内核操作系统复习概念

**逻辑地址：**

每一个逻辑地址都由一个段或偏移量组成，偏移量指从开始的地方到实际地址的距离。

**线性地址（虚拟地址）：**

32&64位的无符号整数来表示表达地址，例如线性地址常用16进制数字表示地址范围0x00000000到0xffffffff

**物理地址：**

从电脑上微处理器对的地址引脚发送到内存总线上的电信号

内存的控制单元MMU通过分段单元的硬件把一个逻辑地址转换为虚拟地址，然后分页单元的硬件把虚拟地址转换成物理地址

**段是什么：**

段描述了一块有限的内存区域，区域的起始位置存在专门的[寄存器](https://so.csdn.net/so/search?q=%E5%AF%84%E5%AD%98%E5%99%A8&spm=1001.2101.3001.7020" \t "/home/saltfish/文档\\x/_blank)（段寄存器）中。

**段是逻辑概念**，是虚拟[地址空间](https://so.csdn.net/so/search?q=%E5%9C%B0%E5%9D%80%E7%A9%BA%E9%97%B4&spm=1001.2101.3001.7020" \t "/home/saltfish/文档\\x/_blank)的基本单位，大小不固定，取决于我们当前运行的程序，**把虚拟地址转换为线性地址**  
段的三个方面来描述：**基地址，界限，保护属性**

**硬件中的分段**

处理器有两种不同的方式进行地址转换，这两种方式分别是实模式和保护模式，

实模式主要维持早期的模型兼容

重要的是保护模式

**逻辑地址的组成：**

是由一个段标识符和一个指定端内相对地址的偏移量组成的

**段标识符：**是一个16位的字段又称段选择符

**偏移量**：是一个32位长的字段

**如何找到段选择符：**

**处理器提供了段寄存器**，段寄存器唯一目的是存放段选择符（cs,ss,ds,es,fs,gs）但程序可以把同一个段寄存器用于不同的目的，所以就先将值保存在内存中用完之后再恢复。

**有专门的用途的寄存器：**

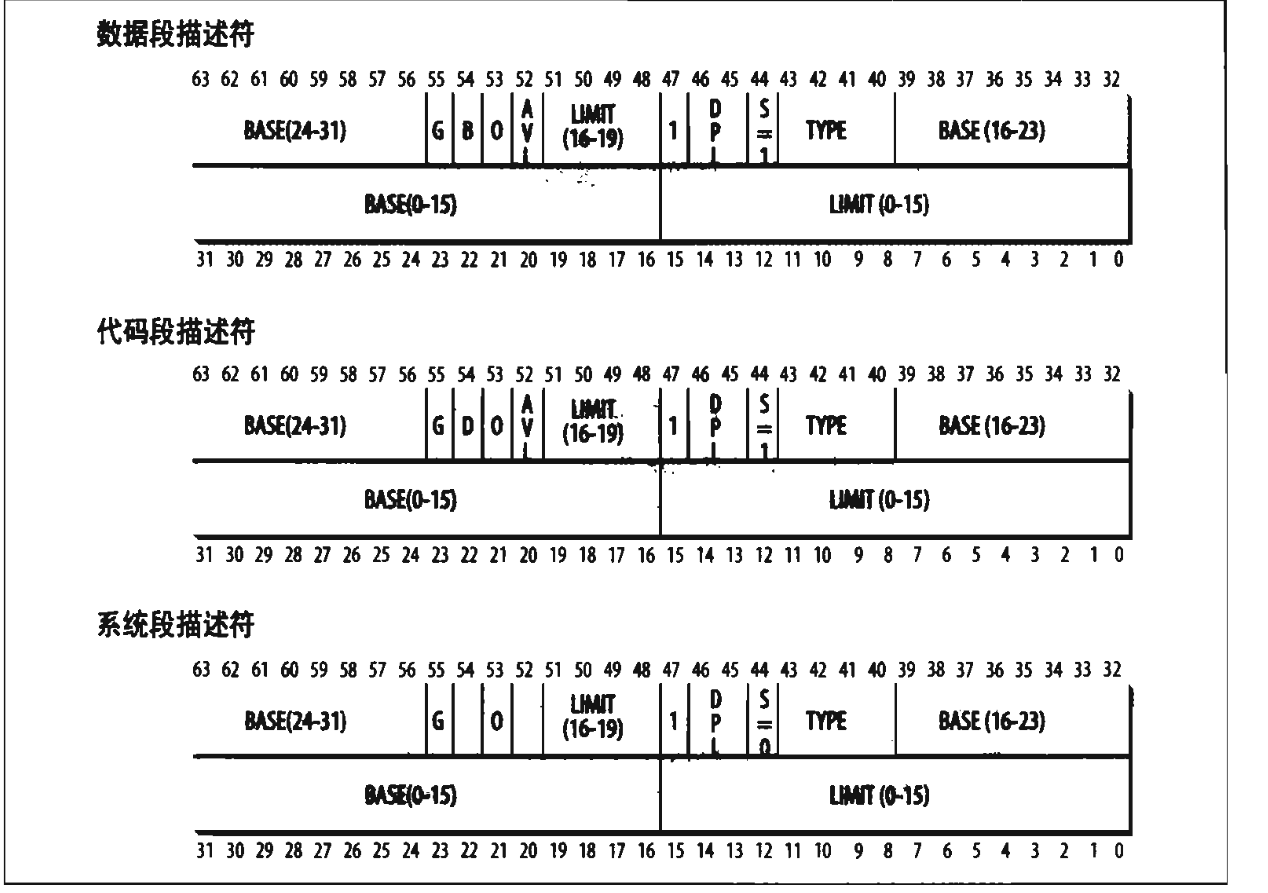
cs 代码段寄存器，包含程序指令的段，同时含有一个两位的字段指名cpu当前特权级0最高3最低用来分辨内核态和用户态

ss 栈段寄存器，指向包含当前程序栈的段

ds 数据段寄存器指向静态数据和全局数据段

计算机中每一个段都有一个段描述符，所有的段描述符放在全局描述符表GDT或者局部描述符表LDT

段描述符结构如下



**分页：**

分页是为了把线性地址转换成物理地址

线性地址被分为固定长度的组，就称为页，页内部连续的线性地址被映射到连续的物理地址中。

分页单元把所有的内存RAM分成了固定长度页框（物理页）每一个页框包含一个页，页框和页的长度一致。

分页单元处理4kb大小的页

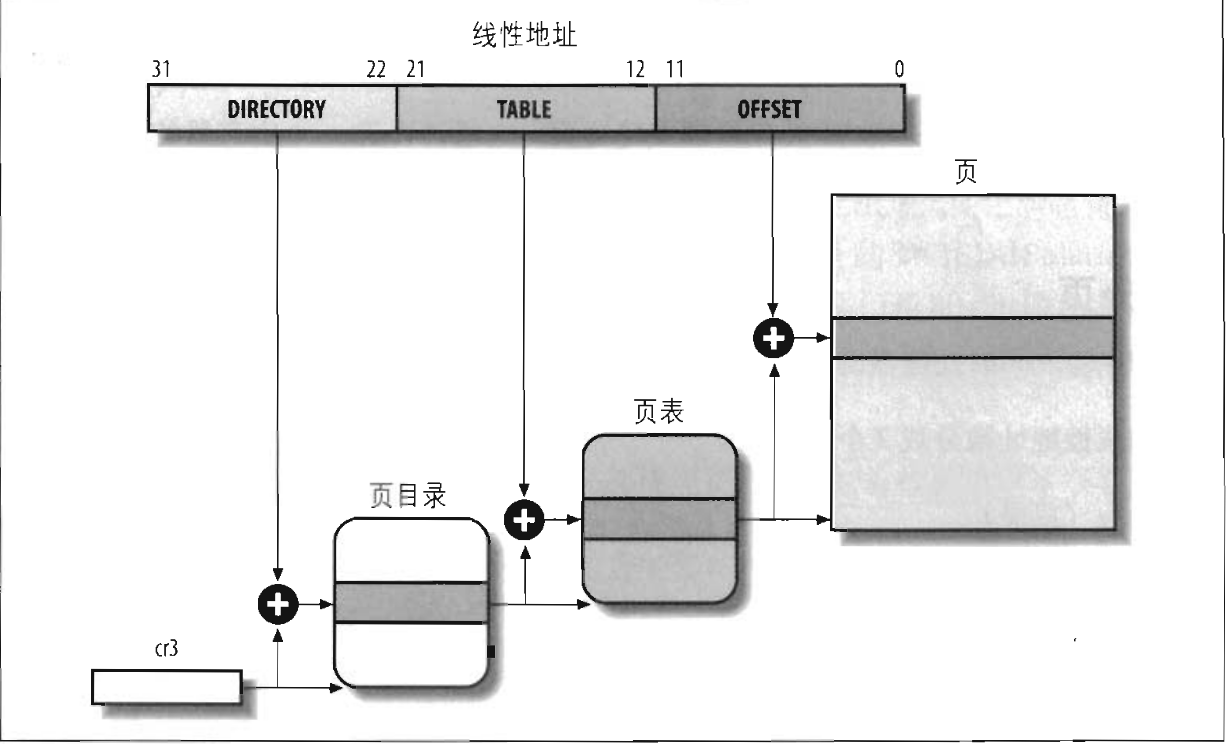
32位的虚拟地址被分成三个部分

directory（目录）10位 高

Table（页表）10位 |

offset（偏移量）12位 低

cr寄存器系列负责转换成物理地址和管理当前页目录的物理地址，虚拟地址内的页目录决定了目录项，目录项指向合适的页表，页表字段决定页表中的表项，表项中有页所在的页框物理地址。

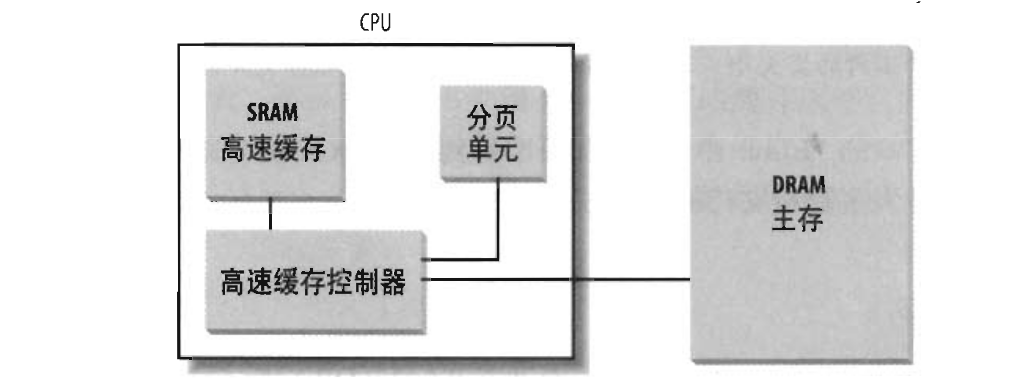


因为偏移量的字段决定页框内的相对位置，又是12位长，所以每一页就是4K

**硬件高速缓存**

动态的·RAM芯片存取时间要比时钟周期慢个几百倍，相当于操作RAM存放结果这样的指令cpu会等待很长时间，所以才有了硬件高速缓存

基于局部行原理得知最近常用的代码其实十分有意义所以才加上了这个缓存。



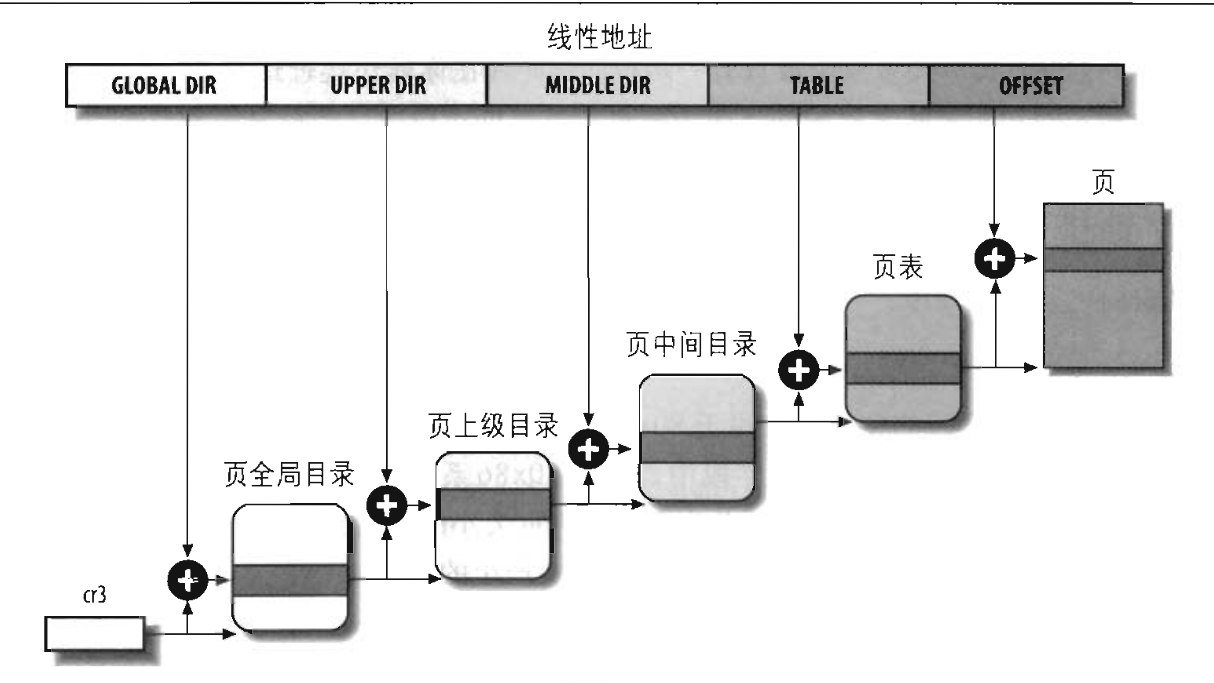
**Linux的分页是四级分页：**

页全局目录

页上级目录

页中间目录

页表



Linux把cr3控制寄存器的内容巴奥存在一个执行进程的描述符pid中把下一个要执行进程的描述符的值装进cr3里然后才完成了处理任务