

汇编语言教程 - - 教程(4) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/137/2021_2022__E6_B1_87_E7_BC_96_E8_AF_AD_E8_c98_137771.htm 第三章 操作内存在前面的章节中，我们已经了解了寄存器的基本使用方法。而正如结尾提到的那样，仅仅使用寄存器做一点运算是没有什么太大意义的，毕竟它们不能保存太多的数据，因此，对编程人员而言，他肯定迫切地希望访问内存，以保存更多的数据。我将分别介绍如何在保护模式和实模式操作内存，然而在此之前，我们先熟悉一下这两种模式中内存的结构。

3.1 实模式

事实上，在实模式中，内存比保护模式中的结构更令人困惑。内存被分割成段，并且，操作内存时，需要指定段和偏移量。不过，理解这些概念是非常容易的事情。请看下面的图：段-寄存器这种格局是早期硬件电路限制留下的一个伤疤。地址总线在当时有20-bit。然而20-bit的地址不能放到16-bit的寄存器里，这意味着有4-bit必须放到别的地方。因此，为了访问所有的内存，必须使用两个16-bit寄存器。这一设计上的折衷方案导致了今天的段-偏移量格局。最初的设计中，其中一个寄存器只有4-bit有效，然而为了简化程序，两个寄存器都是16-bit有效，并在执行时求出加权和来标识20-bit地址。偏移量是16-bit的，因此，一个段是64KB。下面的图可以帮助你理解20-bit地址是如何形成的：段-偏移量标识的地址通常记做 段:偏移量 的形式。由于这样的结构，一个内存有多个对应的地址。例如，0000:0010和0001:0000指的是同一内存地址。又如， $0000:1234 = 0123:0004 = 0120:0034 = 0100:0234$ $0001:1234 = 0124:0004 = 0120:0044 = 0100:0244$ 作为负

面影响之一，在段上加1相当于在偏移量上加16，而不是一个“全新”的段。反之，在偏移量上加16也和和在段上加1等价。某些时候，据此认为段的“粒度”是16字节。练习题尝试一下将下面的地址转化为20bit的地址：2EA8:D678 26CF:8D5F 453A:CFAD 2933:31A6 5924:DCCF 694E:175A 2B3C:D218 728F:6578 68E1:A7DC 57EC:AEEA 稍高一些的要求是，写一个程序将段为AX、偏移量为BX的地址转换为20bit的地址，并保存于EAX中。[上面习题的答案] 我们现在可以写一个真正的程序了。经典程序：Hello, world ... 应该得到一个29字节的.com文件

```
.MODEL TINY .CODE CR equ 13 LF equ 10
TERMINATOR equ ' $ ' ORG 100h Main PROC mov dx,offset
sMessage mov ah,9 int 21h mov ax,4c00h int 21h Main ENDP
sMessage: 100Test
```

下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com