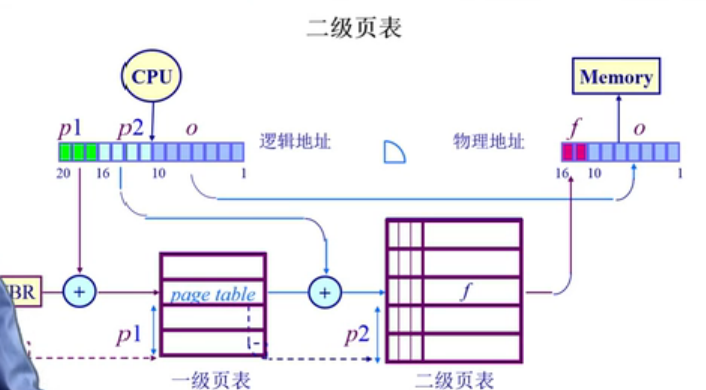
**Cpu得到逻辑地址先查TLB，如果TLB里没有的话就要去查内存中页表了，，如果这个物理帧合法则会将该页表项更新到TLB里。**

得到地址映射关系很快，不需要多次访问

空间问题

多级页表：

二级页表：



对于逻辑地址的page number拆分成两块，**但是偏移没有变**。

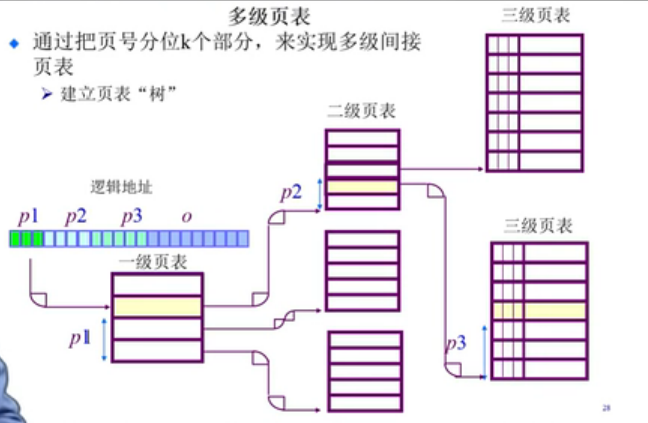
**一级页表的起始地址cpu是知道的。**

一级页表通过**p1,查询**，一级页表中存的不是f，而是**二级页表的起始地址**。

当知道二级页表的起始地址，**通过p2查询，得到对应的frame number**

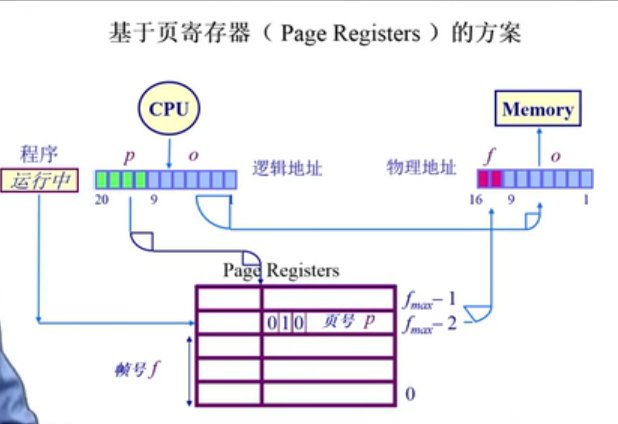
在多级页表下，一级页表中不存在的页表项其对应的二级页表就可以不再存在内存了。

多级页表：

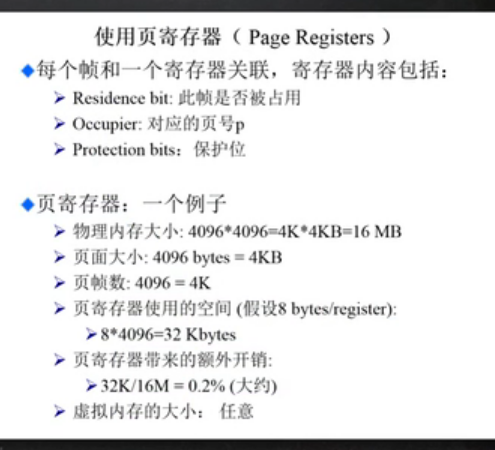


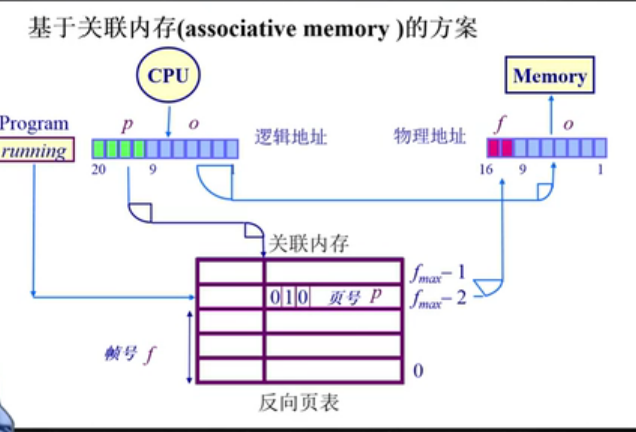
**以时间来换空间，时间开销很大，但是省空间了**。

反向页表：

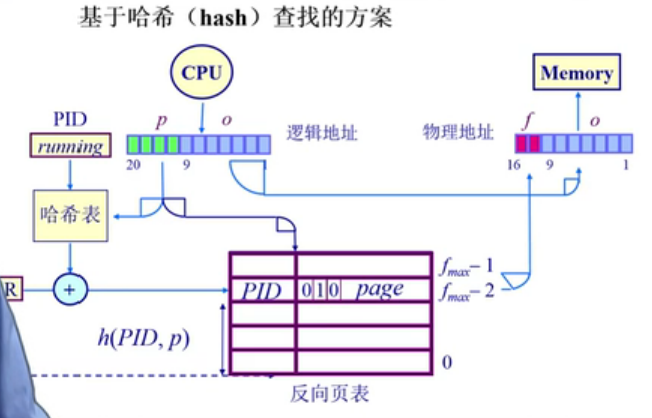


**用页帧号查找页号**





**关联存储器的开销太大，存在CPU里。**



但一个哈希可能有多个输出，哈希碰撞