逻辑地址：由一个段标识符和偏移量组成，表示从段开始的地方到实际地址之间的距离。

线性地址：又叫虚拟地址，32位无符号整数。最高可以表达4GB的地址。范围0x00000000到0xffffffff。

物理地址：内存芯片内存单元寻址，从CPU的地址引脚发送到总线上的电信号相对应。由32位或36位无符号整数表示。

内存控制单元MMU通过分段单元将逻辑地址转换为线性地址，然后通过分页单元将线性地址转换为物理地址。

段标识符也叫做段选择符，用来从段描述符表中选择具体的段。某个具体的段描述了一个段开始的线性地址。段标识符长度为16位。段偏移量长度为32位。段标识符格式：

|  |  |  |
| --- | --- | --- |
| Index | TI | RPL |

index：指定放在GDT或LDT中相应段描述符的入口。

TI：0表示段描述符在GDT中，1表示段描述符在LDT中。

RPL：请求特权级。当段标识符装入cs寄存器时，指示出CPU当前的特权级。还可以在访问数据段时有选择降低CPU的特权级。

CPU提供了一些段寄存器，用来存放段标识符。其中：

cs：代码段寄存器，指向存续指令的段。寄存器其中两位用以指明CPU当前特权级CPL。值位0代表最高优先级。值位3代表最低优先级。linux用0和3代表内核态与用户态。

ss：栈段寄存器，指向当前程序栈的段。

ds：数据段寄存器，指向静态数据或全局数据段。

段描述符是放在GDT全局描述符表或局部描述符表LDT中，8字节大小用来存放一个段的信息。包含以下字段：

Base：段首的线性地址。

G：粒度。如果为0，则段大小以字节为单位，否则以4096的倍数计算。

S：系统标志。如果为0，标识一个系统段。否则就是一个普通的代码段或者数据段。

GDT是全局段描述符表。在内存中的地址和大小放在CPU的gdtr寄存器。

LDT是局部段描述符表。在内存中的地址和大小放在CPU的ldtr寄存器。

代码段描述符和数据段描述符都可以放在GDT或LDT中，描述符S标志位1。

任务状态段描述符（TSSD）：用于保存CPU寄存器的内容，只能放在GDT中。根据其对应的进程是否在CPU上执行，其Type字段的值分别为11或9。S标志位为0。

局部描述符表描述符（LDTD）：只出现在GDT中。Type字段值为2，S标志位为0。

从CPU的段寄存器中得到段标识符。根据其中的索引值在描述符表找到段描述符。再根据段描述符的base地址和逻辑地址的偏移量，就可以找到对应的段。

为了能够快速完成逻辑地址到线性地址的转换。CPU提供了一种非编程的寄存器，用来存放段描述符。当段标识符放入段寄存器，对应的段描述符也就被放入非编程的寄存器。这样后续就可以直接根据段描述符得到线性地址。减少了对GDT或LDT的访问。

整理一下如何根据逻辑地址得到线性地址

1.检查段标识符的TI字段，判定段描述符是在GTD还是在LDT。如果在GTD，从gdtr寄存器得到GDT线性地址。在LDT，则从ldtr寄存器得到LDT的线性地址。

2.根据段标识符的index字段找到段描述符。段描述符线性地址 = GTD/LTD线性地址 + index \* 8。

3.段描述符的Base字段值 + 逻辑地址的偏移量就是逻辑地址对应的线性地址。

有了不可编程寄存器之后，可以省下前两个步骤。

对于linux系统，逻辑地址与线性地址总是一致的，即逻辑地址中的偏移量数值上与线性地址值相同。因为linux段描述符中的Base是0。