
LUNIX
Operating System
从零开始
DEVELOPING YOUR OWN
自制操作系统
OPERATING SYSTEM FROM SCRATCH
O Brave New Kernel
安装 GDT
OPERATING SYSTEM DEVELOPMENT
TUTORIAL SERIES
EP 5-6

关于 LGDT

指令 lgdt ： 加载 GDT 的地址到 GDTR

用法： `lgdt (%reg)`

寄存器 %reg 的值是 GDTR 的值**的地址**

Recall:

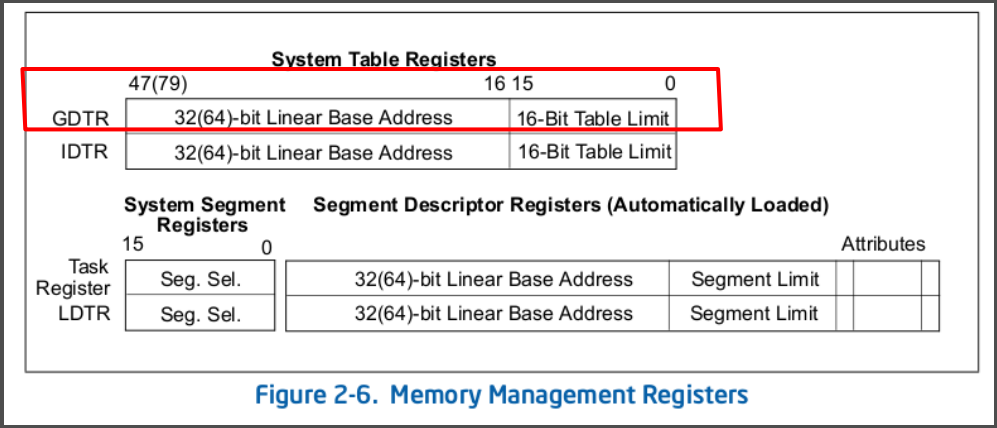


Figure 2-6. Memory Management Registers

WHY?

GDTR 的大小为 48bits
而 x86 的操作数宽度和所有通用寄存器只有 32bits
所以 GDTR 的值只能放在内存，而后由 CPU 直接从内存读出。

Some Reminders….

需要处理的寄存器：

CS, ES, SS, DS, FS, GS

gdt 结构：

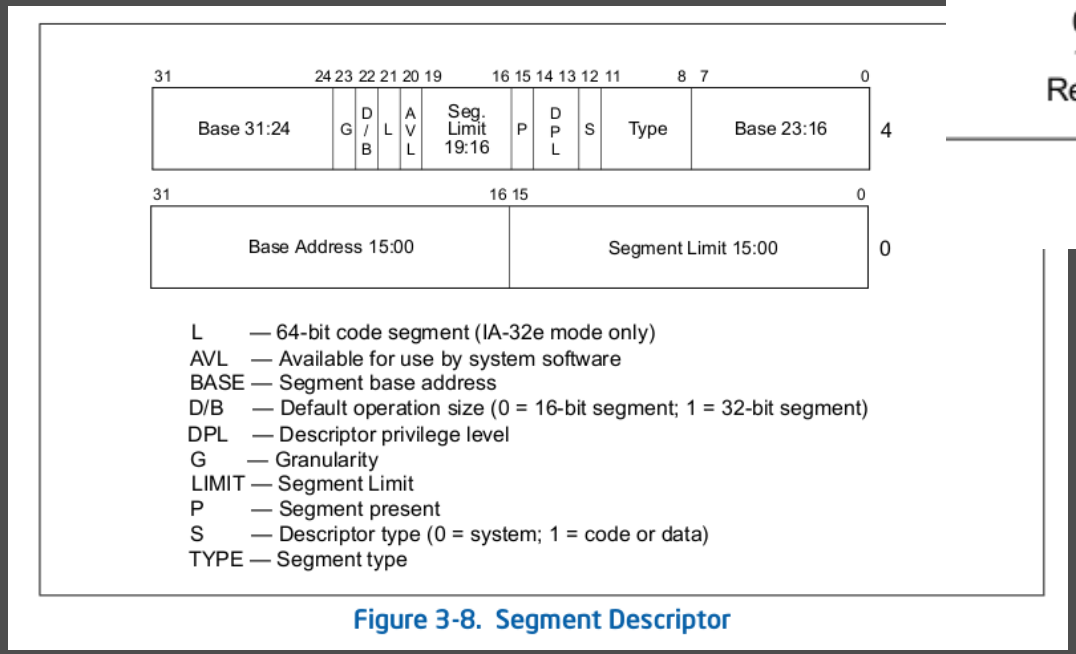


Figure 3-8. Segment Descriptor

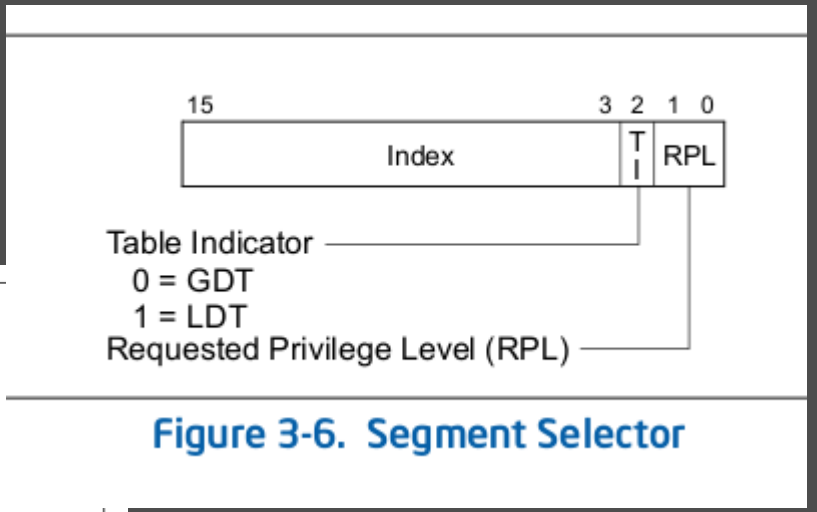


Figure 3-6. Segment Selector

Let’s Code

关于修改 CS:IP 的那些事儿

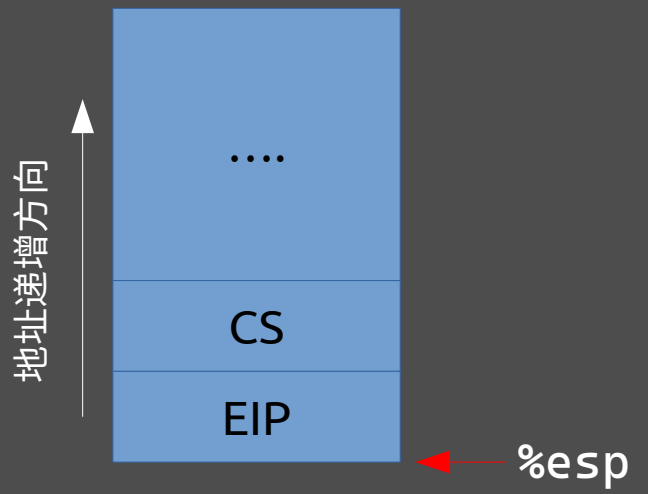
修改 IP： 各种 jump ，返回，调用 （近距离）

修改 CS （与 IP）： 远距离 { 跳转 | 返回 | 调用 }

Example： `retf` – 远距离返回 (far return)

```
RETURN-TO-SAME-PRIVILEGE-LEVEL:
    IF the return instruction pointer is not within the return code segment limit
        THEN #GP(0); FI;
    IF OperandSize = 32
        THEN
            EIP := Pop();
            CS := Pop(); (* 32-bit pop, high-order 16 bits discarded *)
        ELSE (* OperandSize = 16 *)
            EIP := Pop();
            EIP := EIP AND 0000FFFFH;
            CS := Pop(); (* 16-bit pop *)
    FI;
```

Source: Intel manual. pp.1748




```
pushl <new CS value>
pushl <label>
retf
```