探究 x86 中实模式与保护模式

题目：

描述 x86 中实模式和保护模式的寻址区别，并说明在这两种模式中逻辑地址、线性地址和物理地址的关系

- 第 7 组 -

汇报人：胡峻豪，胡天磊

小组成员：胡才郁，张俊雄

时间： 2022.9.23

目录

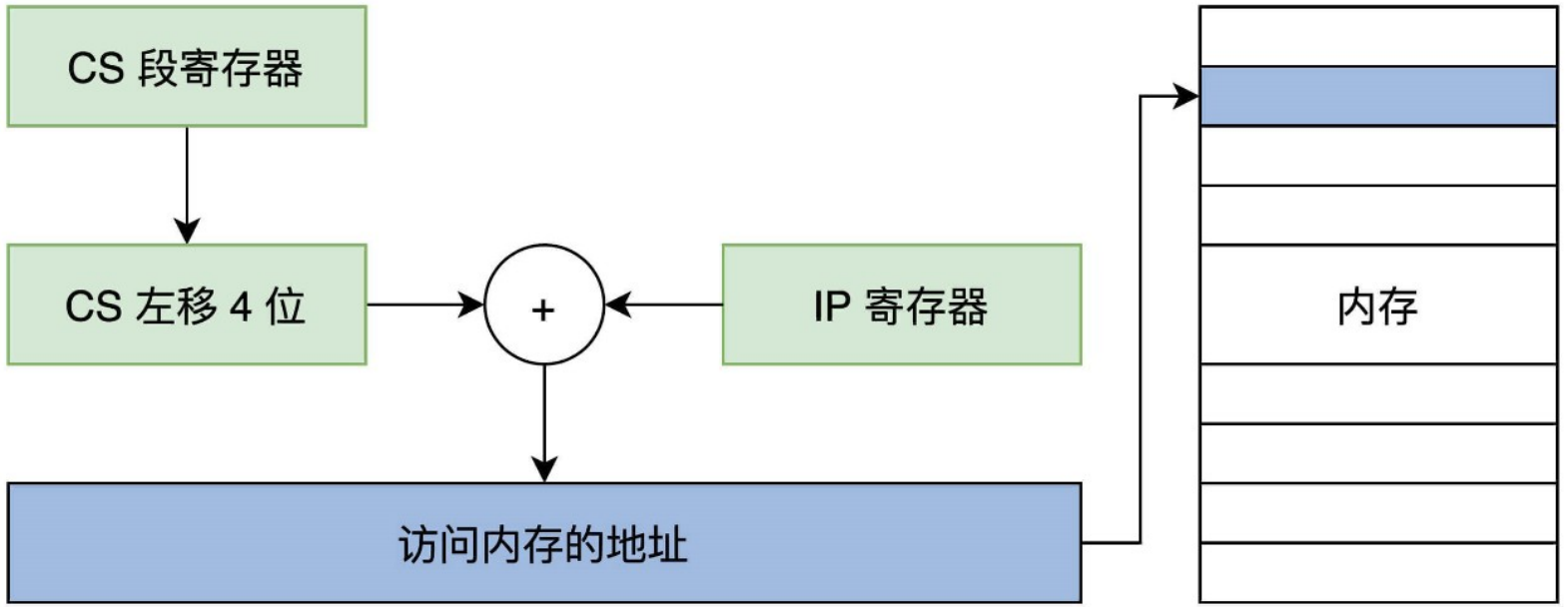
1.1 实模式寻址方式

X86 的实模式：
（1）运行真实的指令，执行指令真实的功能；
（2）访问内存的地址是真实的，对应的就是物理地址，不是“虚”的（开启 MMU 后的虚拟地址）

实模式被特殊定义为 20 位地址内存可访问空间上，这就意味着它的容量是 2 的 20 次幂（1M）的可访问内存空间，软件可通过这些地址直接访问 BIOS 程序和外围硬件。

所以为了在 8086/8088 下能够访问 1M 内存， Intel 采取了分段寻址的模式： 16 位段基地址 :16 位偏移 EA 。其绝对地址计算方法为： 16 位段基地址左移 4 位 +16 位偏移 =20 位地址。

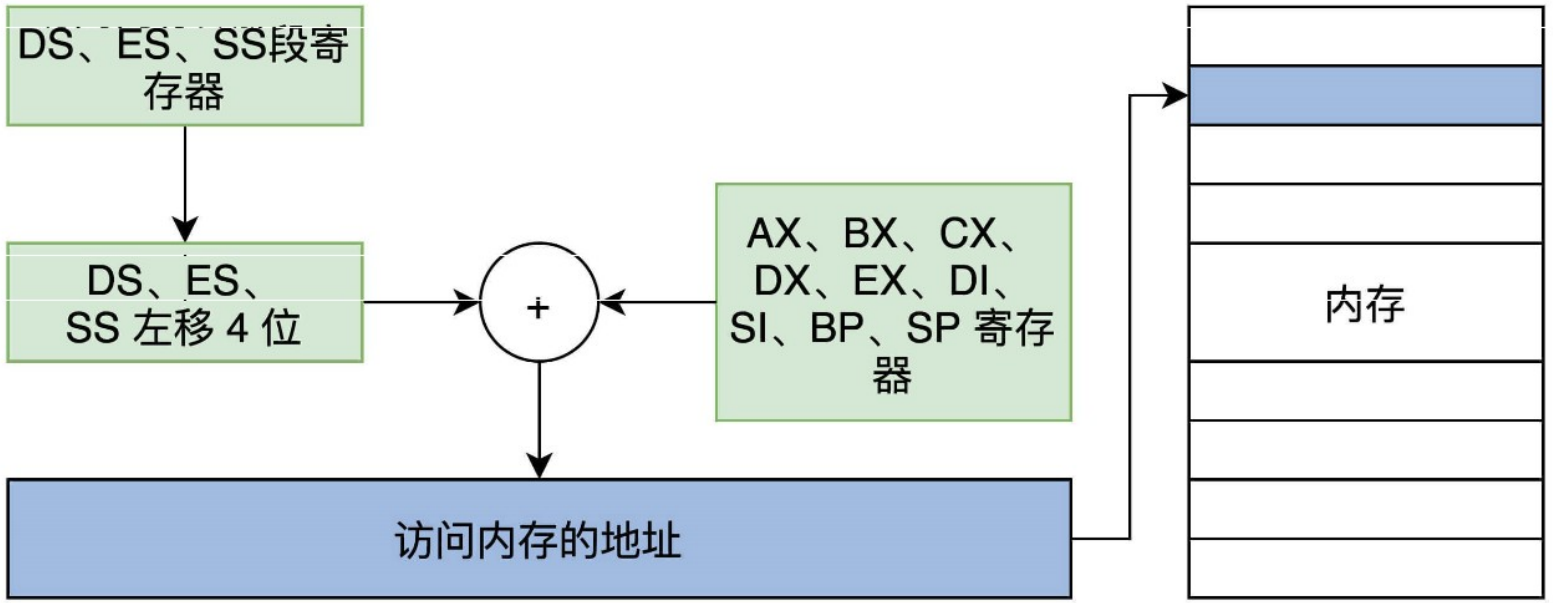
1.1 实模式寻址方式



取指

对于指令地址， CPU 会通过 CS 和 IP 寄存器的值组合而来，值为 CS 所存储的地址左移 4 位加 IP 的值：（CS << 4） + IP 。

1.1 实模式寻址方式



地址计算规则和指令地址计算规则类似

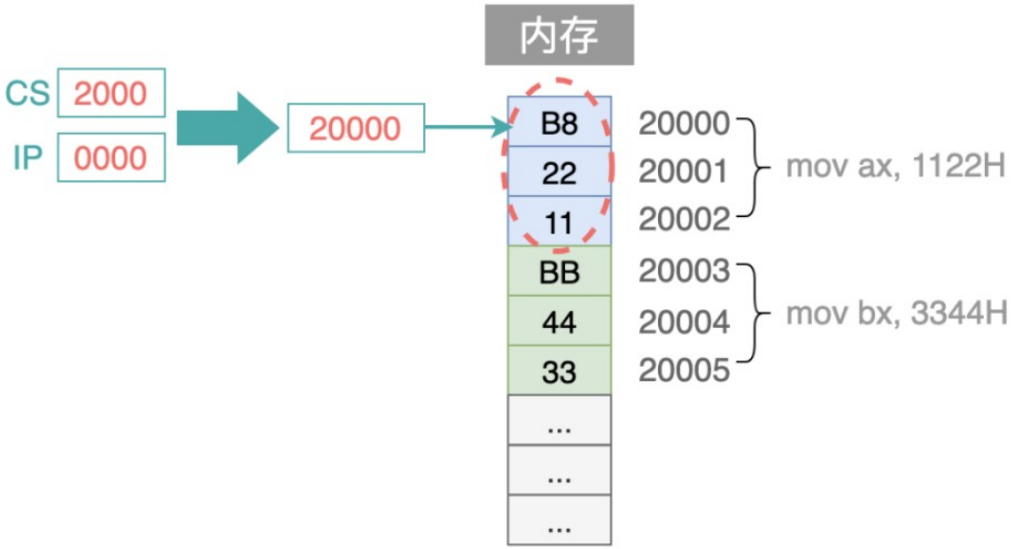
访问内存数据

1.2 实模式地址关系

为了充分利用地址空间，采用：段基址 + 段偏移 的方式，对 20 位的地址空间进行寻址。

（下一指令）物理地址 =16CS+IP，其中 CS 存放段基址， IP 存放段偏移，便恰好可以对 20 位地址空间进行寻址（对堆栈的访问则是 SS:SP；对数据段的访问是 DS:DI 或 DS:BX）。

以上就是分段机制，（段基址：段偏移）称为**逻辑地址**；（16 段基址 + 段偏移）就是**物理地址**（分段机制中，物理地址就是**线性地址**）



2.1 保护模式背景

Intel 首先在 80286 中提出了保护模式，但实际上它只是一个指引。 80286 虽然有了保护模式但其依然是 16 位的 CPU，其通用寄存器还是 16 位宽，只不过其地址线由 20 位变成了 24 位，即寻址空间扩大到了 16MB （但受限于寄存器位宽，单个寄存器的寻址空间仍然为 64KB ）。

直到 80386，保护模式才得到了极大的改善。它的地址总线和寄存器都是 32 位的，因此其单寄存器的寻址空间扩大到了 4GB 。

为了改进实模式下内存访问的不安全性，保护模式给内存段添加了**段属性**来限制用户程序对内存的操作权限。

下面是一个例子：
8086 有 20 根地址线，用户程序可通过 段地址 x16+ 偏移地址 的方式访问 1MB 内的任意地址。

```
mov ax, 0
mov ds, ax
mov byte [0x30], 66
```

这段代码首先将数据段地址置为 0x0000，然后将立即数 66 作为一个字节写入偏移量为 0x30 的地方，但是 8086 系统从内存地址为 0 起始的 1KB 空间是中断向量表，中断向量表就轻易地被破坏了。

能不能提供一种机制，**限制各自的程序只能在自己的空间运行**，彼此隔离，大家各守其位，各尽其职，谁也不能代俎越庖影响到其他程序。基于这样的诉求，X86 架构处理器的**保护模式**应运而生。

2.2 保护模式管理方式

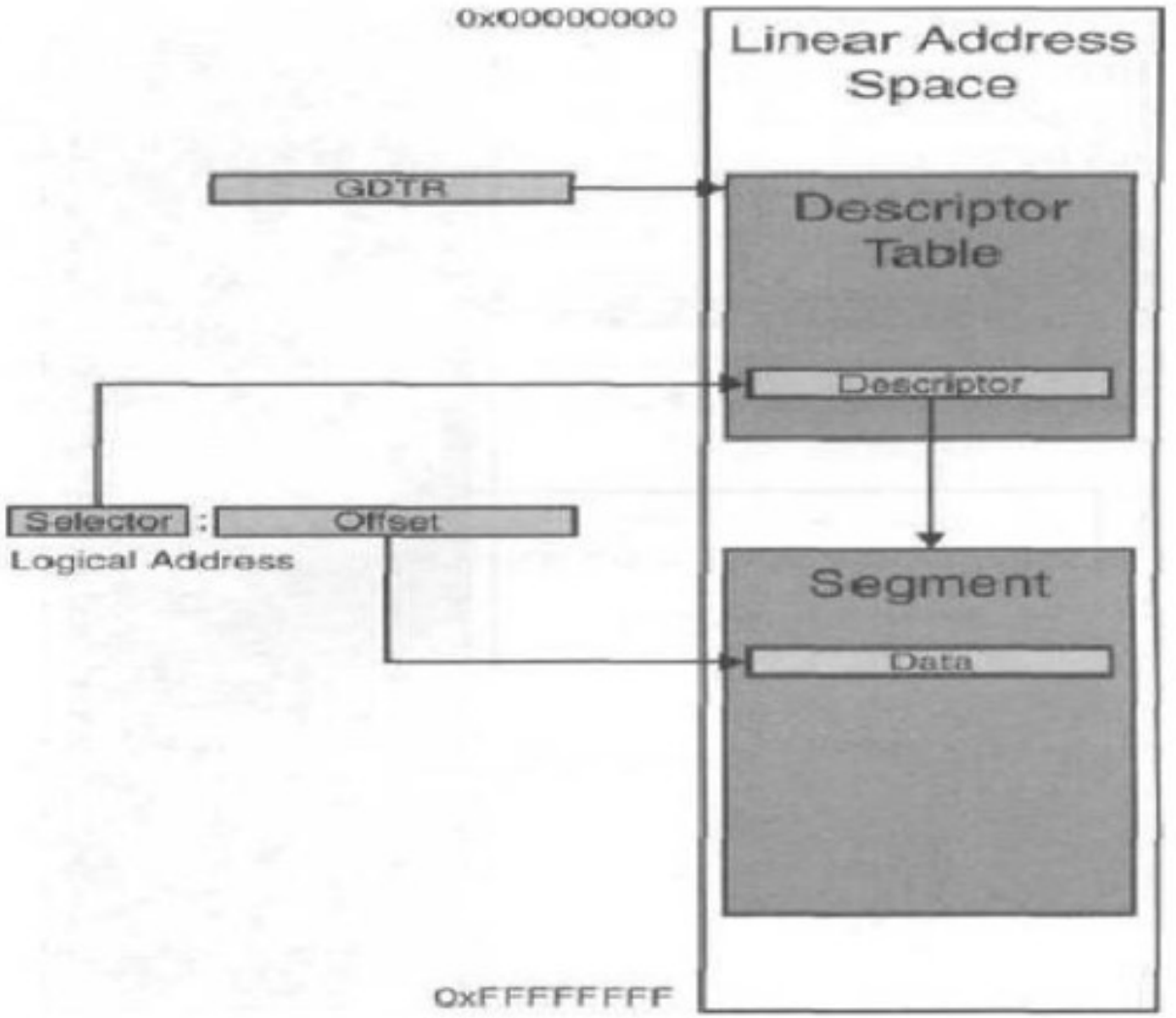
在保护模式中，内存的管理模式分为两种——段模式和页模式。其中页模式也是基于段模式的。也就是说，保护模式的内存管理模式事实上是：纯段模式和段页式。进一步说，段模式是必不可少的，而页模式则是可选的——如果使用页模式，则是段页式，否则这是纯段模式。

2.3 保护模式下地址关系

在保护模式下，我们的偏移值从 20 位变成了 32 位，存放在 eip 寄存器下。其中段选择符 (段基址)+ 偏移量就是逻辑地址；逻辑地址经过分段部件变换成为线性地址；如果不分页，得到的线性地址就是物理地址。如果分页，则线性地址要经过分页部件变换后才是物理地址。

2.4 段模式下寻址方式

相比于实模式，保护模式下分段机制是利用一个称作段选择子的偏移量，从而到描述符表找到需要的段描述符，而这个段描述符中就存放着真正的段的物理首地址，再加上偏移量，就找到真正的物理地址。



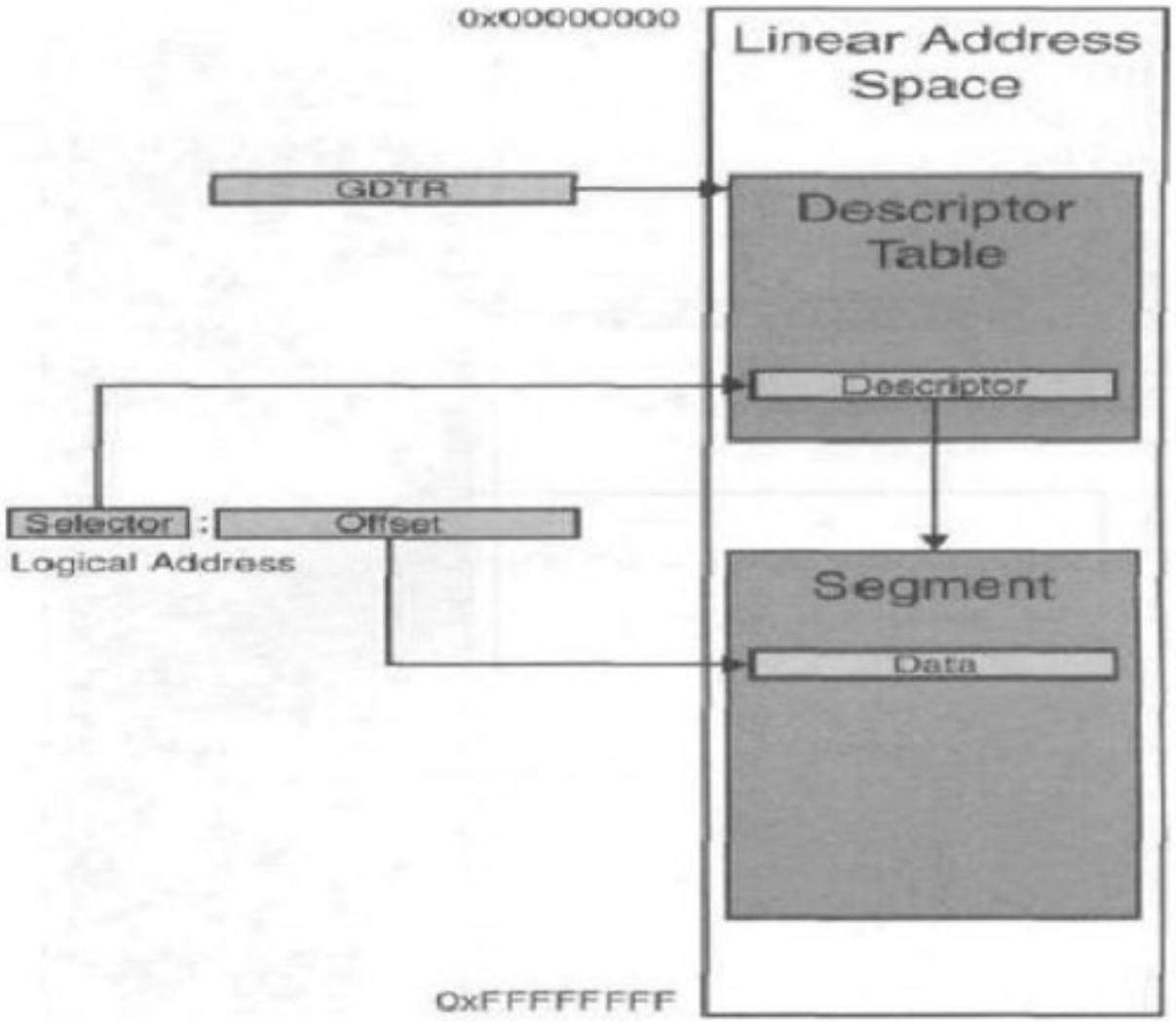
2.4 段模式下寻址方式

1. 段描述符

保护模式引入了全局描述符表（Global Descriptor Table， GDT）， GDT 存放在内存中，只有一张且全局可见。

GDT 的每一项是描述段类型属性的数据结构——段描述符，一个段描述符占 2 个存储单元。 GDT 中的每一个段描述符都描述了一个内存段的基本属性，如段基址、段界限、类型、 DPL 等等。

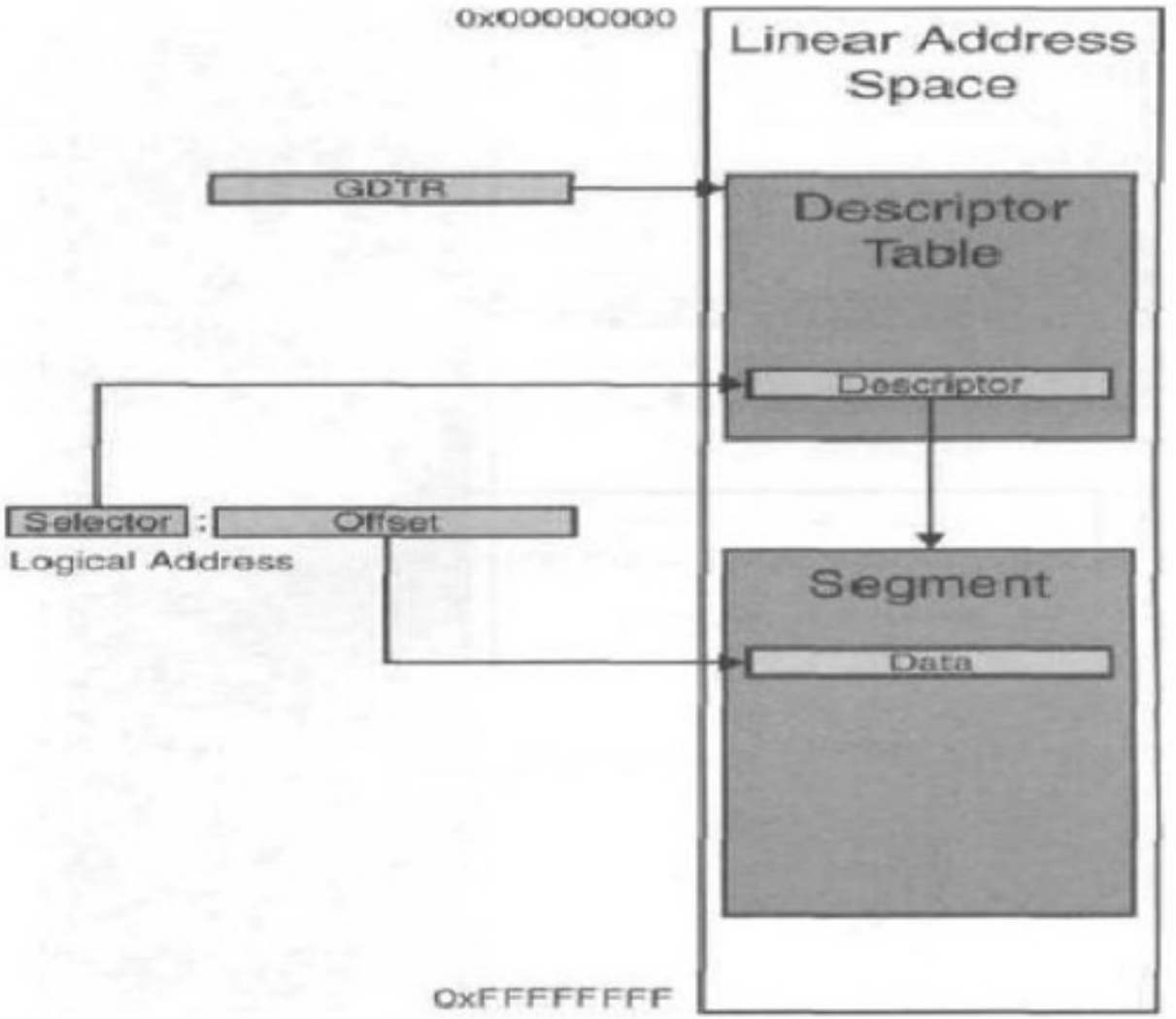
GDT 有很多描述符：代码段描述符、数据段描述符、栈段描述符。



2.4 段模式下寻址方式

2.GDTR
保护模式下增加了两个寄存器， GDTR 和 LDTR 新增的寄存器可以不和上个版本兼容不是 16 位，是 32 位。

GDTR 其全称为 Global Descriptor Table Register （全局描述符表寄存器）。专门用来存放段描述符表的首地址，以便找到内存中段描述符表。

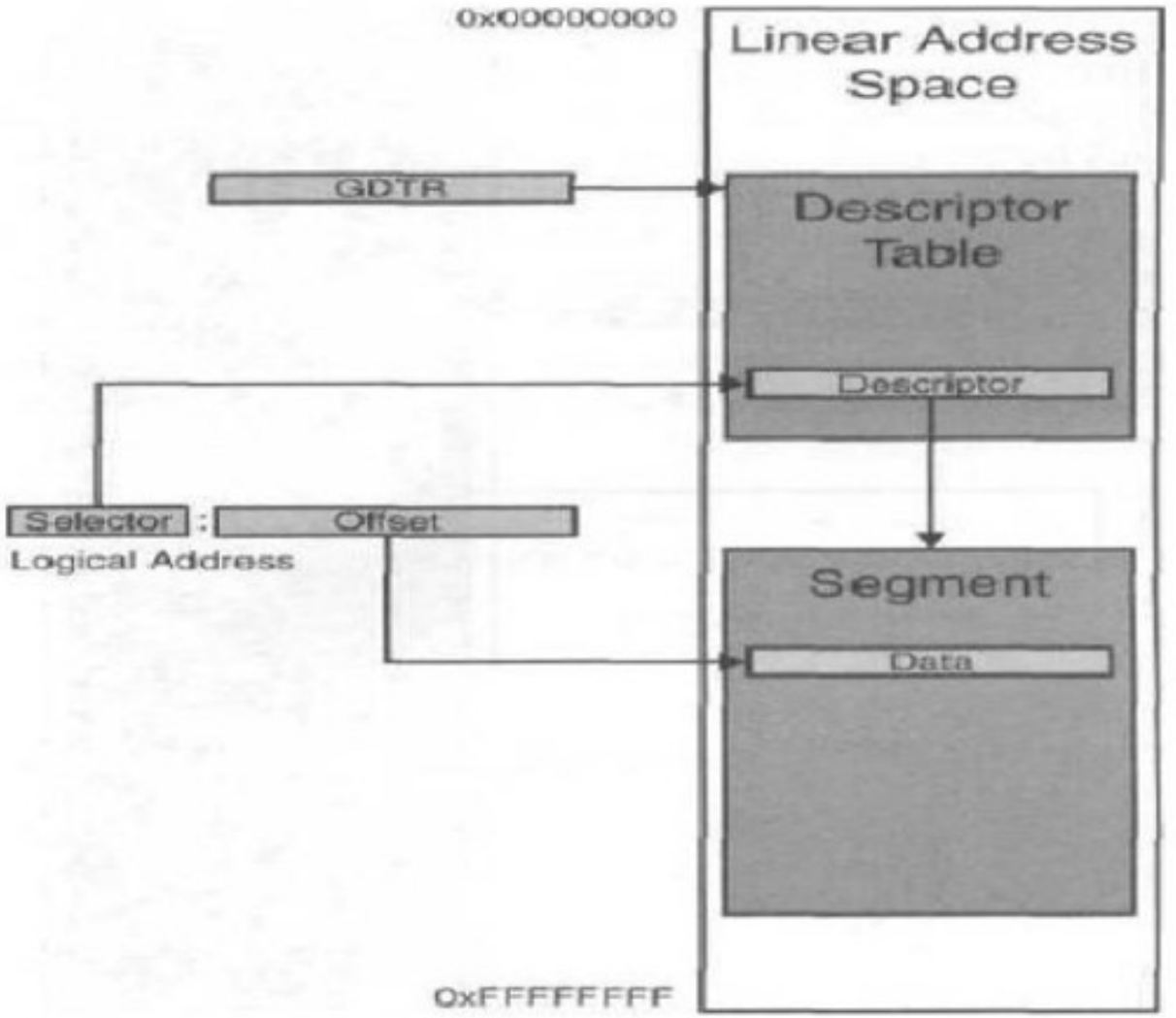


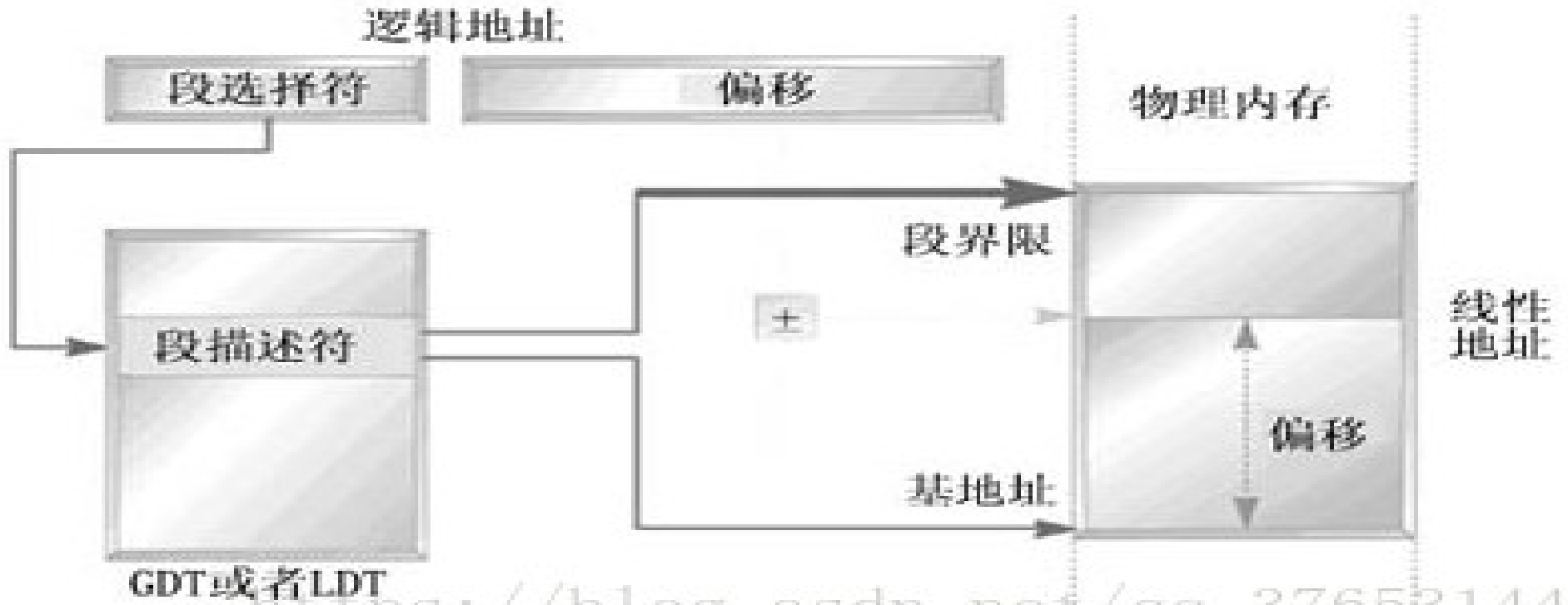
2.4 段模式下寻址方式

3. 段选择子：

进入 32 位保护模式，段寄存器 CS 、 DS 、 ES 、 FS 、 GS 、 SS ，它们还是 16 位的，但是不再存着所谓的段地址了，而是成为段选择器，里面存着段选择子。

段选择子里面有描述符索引，根据这个索引去位于内存的 GDT （Global Descriptor Table 全局描述符表）里找真正的段基地址（线性地址）。通过 Selector （段选择子）找到存储在 Descriptor Table （描述符表）中某个 Descriptor （段描述符），该段描述符中存放有该段的物理首地址，所以就可以找到内存中真正的物理段首地址 Segment





谢谢观看

- 第 7 组 -

汇报人：胡峻豪，胡天磊

时间： 2022.9.23

小组成员：胡才郁，张俊雄